sensors

MDPI

Article

Distributed Resources Allocation Method for Space–Ground Integrated Mobile Communication System

Tingyin Zhao [1,2] and Zhidu Li [1,2,*]

[1] School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; 2021210213@stu.cqupt.edu.cn

[2] Information and Communications Institute, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

* Correspondence: lizd@cqupt.edu.cn

Abstract: This paper presents an innovative approach towards space–ground integrated communication systems by combining terrestrial cellular networks, UAV networks, and satellite networks, leveraging advanced slicing technology. The proposed architecture addresses the challenges posed by future user surges and aims to reduce network overhead effectively. Central to our approach is the introduction of a marginal mobile station (MS)-assisted network resource allocation decision architecture. Building upon this foundation, we introduce the DP-DQN model, an enhanced decision-making algorithm tailored for MSs in dynamic network environments. Furthermore, this study introduces a feedback mechanism to ensure the accuracy and adaptability of the marginalization model over time. Through extensive simulations and experimental validations, our DP-DQN-based edge decision method demonstrates substantial potential in alleviating core network overhead while improving success access ratios compared to conventional methods.

Keywords: space–ground integrated system; resources allocation; network slicing; GNSS

check for updates

Citation: Zhao, T.; Li, Z. Distributed Resources Allocation Method for Space–Ground Integrated Mobile Communication System. *Sensors* **2024**, *24*, 4711. https://doi.org/10.3390/s24144711

Academic Editor: Omprakash Kaiwartya

Received: 24 June 2024
Revised: 17 July 2024
Accepted: 18 July 2024
Published: 20 July 2024

1. Introduction

The integrated space–ground communication systems hold a promising future in the evolving landscape of mobile communication networks, particularly with their potential to expand coverage significantly [1–3]. Such systems are pivotal in advancing the Internet of Things (IoT), enabling a vast network of interconnected devices across more extensive and varied terrains [4–6]. Moreover, the fusion of these integrated systems with existing cellular networks promises to diversify and enrich the service types available to individual users, marking a significant leap forward in communication technology [7,8].

To realize the full potential of space–ground integrated communication systems, one crucial challenge is the efficient allocation of resources [9–11]. This involves the development of network slicing strategies and cross-network resource scheduling algorithms that take the unique characteristics of both satellite and cellular networks into account. Specifically, the location of MS will no longer be limited to the ground, but will expand and include sea and air [12–15]. Thus, certain devices can access not only the cellular network but also the satellite or drone network at the same time. When necessary, the MS also needs to switch between the network. Therefore, for the new generation of network resource allocation algorithms, it is necessary to include location information in order to maximize network efficiency and ensure seamless service delivery across the different layers of the integrated system.

Despite numerous advancements, current resource allocation methods, particularly those based on deep reinforcement learning techniques such as Deep Q-Networks (DQN) and Deep Deterministic Policy Gradient (DDPG) exhibit limitations. These algorithms often fall short in real-time performance and fail to consider the integration of multiple physical channels within the space–ground system, thus leaving considerable room for improvement in network efficiency. These gaps in existing methodologies underscore the necessity for our research.

To address these shortcomings, we introduce a novel integrated network slicing approach coupled with a Distributed Deployment Deep Q-Network (DP-DQN) resource allocation method. Our approach decentralizes the decision-making process by distributing the decision models to the user end, thus facilitating on-the-spot network resource decisions. This not only ensures real-time responsiveness but also reduces network overhead while accommodating the demands of various physical channels [16–19]. Through this innovation, our study seeks to significantly enhance the operational efficiency of space–ground integrated communication systems.

The contributions of this paper are summarized as follows:

- Firstly, we propose a new converged system based on the existing infrastructure of cellular and satellite communication networks, in which all resources are deployed in a unified manner rather than "separate and conquer".
- Secondly, based on the idea of SDN, we propose a new network controller based on a distributed resource allocation decision-making model, which assigns the implementation body of decision-making behavior from the original core network controller to the access mobile terminal, so as to alleviate the dilemma of the rapid increase in access volume which will cause a huge burden on the core network in the future.
- Thirdly, based on DQN and negative feedback network technology, we propose a new DP-DQN model based on reinforcement learning for the overall system resource allocation, so that the mobile station can consider the problem of resource allocation from the perspective of the core network overhead, so as to make access requests conducive to the efficiency of the core network.

The remainder of this article is organized in the following order. The related works involved in this paper are reviewed in Section 2. In Section 3, the allocation system model is proposed and the optimization problem is formulated. Data rate optimization is investigated in Section 4 while physical network optimization is studied in Section 5. Finally, simulation and numerical results are presented and discussed in Section 6 followed by the Section 7.

2. Related Work

In this section, the paper introduces the related works from three aspects: network slicing, satellite navigation and resource allocation methods. In previous studies, these three topics were usually studied separately.

2.1. Network Slicing

Slicing technology is proposed to enhance the differentiated service capability of the network, which is mainly carried by the cellular mobile communication network. In recent years, the concept of satellite slicing has been proposed, so we introduce it from the following two parts: cellular network slicing and satellite network slicing.

In Cellular Scenarios, emphasis is usually placed on improving slice efficiency and increasing service types while reducing inter-slice interference. In [20], the author introduced a novel cooperative multi-agent reinforcement learning (RL) algorithm for RAN slicing, designed to adapt to variable slice numbers and effectively scale as they grow. In [21], the author formulated the slice-based service function chain embedding (SBSFCE) problem as an integer linear programming (ILP) that aims to fulfill differentiated requirements of flows. In [21], the proposed architecture leveraged new SDN extended federation modules in compliance with the ETSI requirements for inter-MEC system coordination. In [22], the author presented an innovative actor–critic Reinforcement Learning (RL) model named Slice Isolation based on RL (SIRL) to ensure the isolation between different slices while maximizing the user requests. In [23], the author proposed a QoS and security-oriented slicing resource allocation scheme in a multi-cell and multi-slice scenario in order to minimize the slice interference.

In Satellite Scenarios, emphasis is usually placed on improving resource usage efficiency and communication coverage. In [24], the author proposed a trust-based satellite Internet resource-slicing access authentication scheme to solve the efficient and secure access requirements in satellite communication. In [25], the author proposed a two-layer dynamic reconfigurable RAN slicing architecture for the ultra-dense low earth orbit satellite network (UD-LSN) to improve the slices' efficiency. In [26], an architecture of IoT supportable satellite edge computing (SatEC) enabled by LEO satellite was proposed for global coverage extension and 3-D mobility enhancement. In [27], the author proposed an optimization satellite slicing framework able to exploit the available resources allocated to the defined network slices in order to meet the diverse QoS/QoE requirements exposed by the network actors. In [28], a hierarchical resource slicing framework was proposed for dynamic allocation of multidimensional resources to solve the problem of resource slicing and scheduling of joint 3C resources in a RAN edge scenario assisted by LEO content caching.

In summary, previous research on network slicing technology has regarded satellite and cellular networks as two independent individuals, and ignored the possibility of combining the two systems. This circumstance in the future wide coverage and diversified services of the new generation of mobile communication scenarios will lead to the problem of resource utilization.

2.2. Satellite Navigation

The Global Navigation Satellite System (GNSS) plays a pivotal role in delivering positioning services that are essential for a wide range of applications, from personal navigation to complex industrial operations. In recent years, there have been two research hotspots: satellite signal processing methods and navigation algorithms in severe climate situations.

In Signal Processing, the focus of research is often on how to improve the positioning accuracy by changing the transmission sequence or modulation mode. In [29], the author proposed an optimization-based tightly-coupled precise point positioning (PPP) inertial navigation system (INS) vision integration method to achieve precise and continuous state estimation. In [30], the author presented a new broadband multi-carrier navigation modulation, namely orthogonal frequency division multiplexing with binary offset carriers (OFDM-BOC) in order to improve spectrum efficiency, tracking performance and anti-interference ability. In [31], a novel M-estimation-based robust iterated cubature Kalman filter (ICKF) was developed to minimize the impact of GNSS outliers while improving the correction effect of high-quality line-of-sight (LOS).

In Severe Climate Algorithms, the focus of research is the algorithm design to reduce the interference in remote areas with severe climate. In [32], the author implemented the tightly coupled integration of navigation satellites with a low-cost micro-electromechanical system's (MEMS) inertial measurement unit (IMU) to improve vertical accuracy and applied nonholonomic constraints to improve the standalone MEMS inertial navigation system (INS) performance during an outage. In [33], the author developed an adaptive–robust fusion strategy for low-cost GNSS systems which can provide reliable fusion positioning solutions when the GNSS signal is challenged.

In summary, navigation systems used to be studied as separate modules. Other navigation-related services simply invoke the navigation module as clients. Such a fragmented design approach obviously has a large room for improvement in resource utilization and system reliability, especially in many emerging scenarios that require navigation system assistance.

2.3. Network Decision Method

In the research on network resource allocation, there are two main aspects: traditional non-AI resource allocation algorithms and AI-assisted intelligent resource allocation algorithms. Because of its unique flexibility, the latter can often greatly improve the efficiency of the network.

In non-AI Methods, a distributed power allocation was performed in [34] to optimize the performance of cell-edge users by Lagrange method. In [35], the author introduced a distributed structure for the resource allocation problem by forming a convex optimization problem, maximizing the overall system utility function. In [36], a new resource allocation optimization and network management framework for wireless networks using neighborhood-based optimization has been proposed instead of fully centralized or fully decentralized methods to reduce interference and increase capacity.

