future internet

Article

Research on Task-Oriented Computation Offloading Decision in Space-Air-Ground Integrated Network

Jun Liu, Xiaohui Lian * and Chang Liu

School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China; liujun@cse.neu.edu.cn (J.L.); 1901667@stu.neu.edu.cn (C.L.)
* Correspondence: 1901661@stu.neu.edu.cn

Abstract: In Space–Air–Ground Integrated Networks (SAGIN), computation offloading technology is a new way to improve the processing efficiency of node tasks and improve the limitation of computing storage resources. To solve the problem of large delay and energy consumption cost of task computation offloading, which caused by the complex and variable network offloading environment and a large amount of offloading tasks, a computation offloading decision scheme based on Markov and Deep Q Networks (DQN) is proposed. First, we select the optimal offloading network based on the characteristics of the movement of the task offloading process in the network. Then, the task offloading process is transformed into a Markov state transition process to build a model of the computational offloading decision process. Finally, the delay and energy consumption weights are introduced into the DQN algorithm to update the computation offloading decision process, and the optimal offloading decision under the low cost is achieved according to the task attributes. The simulation results show that compared with the traditional Lyapunov-based offloading decision scheme and the classical Q-learning algorithm, the delay and energy consumption are respectively reduced by 68.33% and 11.21%, under equal weights when the offloading task volume exceeds 500 Mbit. Moreover, compared with offloading to edge nodes or backbone nodes of the network alone, the proposed mixed offloading model can satisfy more than 100 task requests with low energy consumption and low delay. It can be seen that the computation offloading decision proposed in this paper can effectively reduce the delay and energy consumption during the task computation offloading in the Space–Air–Ground Integrated Network environment, and can select the optimal offloading sites to execute the tasks according to the characteristics of the task itself.

Keywords: Space–Air–Ground Integrated Network; computation offloading; Markov decision process; DQN

check for updates

Citation: Liu, J.; Lian, X.; Liu, C. Research on Task-Oriented Computation Offloading Decision in Space-Air-Ground Integrated Network. *Future Internet* **2021**, *13*, 128. https://doi.org/10.3390/fi13050128

Academic Editor: Paolo Bellavista

Received: 10 April 2021
Accepted: 10 May 2021
Published: 13 May 2021

1. Introduction

Space–Air–Ground Integrated Network (SAGIN) is a large-scale and complex heterogeneous network system formed by the interconnection of space-based information network, internet, and mobile communication network, which combines satellite network with terrestrial network closely. Further, it collects, transmits, and processes space information in real-time to meet the demand for communication services with global coverage. The space–Air–Ground Integrated Network is a strategic and fundamental critical infrastructure network, which can drive the development of China's emerging industries. In the field of military and civilian information applications and Internet of everything era, it plays a crucial role to deal with the demand for massive information services and will be the inevitable trend of the future development of the network [1]. The space-ground integrated network is a large complex hierarchical heterogeneous network, which mainly consists of three parts: space-based backbone network, space-based edge access network and ground-based backbone network. As shown in Figure 1, the space-based network consists of satellites and satellite constellations as well as the corresponding ground infrastructure such as ground stations and network operation control centers. These satellites are in different orbits, and according to the altitude, the satellites can be divided into Geostationary Orbit (GEO) satellites, Medium Earth Orbit (MEO) satellites, and Low Earth Orbit (LEO) satellites [2]. Among them, the space-based backbone network is the core infrastructure of the integrated space–ground information network, using satellites of different levels at each orbital level to unite with each other to provide information transmission and processing industry for the whole space information system. Space-based edge access network contains various types of low-altitude vehicles, forming a single-layer constellation, or dual constellation of low and medium orbit, information transmission through satellite links, which can provide communication services to cover information in the beam center gateway control station and various user terminals. The ground-based network includes five theaters on the ground and various types of ground nodes (such as cell phones, submarines, etc.), mainly consisting of cellular networks, mobile ad hoc network (MANET) [3], Worldwide Interoperability for Microwave Access (WiMAX), Wireless Local Area Network (WLAN) and other ground-based communication systems. Among them, the ground-based backbone network is the ground infrastructure of the integrated space–ground information network, connecting to the ground control station network and communication gateway stations, thus forming a kind of information transmission method for integrated scheduling of heaven and earth resources.