In AI-based Methods, researchers are aiming to offer a more dynamic and responsive approach to resource allocation. In [37], the author proposed a multi-agent deep reinforcement learning (DRL) approach with an action space reduction strategy to achieve the dynamic VNF orchestration, backup and mapping solution. In [38], the author proposed a resource allocation (RA) method using dueling double deep Q-network reinforcement learning (RL) with low-dimensional fingerprints and soft-update architecture (D3QN-LS) based on a Manhattan grid layout urban virtual environment. In [39], an improved deep Q-network (DQN) algorithm was introduced to improve the efficiency of resource utilization. In [40], the author investigated the dynamic offloading of packets with finite block length (FBL) in an edge-cloud collaboration system consisting of multi-mobile IoT devices (MIDs) with energy harvesting (EH), multi-edge servers and one cloud server (CS) in a dynamic environment based on a multi-device hybrid decision-based DRL solution. In [41], based on the enhanced K-mean algorithm and multi-agent PPO (MAPPO) algorithm, a cooperative trajectory design method was proposed for the UAVs to minimize interaction overhead and optimize deployment efficiency. In [42], the author proposed a novel algorithm based on Soft Actor–Critic (SAC) to solve the system cost minimization problem considering vehicle users' satisfaction, RSUs' cost and vehicle workers' reward.

In summary, although AI-based network resource allocation methods solve the problems of traditional non-AI methods in terms of flexibility, the current AI methods, especially the use of deep network cases, still have problems when talking about real-time processing. In addition, these methods have rarely taken space–ground integrated circumstances into consideration.

2.4. Motivation of Our Works

In summary, current research on network slicing, satellite navigation and resource allocation is poor in terms of the converged control and real-time requirement emphasized by the integrated space–ground communication system, which motivates this paper.

3. System Model

In this section, we consider a new kind of distributed resources allocation structure based on the slicing technology of the space–ground integrated communication system.

3.1. Network Scenario

In the architecture designed by us, there are three types of physical information carriers, namely satellite networks, cellular networks and drone networks, which are marked as M_1, M_2 and M_3. At the same time, each physical channel contains three network slices, T_1, T_2 and T_3. By the way, we define a virtual concept as a 'location area'. As shown in Figure 1, each 'area' contains several MSs with the similar properties.

Also, each of the networks M_i will serve a number of MSs. For example, the distances between users from K_{21} to K_{2N} and certain base stations are relative high, where we call them sub-urban or rural areas. In these circumstances, we introduce a satellite network that acts as the carrier of eMBB service instead of cellular network.

In Figure 1, we can see a special access device, the K_{2N}. For the same kind of service slice, its state is bounded between suburban and urban areas, so both networks can be used as its carrier.

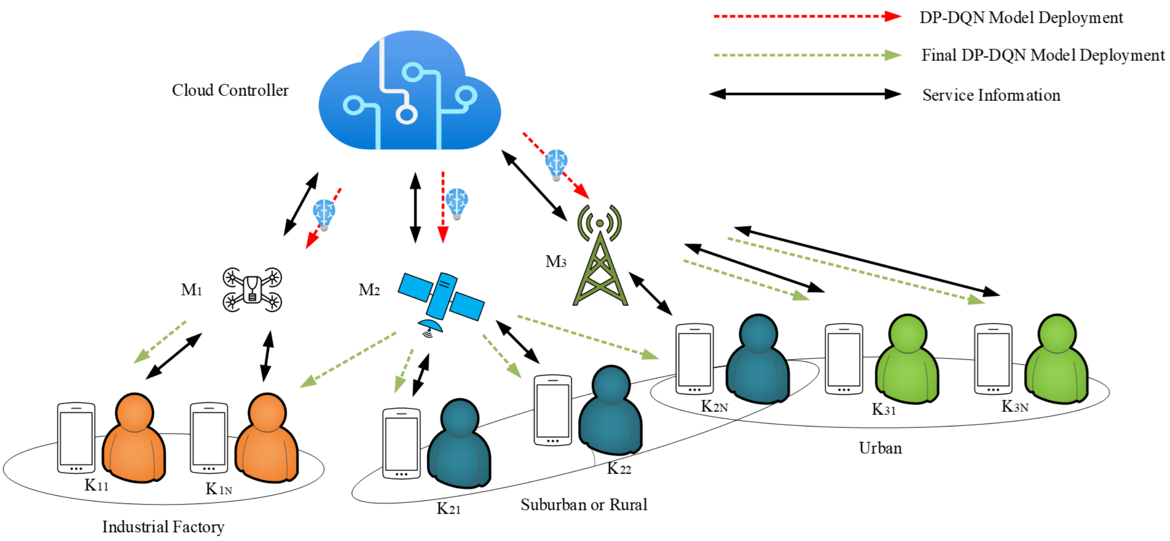


Figure 1. The deployment of decision model: the decision model will be pre-trained on the core network controller and sent to the MS.

In order to select a more suitable network for access, we conduct our works based on core network overhead, which means the network access decision will be made by the MS from the perspective of network overhead.

The whole decision process of network access at the MS side will follow the procedure below.

- First of all, the MS will analyze the data size and service type of the data to be sent, and determine its transmission power.
- Secondly, this information is fed into the DP-DQN model (discussed in detail later in this article) delivered from the network side and determines the type of network that is ultimately accessed.

Different from the traditional network, the training and deployment process of DP-DQN model is based on SDN (Software Defined Network). Here, we introduce the DP-DQN deployment method.

Firstly, as shown in Figures 1 and 2, the basic model will be trained at the side of the core network controller. Specifically, the input parameters are service types, average data volume and current network congestion. After this process, the pre-trained model is sent to each network carrier without distinction. The red dashed line in Figure 1 represents the model transfer flow for this process.

Secondly, each network will then train the model twice, adding its own parameters based on the model of core network controller, such as the signal transmission power on the network side, the average delay of the network and the user amount in the coverage.

Finally, these secondary training models will be passed to the MS at the edge, where these models will undergo the final training. Here, the third (final) training process at the MS side will also conduct integrated learning with all the models received. Here, the training of the model will take the following parameters into account: the user's own average amount of data, location information (i.e., latitude and longitude altitude), the state of the three networks at this time and the records of past resource decisions. Thus, what the core network controller should do is to keep monitoring the crowd state while renewing the basic decision model.

The parameters included in the three training sessions are summarized in Table 1.

Table 1. Summary of parameters in the three training processes.

Training Process	Parameter Descriptions
Round 1	Total Service Types Average Data Volume Network Congestion
Round 2	Network Side Transmission Power Average Delay of the Network User Amount in the Coverage
Round 3	Average Data Amount MS Location Information States of Three Networks Records of Past Resource Decisions

The overall frequency resource mapping relationship is shown in Figure 3. On the left is the frame structure of the three physical networks, in the middle is the time–frequency domain resource allocation diagram and on the right is the data packet to be sent by the user. The data to be sent by the user are first analyzed to determine the type of service they belong to. The data are then mapped to the appropriate resource block. Finally, the access network is determined according to the user's location and other related information.

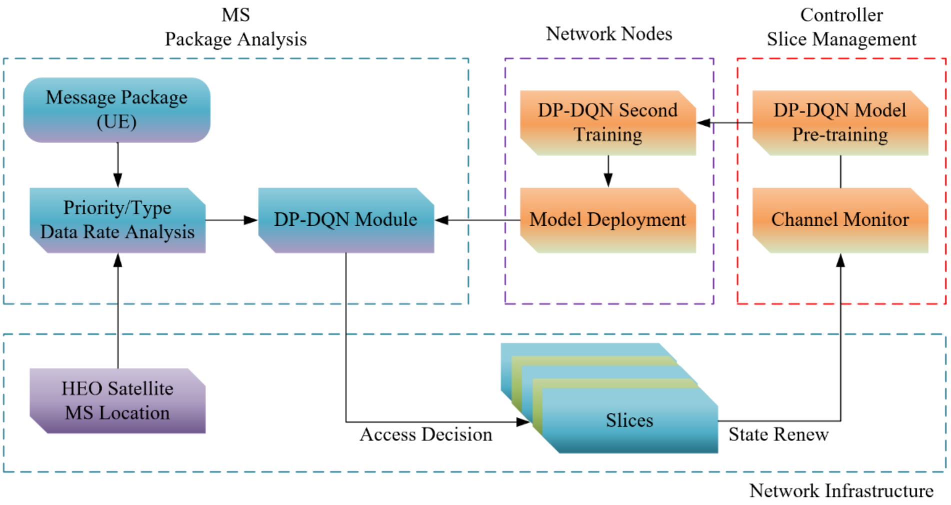


Figure 2. Distributed network decision-making structure.

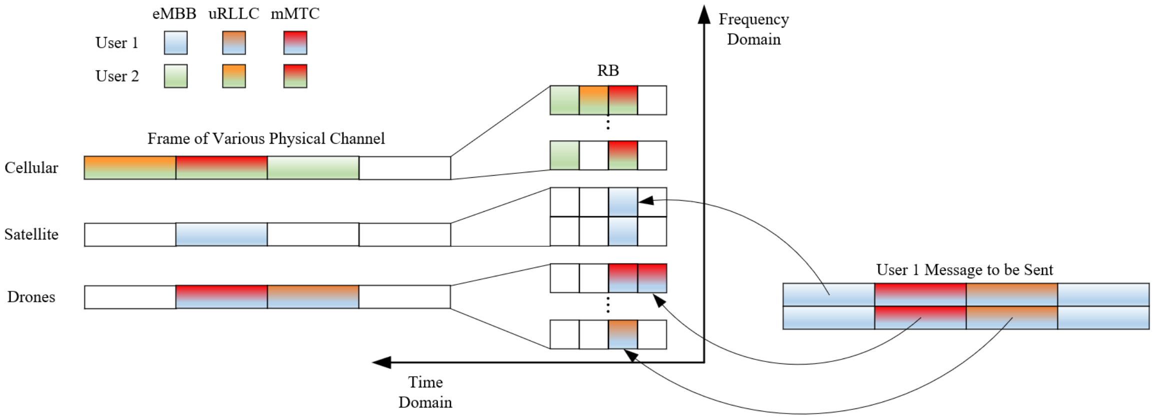


Figure 3. The diagram contains three service types (three slices) for two users. Here, user 1 is located in countryside industrial areas away from the cellular network, so the eMBB service is mapped to the satellite network, while mMTC is mapped to the drone network.

In Figure 2, the allocation decision-making process is given, consisting of three aspects, package and access strategy analysis at the MS side, slice management at the side of the core network controller and the location analysis provided by navigation systems (GNSS).

Figure 2 is consistent with what is described in the previous chapters, and we further elaborate it from the perspective of model training. In the initial training stage, the DP-DQN model will first be trained based on the reinforcement learning method, and then in the process of continuous iterative update of the initial model, data from Channel Monitor will be used to improve the adaptability of the model. Then, the secondary training will be performed on each network node before the final deployment of the model. On the client side, the data to be sent are packaged by Message Package block, analyzed in the Type Analysis module and given a priority. At the same time, the current MS location information is obtained by the MS location module. Each time data are sent, the above information is obtained, and combined with the final resource allocation result and system feedback, the MS side will form a data pool. Subsequent edge DP-DQN models will be trained based on this.

3.2. User Request Procedure

To illustrate the network access process, we take one MS (K_{2N}) as an example, using the architecture in Figures 1 and 2.

Firstly, the core network controller sends the pre-trained model with the real-time crowd state to all the physical network carriers, and then these nodes send the model to the K_{2N} after secondary training.

Secondly, K_{2N} in the certain area M_2 ask for a connection to send messages through the network. Here, the message will be analyzed by the designed algorithm to give it a classification of service type and allocated data rate at the same time. Also, K_{2N} gets its location (i.e., longitude, dimension and altitude) from the HEO navigation satellite as other parameters.

Thirdly, the algorithm embedded in MS will use these parameters to calculate the most suitable RB requirement with transmitting power. Then, K_{2N} will employ the DP-DQN model to decide which physical network (e.g., Satellite Network eMBB Slice) to switch in. Here, for K_{2N} which is located in the near-urban area, the slices allocated from the satellite will be much more expensive than those from the cellular network.

Finally, the analysis result (service type, access network and needed RB number) will be sent to a particular network. Available slice blocks will then be allocated to the user to meet the demand of data rate. If the current network no longer has enough resources due to resource requests from other users, the core network will meet its needs as far as possible according to the principle of proximity. If this is still not satisfied (in practice, such a scenario is very rare), feedback will be sent to the mobile station to recalculate the appropriate access network.

3.3. Network Update Process

In our design, the update of the network will consist of two parts: the core network controller and the MS controller. The former is mainly to update the current usage and congestion index of each slice and network in real time, while the latter is to update the location of MS in real time.

3.3.1. Channel State Update

The channel state monitor block will keep updating every t minutes, checking the usage rate of each slice.

3.3.2. Positioning Update

The location monitor block will keep asking for new data from the HEO satellite every t minutes.

The parameter t is not concrete which will change depending on the characteristics (e.g., speed or altitude) of the MS.

Since the whole process and decisions are made by the MS, there may be some mistakes due to the message latency. For example, when MS performs slice selection analysis, because the network has not had time to update the latest network situation for MS, the slice selected by MS may be assigned to other users at this time, but MS does not know this situation, resulting in allocation (access) failure.

Therefore, the proposed procedure contains a feedback loop which is designed to solve the possible volatility of the network, as you can see in Figure 4. Every failure record of allocation will be transmitted backward to the first step to lower the fail rate of the next allocation decision.

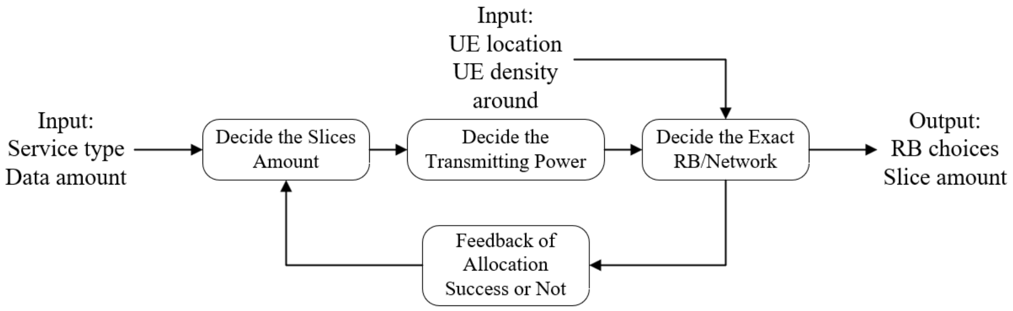


Figure 4. Distributed network decision-making process flow.

3.4. Problem Formulation

As is mentioned above, to relieve the core network overhead, we shift some of the work of allocation from the original core network controller to the mobile station side, that is, the mobile station will first determine which physical slice (satellite or cellular network) is the most suitable for itself. Considering that slices are different in the latency and the channel atmosphere, we introduce a parameter C to describe the cost of certain slices to help the MS make the decision.

$$\mathbf{C} = [\mathbf{Cost}] = G\big(\mathbf{T},\ \mathbf{R'}, \mathbf{A}(t), \mathbf{P}_{kb}(t)\ \big) \tag{1}$$

$$= \lambda_1 \mathbf{T} + \lambda_2 \mathbf{R'} + \lambda_3 \mathbf{A}(t) + \lambda_4 \mathbf{P}_{kb}(t) \tag{2}$$

where $\mathbf{R'}$ denotes the actual allocated data rate of users

$$\mathbf{R'} = \mathbf{R}\mathbf{G_e}(\mathbf{R},\mathbf{T}) = \mathbf{R}\frac{\mathbf{G}(\mathbf{R},\mathbf{T})}{100} \tag{3}$$

among which, $G_{ij}(R,T) \in [0,100]$ is an urgency factor to describe how much content in the package is valuable to be given the first priority, which has been fully investigated in [43–45].

The resources allocated for user access should be as small as possible under the premise of meeting its priority and data volume, so as to improve the data utilization efficiency of the system. Hence, the allocation procedure can be transformed to an optimization problem.

$$\underset{\mathbf{A}(t),\mathbf{P}_{kb}(t),\mathbf{R'}}{\mathbf{minimize}}\ (\mathbf{C}) = \underset{\mathbf{A}(t)}{\mathbf{minimize}}\big(G\big(\mathbf{T},\mathbf{R'},\mathbf{A}(t),\mathbf{P}_{kb}(t)\big)\big) \tag{4}$$

$$s.t.\ \ A_{ij}(t) \in [0,1], 1 \le i \le 3, 1 \le j \le N \tag{5}$$

$$\sum_{i=1}^{3}\sum_{j=1}^{N} A_{ij} \le C_{ch} \tag{6}$$

Here, in our research, we propose three types of slices as mentioned before, thus, $T = Z^+, [1,3]$ which is a $1 \times N$ array where N indicates the number of total requests. Parameter $A = R^+, [0,1]$ is a $3 \times N$ array aimed to describe the state of every slice and

RB (used, crowed, etc.) which is positively correlated with channel allocation cost in the formula. P indicates the density of MS around a particular mobile devices.

Sorting through the optimization problems we just listed, we can get the following formula.

$$\mathbf{P1}:\underset{\mathbf{A}(t),\mathbf{P}_{kb}(t),\mathbf{R}'}{\mathbf{minimize}}\left(\lambda_1\mathbf{T}+\lambda_2\mathbf{R}'+\lambda_3\mathbf{A}(t)+\lambda_4\mathbf{P}_{kb}(t)\right) \tag{7}$$

$$s.t.\ \sum_{i=1}^{3}\sum_{j=1}^{N}A_{ij}\le C_{ch} \tag{8}$$

$$T_{ij}\in\{1,2,3\},1\le i\le 3,\ 1\le j\le N \tag{9}$$

$$A_{ij}(t)\in[0,1],1\le i\le 3,1\le j\le N \tag{10}$$

The problem is to find the best couple $\mathbf{A},\mathbf{P}$ and $\mathbf{R}'$ to minimize the cost function. Formula (7) restricts the maximum capacity of the logical channel, while formula (10) defines the allocation state of the slice. Corresponding to the channel slicing problem, the goal of each MS when asking for access to the network is to find the best physical network to switch in reflected by $\mathbf{A}$, the transmitting power $\mathbf{P}$ and proposed data rate $\mathbf{R}'$.

However, the optimization problem listed here is a NP-hard problem which cannot be worked out by the MS quickly. Thus, we divide the whole problem into two sub-optimization problems: allocated data rate optimization, **P2**, and allocated physical network optimization, **P3**.

The first optimization problem **P2** is to find the optimal transmit power and the number of resource blocks to meet the needs of users under the assumption of a physical distribution network. In the previous section, we discussed the goal for the MS in the resource allocation problem **P1** as a minimization problem, where one of the goals is to minimize R'. In other words, it is to make R as large as possible, but not more than R', which is the minimum transmission rate to meet the requirements of k service.

$$\mathbf{P2}:\underset{\mathbf{P}_{kb}(t),\boldsymbol{\varnothing}}{\mathbf{maximize}}\ (\mathbf{R}_k)=\sum_{b\in B}\boldsymbol{\varnothing}\mathbf{B}_{kb}\frac{\sum_{b\in B}a_{kb}(t)}{M}\times\log_2(1+\frac{\mathbf{P}_{kb}(t)\mathbf{h}_{kb}(t)}{\sigma_k^2(t)}) \tag{11}$$

$$s.t.\ \mathbf{R}_k\le\mathbf{R}'_k \tag{12}$$

$$\sum_{b\in B}B_{kb}\le B_b \tag{13}$$

$$\sum_{b\in B}P_{kb}\le P_b \tag{14}$$

$$\sum_{i=1}^{3}\sum_{j=1}^{N}A_{ij}(t)\le C_{ch} \tag{15}$$

$$T_{ij}\in\{1,2,3\},1\le i\le 3,\ 1\le j\le N \tag{16}$$

$$A_{ij}(t)\in[0,1],1\le i\le 3,1\le j\le N \tag{17}$$

In the above optimization problem, the symbol χ represents the proportion of resource panes that can actually be used in an allocated resource block.

$$\chi=\frac{\sum_{b\in B}b_{kb}(t)}{M} \tag{18}$$

The parameter $b_{kb}(t)$ describe the state of each slice (i.e., whether it will be allocated to current service T or not) which is a crucial factor in the allocation process.

$$b_{kb}(t)=\begin{cases}1, & \text{if } \textit{the slice has been allocated}\\ 0, & \text{else}\end{cases} \tag{19}$$

The second optimization problem is to select the most appropriate allocated physical channel according to the characteristics of the users, so as to minimize the total overhead when the transmission power and the number of allocated resource blocks are determined.

$$\mathbf{P3}:\underset{\mathbf{A}(t)}{\mathbf{minimize}}\ (\mathbf{C}) = \underset{\mathbf{A}(t)}{\mathbf{minimize}}\big(G\big(\mathbf{T},\mathbf{R}',\mathbf{A}(t),\mathbf{P}_{kb}(t)\big)\big) \tag{20}$$

$$s.t.\ \ A_{ij}(t) \in [0,1], 1 \le i \le 3, 1 \le j \le N \tag{21}$$

$$\sum_{i=1}^{3}\sum_{j=1}^{N} A_{ij} \le C_{ch} \tag{22}$$

The sub problems **P2** and **P3** will be discussed in the following sections. The whole solving process has been illustrated in Figure 4. All the variables defined in this article are summarized in Table 2.

Table 2. Summary of parameters.

Parameters	Description
$\mathbf{R}$	Reference transmission rate matrix of services
$\mathbf{Z}$	Package size matrix of users
$\mathbf{T}$	Service type matrix of users
$\mathbf{S_p}$	Priority score matrix of users
$\mathbf{R}'$	Actual allocated data rate matrix
$\mathbf{P}_{kb}(t)$	Transmitting power matrix of users
M	Number of MS per unit time
B_{kb}	Allocated bandwidth of a certain user
$P_{kb}(t)$	Transmitting power of a certain user
$h_{kb}(t)$	Channel state array of a certain physical link
$\sigma_k^2(t)$	Noise power of a certain channel
L_i	Location parameter of a certain user
D_{ij}	Distance between two access node in Km
$\rho(t)$	MS density of a certain location area
C	Cost function of an allocation decision
C_{ch}	Capacity of a certain slice
$\mathbf{A}(t)$	Slice allocation states(allocated or not)
$\mathbf{Ø}$	Resource availability of a certain slice
$\mathbf{S}(\mathbf{t})$	User real time parameter (e.x. data amount)
$\mathbf{PW}(\mathbf{t})$	Penalty function of DP-DQN
$\mathbf{DA}$	Congestion Rate of slices
$\mathbf{L_c}$	The distance from MS to cellular access node
$\mathbf{L_u}$	The distance from MS to UAV access node
AS	The successful access ratio of several MS
Bd_c	The network overhead of control panel
Bd_u	The network overhead of user panel

4. Data Rate Optimization

In our research, logical channels are defined from the perspective of the system, that is, a logical channel contains different mappings of physical channels, so different resource blocks correspond to different channel noise and raw resource allocation. Therefore, when solving the optimization problem, the MS needs to solve R separately in several different physical mappings and get the final optimal scheme.

Thus, the problem **P2** can be conducted in these two process named as:

- **P4**: First, it is assumed that in all physical mapping cases, the transmitted power of MS is the same. On this basis, **R** is calculated and the optimal allocation resource block amount is selected.
- **P5**: Then, when the allocated resource block is determined, the transmission power $P_{kb}(t)$ is changed to obtain the optimal transmitting rate **R**.

4.1. **P4** *Optimization*

For a given transmitting power and the current slice state, the RB allocation question can be formulated as

$$\textbf{P4}:\underset{\textbf{Ø}}{\textbf{maximize}}\ (\mathbf{R}_k) = \sum_{b\in B} \textbf{Ø}\mathbf{B}_{kb} \frac{\sum_{b\in B} a_{kb}(t)}{M} \times \log_2(1 + \frac{\mathbf{P}\mathbf{h}_{kb}(t)}{\sigma_k^2(t)}) \tag{23}$$

$$s.t.\ \ \mathbf{R}_k \leq \mathbf{R'}_k \tag{24}$$

$$\sum_{b\in B} B_{kb} \leq B_b \tag{25}$$

$$\sum_{i=1}^{3}\sum_{j=1}^{N} A_{ij}(t) \leq C_{ch} \tag{26}$$

$$T_{ij} \in \{1,2,3\}, 1 \leq i \leq 3,\ 1 \leq j \leq N \tag{27}$$

$$A_{ij}(t) \in [0,1], 1 \leq i \leq 3, 1 \leq j \leq N \tag{28}$$

where **Ø** is a matrix of χ_{kb}.

We design an optimization Algorithm 1 for the MS to find out the proper slices named FPCB (Fixed Power Change the Block).

Algorithm 1 Algorithm of FPCB (**P4**)

```
Require: $A$, $P$, $B_{kb}$, $P$ (This is Inputs)
Ensure: $B$, $A$ (This is Outputs)
Set the initiate solution $B^{(0)}, P, A$
while $A$ is not full do
  calculate the capacity of each RB
  for each $n \in [1, length(A)]$ do
    if $A_n$ is not full then
      for each $i \in [1, length(A_n)]$ do
        allocate this slice to the MS
        calculate the rate of this slice in the block n
        $R$ = $R$ + $R_i$
      end for
    else
      jump to the next RB
    end if
    Compare which of the block has a higher rate and select it as $B$
    Try other kind of $B$
    Renew the channel allocation $A$
  end for
end while
return Outputs $B$, $A$
```

4.2. **P5** *Optimization*

Now that the proper slices of RB have been found out, the task is to change the transmitting power accordingly to get the result.

$$\textbf{P5}:\underset{\mathbf{P}_{kb}(t)}{\textbf{maximize}}\ (\mathbf{R}_k) = \sum_{b\in B} \textbf{Ø}\mathbf{B}_{kb} \frac{\sum_{b\in B} a_{kb}(t)}{M} \times \log_2(1 + \frac{\mathbf{P}_{kb}(t)\mathbf{h}}{\sigma_k^2(t)}) \tag{29}$$

$$s.t. \sum_{b \in B} P_{kb} \leq P_b \tag{30}$$

Since we assume that the data of the same service request from the same MS will all be mapped to the same type of physical channel (for example, language information from user A in remote areas can all be mapped to satellite channels), the channel matrix here can be set to be the same when the transmission power is optimized, that is, the transmission power of each resource block is the same. Thus, we change the restriction of the $P3$ as

$$s.t. P_{kb} = \frac{P_b}{N} \tag{31}$$

5. Physical Network Optimization

In the previous section, the algorithm has given several feasible physical RBs for the MS to switch in. The goal of this section (the third step in Figure 4) is to propose a method to finally choose one particular network among the given choices based on the analysis of penalty function.

We propose a new DP-DQN algorithm based on deep reinforcement learning in this section to solve the optimization problem—select the appropriate access physical channel to minimize the overall network cost (i.e., from the perspective of core network overhead instead of single MS).

It is worth mentioning that the training method and use of DQN used in this study are slightly different from the conventional situation. In this study, the training of DQN is located in the controller side of the core network, while the actual deployment of the model is located in the MS side. This design is because we hope to achieve an equivalent feedback loop through core network controller training and marginal terminal deployment, so that the mobile terminal can consider the access selection problem from the perspective of core network overhead.

After the training is completed, DQN will be deployed to the mobile terminal. Here, we draw on the idea of federated learning by deploying the edge decision model while absorbing the real-time data of each mobile station for final training to improve the accuracy of the model and the overall efficiency of the network. So we call it DP-DQN, or Distributed-Deployed DQN.

As you can see in Figure 5, during the training process, we will simulate some of the possible access request from the MS to the network. Then, the cost of certain actions will be calculated based on the designed formula to estimate the overhead of this allocation decision made on the network.

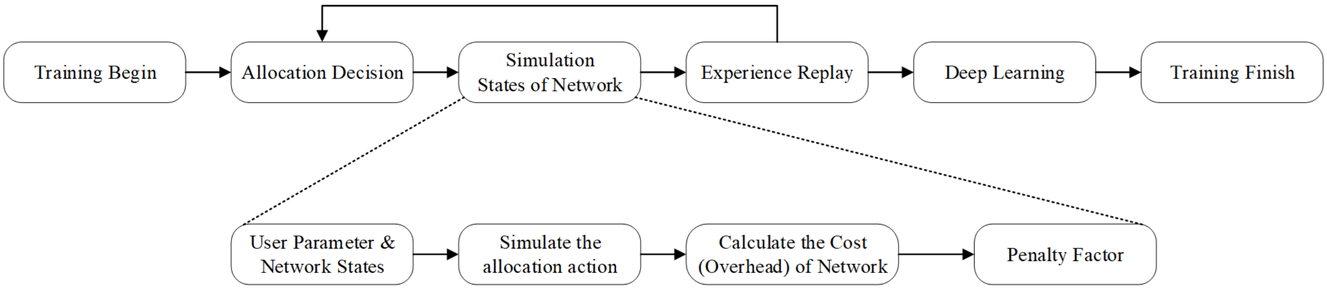

Training Begin
Allocation Decision
Simulation States of Network
Experience Replay
Deep Learning
Training Finish
User Parameter & Network States
Simulate the allocation action
Calculate the Cost (Overhead) of Network
Penalty Factor

Figure 5. Proposed DP-DQN pre-training process at the side of the core network controller.

We define the actions State Space and Action Space here. Instead of the Reward Function in the previous section, we define a Penalty Function to put the most emphasis on the action of the MS. In other words, if the MS makes a request which will put more burden on the core network, penalty will be given to the MS to make sure the overhead of the network is being kept at a relatively low level.

5.1. State Space

The state space of the proposed model is the current crowd situation of the network which is sent in the form of periodic packages to the mobile devices. The definition of this part is similar to that of Q-learning. The difference is that here, we focus more on the overhead of the network as a state factor.

$$\mathbf{S}(\mathbf{t}) = \mathbf{A}(t), k \in K, b \in B \tag{32}$$

Here, in addition to slicing state information similar to traditional Q-learning and DQN, we introduce some real-time user state information in order to better allocate network resources (e.x. user location, transmitting power, user data amount, etc.). Thus, the state space can be summarized as follows:

$$\mathbf{S}(\mathbf{t}) = [\mathbf{A}(t), \mathbf{P}_{kb}(t), \mathbf{D}_k(t), \mathbf{R'}_k(t)], k \in K, b \in B \tag{33}$$

where $\mathbf{D}_k(t) = \{\mathbf{L_c}(t), \mathbf{L_u}(t), \rho(t)\}$, which is the location status parameter of the mobile device and represents the shortest distance from the mobile station to the cellular and UAV network access point and MS density in certain areas, respectively, collected and analyzed by the satellite navigation module.

$$L_{c,2,1}(t) = \sqrt{d_{2,1}(t)^2 + (h_2(t) - h_1(t))^2} \tag{34}$$

$$\rho(t) = \frac{card(U(t))}{\pi R^2} \tag{35}$$

The parameters in the above two formulas can be obtained in the following ways. The position relationships between mobile stations are shown in Figure 6.

$$d_{2,1}(t) = 2rarcsin\sqrt{sin^2\frac{\Delta\phi_{2,1}}{2} + cos\phi_1 cos\phi_2 sin^2\frac{\Delta\theta_{2,1}}{2}} \tag{36}$$

$$\Delta\phi_{2,1} = \phi_2 - \phi_1 \tag{37}$$

$$\Delta\theta_{2,1} = \theta_2 - \theta_1 \tag{38}$$

$$U(t) = \{(x, y) | \sqrt{(x_i - x)^2 + (y_i - y)^2} < R\} \tag{39}$$

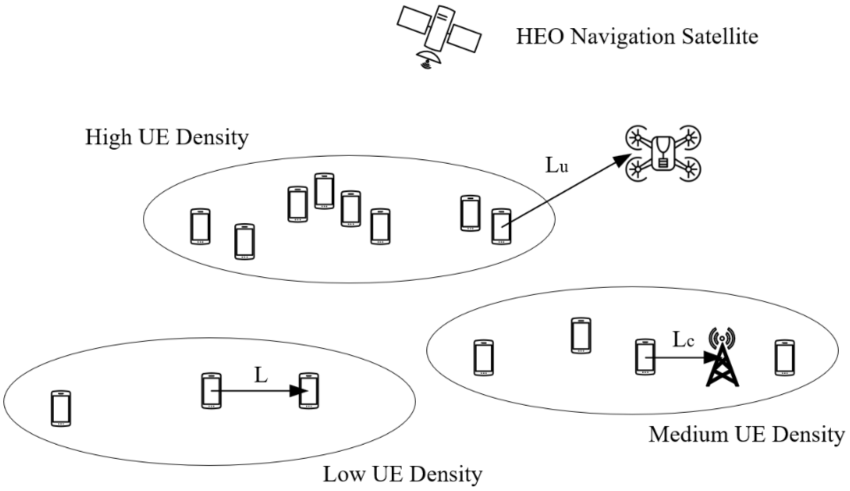


Figure 6. Schematic diagram of user density per unit area.

5.2. Action Space

The action space is defined as the allocation of slices for the new access mobile devices which can be divided into two steps:

- Give the slice choice $\mathbf{B}_{kb}$ which indicates that the MS will select which specific slicing network to access, and also select several specific slicing resources in this network

$$\mathbf{a(t)} = \mathbf{B}_{kb}, k \in K, b \in B. \tag{40}$$

- Renew the slice state from $\mathbf{A}_{kb}(t)$ to $\mathbf{A}_{kb}(t+)$.

For example, the current number of $\mathbf{A}_{i,j}(t)$ is 0 which means it is available now; the action may allocate it to the new MS and set it to 1.

5.3. Penalty Function

The definition of a penalty function is to make the model trying best to achieve the highest efficiency at certain circumstance. In other word, the decision will be made by the MS from the perspective of releasing the overhead of core network.

In order to make the difference between the slices of different maps more obvious (e.x. satellite, cellular or low altitude drone network), we define the penalty function target-that is, we define the penalty function separately for the three types of networks.

- Penalty Function of Satellite Network

$$\begin{aligned}\mathbf{PW}_1(t) &= -log(G'(\mathbf{DA},\ \mathbf{R}', \mathbf{L_c}, \mathbf{L_u}, \mathbf{P}_{kb}(t)\ )) \\ &= -log(\mathbf{DA} + \lambda_{11} * \mathbf{R}' + \lambda_{12} * \mathbf{L_c} + \lambda_{13} * \mathbf{L_u} + \lambda_{14} * \mathbf{P}_{kb}(t))\end{aligned} \tag{41}$$

- Penalty Function of Cellular Network

$$\begin{aligned}\mathbf{PW}_2(t) &= -log(G'(\mathbf{DA},\ \mathbf{R}', \mathbf{L_c}, \mathbf{P}_{kb}(t)\ )) \\ &= -log(\mathbf{DA} + \lambda_{21} * \mathbf{R}' + \lambda_{22} * \mathbf{L_c} + \lambda_{23} * \mathbf{P}_{kb}(t))\end{aligned} \tag{42}$$

- Penalty Function of UAV Network

$$\begin{aligned}\mathbf{PW}_3(t) &= -log(G'(\mathbf{DA},\ \mathbf{R}', \mathbf{L_u}, \mathbf{P}_{kb}(t)\ )) \\ &= -log(\mathbf{DA} + \lambda_{31} * \mathbf{R}' + \lambda_{32} * \mathbf{L_u} + \lambda_{33} * \mathbf{P}_{kb}(t))\end{aligned} \tag{43}$$

If we combine these three cases into one formula, we get the following more general case.

$$\begin{aligned}\mathbf{PW}(t) &= -log(G'(\mathbf{T},\ \mathbf{R}', \mathbf{S(t)}, \mathbf{P}_{kb}(t)\ )) \\ &= -log(\mathbf{DA} + \lambda_1 * \mathbf{R}' + c_1 * \lambda_2 * \mathbf{L_c} + c_2 * \lambda_3 * \mathbf{L_u} + \lambda_4 * \mathbf{P}_{kb}(t))\end{aligned} \tag{44}$$

Here, the parameter **DA** and **P** has been determined by the previous sections, while $\lambda_i \in [0,1]$ is a parameter aimed to describe the sensitiveness of the system. Moreover, $C_i \in \{0,1\}$ is the network label of the system, which is used to indicate the difference in the penalty function corresponding to different physical networks (i.e., cellular network or satellite network).

It is worth mentioning that in the process of training, the system dynamically adjusts the value of sensitive factors according to the current network state and other information. This is an optimization problem with only four finite variables, so it can be quickly solved by existing optimization algorithms or grid search methods.

6. Numerical Results

In this section, we build a simulation of the architecture we proposed in MATLAB (Version R2020a) and Python (Version 3.10) to experimentally simulate the real allocation decision circumstance at the MS side depending on the model deployed by the controller of the core network so as to prove the proposed better performance. The result will be given about the access ratio as well as the cost of the calculation resources of the devices in the form of the CPU usage.

6.1. Simulation Procedure

The architecture of the whole experimental approach to our ideas in this study is shown in Figure 7.

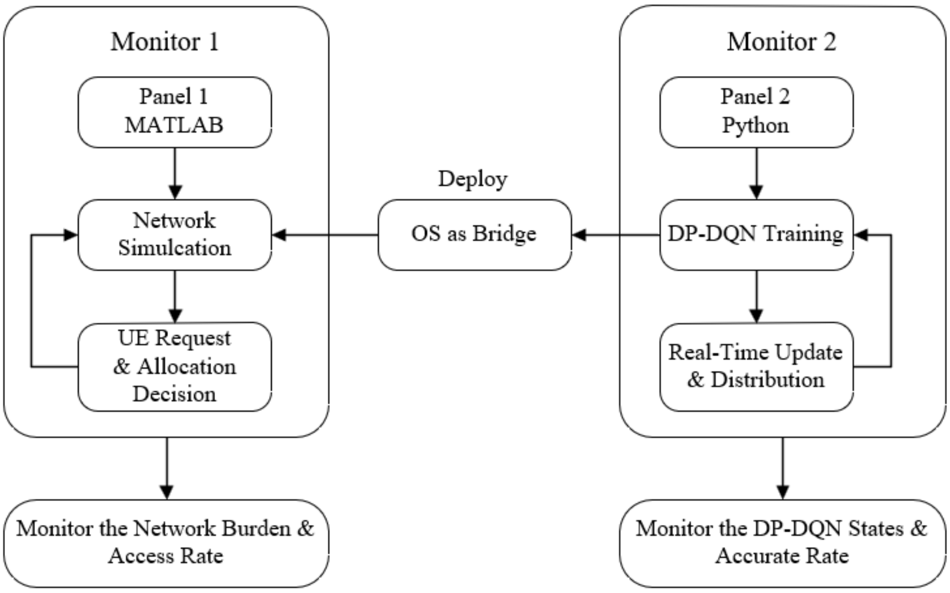


Figure 7. Simulation flow diagram: the DP-DQN model is trained in python and transferred to MATLAB with the operating system as a bridge to simulate the process of decision model delivery.

We first define MS objects programmatically in MATLAB, and randomly generate its arrival time, transmit power, packet size and other parameters for each object. Then, we use python to conduct the DP-DQN reinforcement learning training proposed by us. After obtaining the model, the operating system of PC is used as a bridge to pass the trained model into the MATLAB function. This step also simulates the distributed deployment process of the model trained by the core network to the user's mobile station.

In the simulation, we assume that the maximum amount of users over a 1 h duration is 3000, among which 1500 users are asking for a connection to send messages while the other 1500 users are asking for slices to receive messages. The max information package of one request will not exceed 20 GB and will not be below 2 MB. At the same time, each message package will be given an emergence level randomly, which is a positive integer between 1 and 3. We will define three types of service types which have different data rates and latencies. Every user will have a reference service type. To make the simulation more universal, all of the parameters listed above will be given randomly in the simulation process. Meanwhile, the requests from the users have different arrival times, which will be randomly generated in the loop. And, in our research, we assume that the arrival time of each user obeys the Poisson process.

Also, to take the location and density of the MS, we will also arrange location (longitude, latitude and altitude) for each MS (i.e., user) randomly. To simulate the scenario of metropolises with rural areas. The generation of the location will not be absolutely random; most of the users' locations will be around the main city which is set as 4 in our work and several users will be around the small towns, which means in these kind of area the access request will not be as much as those in metropolises.

In the table, the element SF indicates Sensitive Factor. Moreover, we will define three types of physical channels (e.g., terrain cellular, satellite and drone network). Each channel will be cut into three slices based on the total service types. Then, the slices will also contain 1800 or less different sub-carrier. The minimum particle have a maximum transmitting rate of 3 MBps. In addition, λ_i in the table is the sensitive factor of each input parameter, which is set as variable (non-fixed value), and experiments are conducted on different λ_i. Here, we use a grid search algorithm with a step size of 0.1 to determine the most appropriate value in a certain state, shown in Table 3.

Table 3. Summary of simulation parameter settings.

Parameter	Description	Value
$\mathbf{M}$	Total MS amount.	3000
$\mathbf{M}_U$	Total MS uplink request amount.	1500
$\mathbf{M}_D$	Total MS downlink request amount.	1500
$\mathbf{R}_n$	Total link type amount.	3
$\mathbf{T}_n$	Total service type amount.	3
$\mathbf{S}_m$	Minimum data rate of one RB.	3 MBps
$\mathbf{RB}_m$	Maximum RB amount of one slice.	1600
$\mathbf{U}_m$	Maximum data amount of one request.	2 GBps
λ_1	SF of MS resources request.	alterable
λ_2	SF of MS transmitting power.	alterable
λ_3	SF of MS distance to base station.	alterable
λ_4	SF of MS distance to UAV devices.	alterable
$\lambda_{poission}$	Minimum data rate of one RB.	alterable
$\mathbf{SM}_T$	Total simulation span.	1 hour

All of the slices which have been allocated to users will be recorded in an array until the process of transmitting messages has been finished.

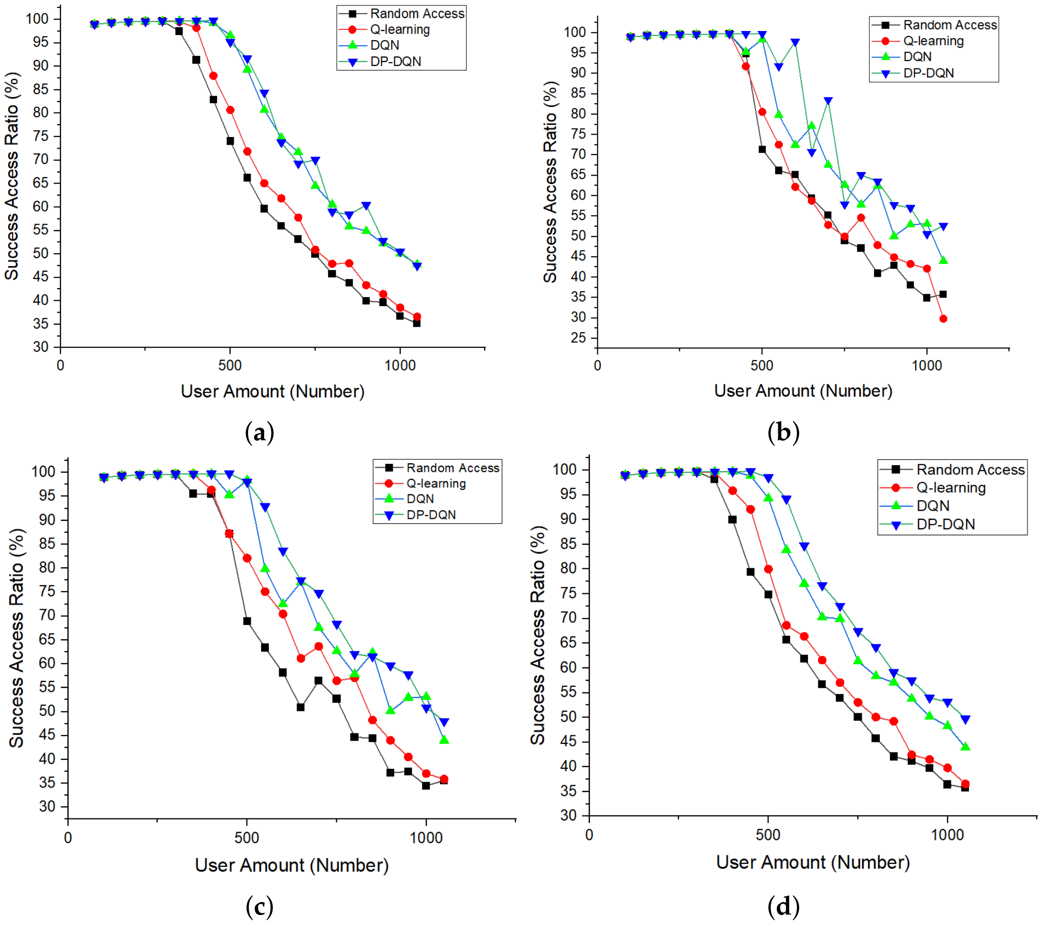
6.2. Network Access Ratio Performance

As we mentioned in the previous chapters, we introduced the concept of simulation time in the design of the system simulation model, which is the generation of random numbers for the user request issue time. In other words, when the simulation duration is determined, changing the number of access users is equivalent to changing the user access request per unit time, that is, the number of user access densities in the time domain.

In order to better show the difference between decision-making methods, we selected four groups of data samples as comparative experiments towards the proposed method. Also, we set different SFs of the penalty function during the simulation, considering the existence of multiple parameters that may have different impacts on the system. Because there are not many combinations of SFs, they can be easily found by periodic grid search in practical systems. Here, we chose four typical sets of SFs for presentation summarized in Table 4.

Table 4. Summary of SF setting in Figure 8.

Parameter	Value (a)	Value (b)	Value (c)	Value (d)
λ_1	0.50	0.50	0.70	0.50
λ_2	0.50	0.50	0.80	0.80
λ_3	0.50	0.80	−0.80	−0.60
λ_4	0.50	0.50	0.50	−0.30



(**a**) (**b**)

(**c**) (**d**)

Figure 8. MS successful access ratio of proposed slicing method with different user amounts in the circumstances of different sensitive factors summarized in Table 4. Since the differences of sensitivity factors in (**a**–**c**) are relatively small, the impact of different parameters on the system cannot be reflected, so its effect is slightly lower than that of (**d**).

As shown in Figure 8, the method without any advanced resource allocation called Random Access here has the lowest access ratio. Also, the basic access resource allocation algorithm powered by Q-learning is shown, that is, the so-called "Channel Cost" method proposed by many scholars in the past. It can be clearly seen that our proposed DP-DQN method has great advantages over other methods, especially when the unit time access volume gradually increases. It is worth mentioning that in this group of experiments, while changing the number of users making requests per unit time, the number of resources available for deployment per unit time in the control network remains unchanged, and the arrival time of users is subject to the same random process (Poisson process).

Secondly, we fixed the number of users unchanged, changed the number of resources in the frequency domain (i.e., the amount of total resource blocks) and adopted the SF pairs that worked best in the previous experiment. It can be seen that our proposed resource allocation method has a higher access ratio than the original methods, shown in Figure 9.

Compared with the experimental group, it is still consistent with the situation just now, and it can be seen that the method we proposed has obvious advantages in the number of resources from 600 to 1000. Especially when the number of resources is relatively short compared to the number of access users, our decision-making method can see a rapid increase in the access success rate, which shows that the method has excellent robustness in the face of sudden large access.

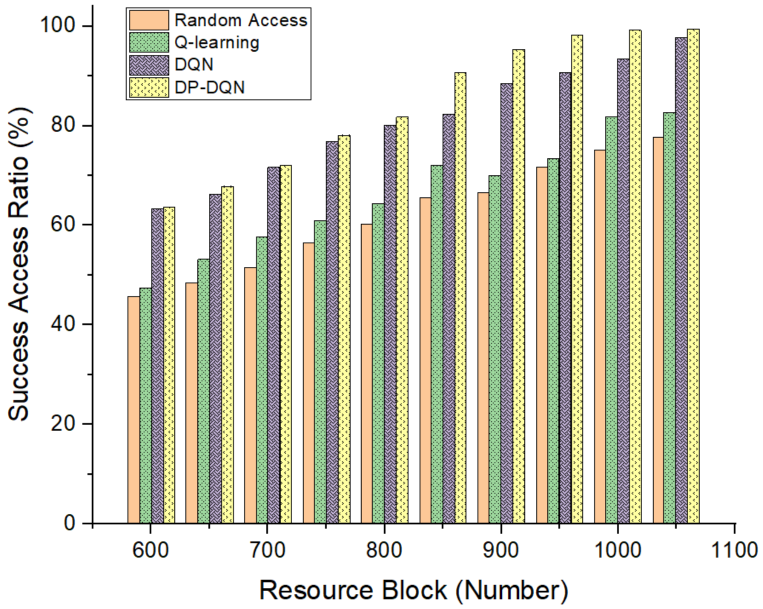


Figure 9. MS successful access ratio of proposed slicing allocation method with different RB amounts.

6.3. Network Overhead Performance

Finally, to further verify the superiority of our proposed method, we define a network overhead function in two aspects (i.e., user panel and control panel). Figure 10 shows the corresponding network overhead of these two panels for different user access quantities. It can be seen that with the same user access volume, the proposed method causes the least network overhead.

Network user panel overhead, that is, the burden brought by the data service carried in the network, which can be measured by the data flow in the network, as shown in the following formula.

$$Bd_u = E(\frac{RB_{used}}{RB_{total}}) = E(\frac{\sum_{k\in M} B_{kb}}{C_{ch}}) \tag{45}$$

The control surface overhead is defined by the transmission volume of the signaling, as shown in the following formula.

$$I \leq -log(\frac{1}{L}) \tag{46}$$

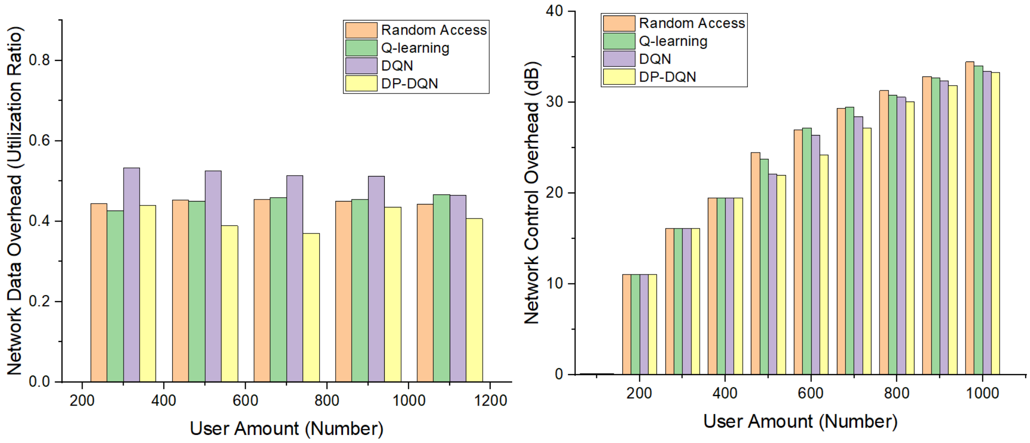
$$Id_{max} = -log(\frac{1}{L})M \tag{47}$$

$$I_{actual} \leq \sum_{i=0}^{\infty} -log(\frac{1}{L_i})MAS^i = -log(\frac{1}{L})M\sum_{i=0}^{\infty} AS^i \tag{48}$$

$$Bd_c = 10log(\frac{I_{actual}}{Id_{max}}) \tag{49}$$

where L indicates the total signaling type in order to describe the cost of a single message on the control panel.

By the way, since in this study we tried to design a way that can marginalize the decision-making process to reduce the overhead on the core network controller, which may challenge the computing power of the mobile station, we also conducted experiments on the CPU computing load in experiments and simulations. Table 5 shows the CPU and cache usage rate during the simulation loop. The results show that the average mobile device is fully capable of running the model generated in our proposed way, which is lower than what many other edge computing-related studies require for mobile devices [46–50].



(**a**) (**b**)

Figure 10. (**a**) User panel overhead of proposed slicing method with different user amounts. (**b**) Control panel overhead of proposed slicing method with different user amounts. In the figure, we compare our proposed approach to the current mainstream solution, which can better reduce the network overhead with the same amount of resources and users.

Table 5. Summary of CPU parameters during simulation.

Parameter	Average Value
CPU utilization	0.62
CPU base speed	2.80 GHz
CPU speed	3.62 GHz
L1 D-cache	320 KB
L2 D-cache	5.0 MB
L3 D-cache	12.0 MB

7. Discussions and Conclusions

The proposed integrated space–ground communication system architecture represents a significant advancement in addressing the challenges of future communication networks, particularly in managing user surge and reducing network overhead. By combining terrestrial cellular networks, drone networks and satellite networks with slicing technology, our architecture offers a versatile solution that enhances network efficiency and reliability.

This study introduces a novel marginal MS-assisted network resource allocation architecture. Leveraging cellular network slicing technology, it dynamically manages user demands and network conditions. Integration of the DP-DQN model enhances real-time resource allocation decisions for MS, optimizing performance. Furthermore, our proposed feedback method ensures the accuracy and adaptability of the marginalization model, crucial for maintaining optimal network performance in dynamic environments. Through extensive simulations, our DP-DQN-based edge decision method demonstrates superior performance in reducing core network overhead and improving access success rates compared to conventional approaches.

As for the analysis of the DP-DQN model in the numerical results section, we found that taking the location information of mobile devices into account significantly improved the resource utilization of each network. Also, in terms of network slicing, including different physical channels into the type of network slices also helps to improve the degree of integration of the system. At the same time, using the new distributed MS-assisted resource decision model, the system overhead (especially the control panel) is greatly reduced, which gives us reason to believe that in the future communication system, compared with the traditional centralized control, distributed decision-making will have a better prospect.

This study identifies promising future research directions for space–ground integrated systems. Our architecture's scalability and adaptability extend its applicability beyond traditional networks to smart cities, disaster response, and remote sensing. As demand for seamless connectivity grows, our approach lays a robust foundation for efficient and resilient communication infrastructures. This research advances communication systems by innovatively addressing network resource allocation and decision-making, bridging terrestrial and satellite networks. Our next steps focus on enhancing system integration through improved methods for acquiring and utilizing MS navigation and remote sensing data.

Author Contributions: Conceptualization, T.Z.; Methodology, T.Z.; Software, T.Z.; Validation, T.Z.; Formal analysis, T.Z.; Investigation, T.Z.; Writing—original draft, T.Z.; Supervision, Z.L.; Project administration, T.Z.; Funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Chongqing, grant number CSTB2023NSCQ-LMX0033, and Science and Technology Research Program of Chongqing Municipal Education Commission, grant number KJQN202300646.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, W.; Lin, X.; Lee, J.; Toskala, A.; Sun, S.; Chiasserini, C.F.; Liu, L. 5G-advanced toward 6G: Past, present, and future. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1592–1619. [CrossRef]
2. He, D.; Guan, K.; Yan, D.; Yi, H.; Zhang, Z.; Wang, X.; Zorba, N. Physics and AI-based digital twin of multi-spectrum propagation characteristics for communication and sensing in 6G and beyond. *IEEE J. Sel. Areas Commun.* **2024**, *41*, 3461–3473. [CrossRef]
3. Liu, Y.; Peng, M.; Shou, G.; Chen, Y.; Chen, S. Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 6722–6747. [CrossRef]
4. Hassan, S.S.; Park, Y.M.; Tun, Y.K.; Saad, W.; Han, Z.; Hong, C.S. SpaceRIS: LEO Satellite Coverage Maximization in 6G Sub-THz Networks by MAPPO DRL and Whale Optimization. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 1262–1278. [CrossRef]
5. Zhong, A.; Li, Z.; Wu, D.; Tang, T.; Wang, R. Stochastic Peak Age of Information Guarantee for Cooperative Sensing in Internet of Everything. *IEEE Internet Things J.* **2023**, *10*, 15186–15196. [CrossRef]
6. Lin, Z.; Lin, M.; de Cola, T.; Wang, J.-B.; Zhu, W.-P.; Cheng, J. Supporting IoT with rate-splitting multiple access in satellite and aerial-integrated networks. *IEEE Internet Things J.* **2021**, *8*, 11123–11134. [CrossRef]
7. Shah, S.D.A.; Gregory, M.A.; Li, S. Toward Network Slicing Enabled Edge Computing: A Cloud-Native Approach for Slice Mobility. *IEEE Internet Things J.* **2024**, *11*, 2684–2700. [CrossRef]
8. Lin, Z.; Niu, H.; An, K.; Wang, Y.; Zheng, G.; Chatzinotas, S.; Hu, Y. Refracting RIS aided hybrid satellite-terrestrial relay networks: Joint beamforming design and optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3717–3724. [CrossRef]
9. Sharif, Z.; Jung, L.T.; Razzak, I.; Alazab, M. Adaptive and Priority-Based Resource Allocation for Efficient Resources Utilization in Mobile-Edge Computing. *IEEE Internet Things J.* **2023**, *10*, 3079–3093. [CrossRef]
10. Xue, Q.; Wei, R.; Li, Z.; Liu, Y.; Xu, Y.; Chen, Q. Beamforming Design for Cooperative Double-RIS Aided mmWave MU-MIMO Communications. *IEEE Trans. Green Commun. Netw.* **2024**. [CrossRef]
11. Li, Z.; Li, F.; Tang, T.; Zhang, H.; Yang, J. Video caching and scheduling with edge cooperation. *Digit. Commun. Netw.* **2024**, *10*, 450–460. [CrossRef]
12. Hoang, T.M.; Xu, C.; Vahid, A.; Tuan, H.D.; Duong, T.Q.; Hanzo, L. Secrecy-Rate Optimization of Double RIS-Aided Space–Ground Networks. *IEEE Internet Things J.* **2023**, *10*, 13221–13234. [CrossRef]
13. He, Y.; Liu, Y.; Jiang, C.; Zhong, X. Multiobjective anti-collision for massive access ranging in MF-TDMA satellite communication system. *IEEE Internet Things J.* **2022**, *9*, 14655–14666. [CrossRef]

14. Lin, Z.; Lin, M.; Champagne, B.; Zhu, W.-P.; Al-Dhahir, N. Secrecy-energy efficient hybrid beamforming for satellite-terrestrial integrated networks. *IEEE Trans. Commun.* **2021**, *69*, 6345–6360. [CrossRef]
15. Li, Z.; Zhu, N.; Wu, D.; Wang, H.; Wang, R. Energy-Efficient Mobile Edge Computing Under Delay Constraints. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 776–786. [CrossRef]
16. Alsenwi, M.; Tran, N.H.; Bennis, M.; Pandey, S.R.; Bairagi, A.K.; Hong, C.S. Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: A deep reinforcement learning based approach. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 4585–4600. [CrossRef]
17. Liu, Y.; Clerckx, B.; Popovski, P. Network slicing for eMBB, URLLC, and mMTC: An uplink rate-splitting multiple access approach. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 2140–2152. [CrossRef]
18. Wu, H.; He, M.; Shen, X.; Zhuang, W.; Dao, N.D.; Shi, W. Network Performance Analysis of Satellite-Terrestrial Vehicular Network. *IEEE Internet Things J.* **2024**, *11*, 16829–16844. [CrossRef]
19. Yan, Z.; Li, D. Convergence Time Optimization for Decentralized Federated Learning with LEO Satellites via Number Control. *IEEE Trans. Veh. Technol.* **2024**, *73*, 4517–4522. [CrossRef]
20. Zangooei, M.; Golkarifard, M.; Rouili, M.; Saha, N.; Boutaba, R. Flexible RAN Slicing in Open RAN with Constrained Multi-agent Reinforcement Learning. *IEEE J. Sel. Areas Commun.* **2023**, *42*, 280–294. [CrossRef]
21. Li, H.; Kong, Z.; Chen, Y.; Wang, L.; Lu, Z.; Wen, X.; Xiang, W. Slice-based service function chain embedding for end-to-end network slice deployment. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 3652–3672. [CrossRef]
22. Javadpour, A.; Ja'fari, F.; Taleb, T.; Benzaïd, C. Enhancing 5G Network Slicing: Slice Isolation Via Actor-Critic Reinforcement Learning with Optimal Graph Features. In Proceedings of the GLOBECOM 2023–2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 4–8 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 31–37.
23. Sun, Y.; Shi, Z.; Wang, J.; Liu, J. MADRL-Enhanced Secure RAN Slicing in 5G and Beyond Multi-Cell Uplink Communication Systems. In Proceedings of the GLOBECOM 2023–2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 4–8 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 2439–2444.
24. Guo, C.; Hu, G.; Pan, C.; Li, F.; Xu, H.; Han, Z. Authentication for Satellite Internet Resource Slicing Access Based on Trust Measurement. *IEEE Internet Things J.* **2024**, *11*, 21788–21806. [CrossRef]
25. Liu, Y.; Ma, T.; Tang, Z.; Qin, X.; Zhou, H.; Shen, X. Ultra-Dense LEO Satellite Access Network Slicing: A Deep Reinforcement Learning Approach. In Proceedings of the GLOBECOM 2023–2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 4–8 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 5043–5048.
26. Kim, T.; Kwak, J.; Choi, J.P. Satellite edge computing architecture and network slice scheduling for IoT support. *IEEE Internet Things J.* **2021**, *9*, 14938–14951. [CrossRef]
27. De Cola, T.; Bisio, I. QoS optimisation of eMBB services in converged 5G-satellite networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12098–12110. [CrossRef]
28. Chen, G.; Qi, S.; Shen, F.; Zeng, Q.; Zhang, Y.D. Information-Aware Driven Dynamic LEO-RAN Slicing Algorithm Joint with Communication, Computing and Caching. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 1044–1062. [CrossRef]
29. Li, X.; Chang, H.; Wang, X.; Li, S.; Zhou, Y.; Yu, H. An Optimization-Based Tightly-Coupled Integration of PPP, INS and Vision for Precise and Continuous Navigation. *IEEE Trans. Veh. Technol.* **2024**, *73*, 4934–4948. [CrossRef]
30. Deng, L.; Yang, Y.; Ma, J.; Feng, Y.; Ye, L.; Li, H. OFDM-BOC: A Broadband Multicarrier Navigation Modulation-Based BOC for Future GNSS. *IEEE Trans. Veh. Technol.* **2024**, *73*, 3964–3979. [CrossRef]
31. Wang, J.; Chen, X.; Shi, C.; Liu, J. Robust M-estimation-Based ICKF for GNSS Outlier Mitigation in GNSS/SINS navigation applications. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–17. [CrossRef]
32. Sirikonda, S.; Parayitam, L.; Hablani, H.B. Tightly Coupled NavIC and Low-Cost Sensors for Ground Vehicle Navigation. *IEEE Sens. J.* **2024**, *24*, 14977–14991. [CrossRef]
33. Shen, K.; Li, Y.; Liu, T.; Zuo, J.; Yang, Z. Adaptive-Robust Fusion Strategy for Autonomous Navigation in GNSS-Challenged Environments. *IEEE Internet Things J.* **2023**, *11*, 6817–6832. [CrossRef]
34. Yu, Y.; Dutkiewicz, E.; Huang, X.; Mueck, M. Downlink resource allocation for next generation wireless networks with inter-cell interference. *IEEE Trans. Wirel. Commun.* **2023**, *12*, 1783–1793.
35. Halabian, H. Distributed resource allocation optimization in 5G virtualized networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 627–642. [CrossRef]
36. Yemini, M.; Goldsmith, A.J. Virtual cell clustering with optimal resource allocation to maximize capacity. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 5099–5114. [CrossRef]
37. Wang, W.; Tang, L.; Liu, T.; He, X.; Liang, C.; Chen, Q. Towards Reliability-Enhanced, Delay-Guaranteed Dynamic Network Slicing: A Multi-Agent DQN Approach with An Action Space Reduction Strategy. *IEEE Internet Things J.* **2023**, *11*, 9282–9297. [CrossRef]
38. Ji, Y.; Wang, Y.; Zhao, H.; Gui, G.; Gacanin, H.; Sari, H.; Adachi, F. Multi-Agent Reinforcement Learning Resources Allocation Method Using Dueling Double Deep Q-Network in Vehicular Networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 13447–13460. [CrossRef]
39. Xiong, X.; Zheng, K.; Lei, L.; Hou, L. Resource allocation based on deep reinforcement learning in IoT edge computing. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1133–1146. [CrossRef]

40. Hu, H.; Wu, D.; Zhou, F.; Zhu, X.; Hu, R.Q.; Zhu, H. Intelligent resource allocation for edge-cloud collaborative networks: A hybrid DDPG-D3QN approach. *IEEE Trans. Veh. Technol.* **2023**, *72*, 10696–10709. [CrossRef]
41. Guan, Y.; Zou, S.; Peng, H.; Ni, W.; Sun, Y.; Gao, H. Cooperative UAV trajectory design for disaster area emergency communications: A multi-agent PPO method. *IEEE Internet Things J.* **2023**, *11*, 8848–8859. [CrossRef]
42. Guo, J.; Zhou, H.; Zhao, L.; Chang, W.; Jiang, T. Incentive-driven and SAC-based Resource Allocation and Offloading Strategy in Vehicular Edge Computing Networks. In Proceedings of the IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA, 20 May 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
43. Ramesh, D.; Sanampudi, S.K. An automated essay scoring systems: A systematic literature review. *Artif. Intell. Rev.* **2022**, *55*, 2495–2527. [CrossRef] [PubMed]
44. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef] [PubMed]
45. Shen, M.; Zhang, J.; Zhu, L.; Xu, K.; Du, X. Accurate decentralized application identification via encrypted traffic analysis using graph neural networks. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 2367–2380. [CrossRef]
46. Li, Z.; Gao, X.; Li, Q.; Guo, J.; Yang, B. Edge Caching Enhancement for Industrial Internet: A Recommendation-Aided Approach. *IEEE Internet Things J.* **2022**, *9*, 16941–16952. [CrossRef]
47. Ning, Z.; Yang, Y.; Wang, X.; Guo, L.; Gao, X.; Guo, S.; Wang, G. Dynamic computation offloading and server deployment for UAV-enabled multi-access edge computing. *IEEE Trans. Mob. Comput.* **2021**, *22*, 2628–2644. [CrossRef]
48. Tang, T.; Yin, Z.; Li, J.; Wang, H.; Wu, D.; Wang, R. End-to-End Distortion Modeling for Error-Resilient Screen Content Video Coding. *IEEE Trans. Multimed.* **2024**, *26*, 4458–4468. [CrossRef]
49. Imteaj, A.; Thakker, U.; Wang, S.; Li, J.; Amini, M.H. A Survey on Federated Learning for Resource-Constrained IoT Devices. *IEEE Internet Things J.* **2022**, *9*, 1–24. [CrossRef]
50. Zhou, H.; Jiang, K.; Liu, X.; Li, X.; Leung, V.C.M. Deep Reinforcement Learning for Energy-Efficient Computation Offloading in Mobile-Edge Computing. *IEEE Internet Things J.* **2022**, *9*, 1517–1530. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.