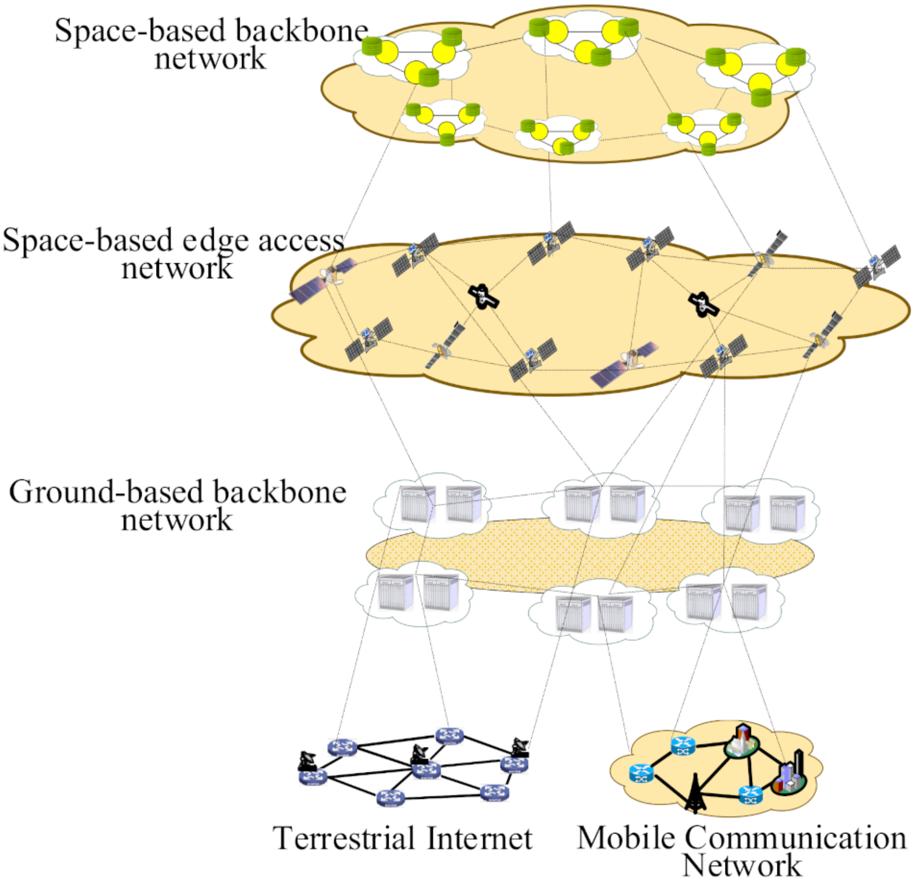
Space-based backbone network
Space-based edge access network
Ground-based backbone network
Terrestrial Internet
Mobile Communication Network

Figure 1. Architecture of the Space–Earth Integration Network. The structure of the Space–Air–Ground Integrated Network is given from the perspective of network nodes, and the space-based backbone network is the core infrastructure of the space–ground integrated information network to meet the needs of various types of communications through the integration of space and ground networks.

The GEO satellites have the characteristics of high coverage, high transmission rate, and dynamic adaptation, but there are limitations of orbit fixation and high processing task delay, while the low-medium orbits have the advantages of wide distribution and short distance, and the computing storage ability of ground base stations is strong. Using air, space, and ground resources to work collaboratively can achieve multi-dimensional strategic information services. However, due to the dynamic change of the Space–Air–Ground Integrated Network topology, the number of tasks to be processed is huge and the number of satellites is limited, so there are many problems such as limited computing resources abilities of network nodes, and untimely task processing, resulting in some important task processing to fail [4]. How to make the network nodes in different orbits cooperate closely and collaborate in-depth to enable complex or large-scale tasks to be processed efficiently, is an urgent problem should be solved at present.

At present, computation offloading technology [5] is widely used in Internet of things and onboard network. It has become an important means of real-time processing and lasting endurance of terminal business, mainly to solve the problems of lack of computing capacity and limited storage resources of computing nodes. Computation offloading is the key technology in Mobile Edge Computing (MEC), which can be executed by offloading the business to other network sites with plentiful computational resources and powerful computing capabilities, and then after the tasks are completed, the results are passed back to the offloading terminal devices, thus saving the cost of the offloading terminal devices and reducing the limitation of limited resources in the mobile system [6]. Therefore, in the Space–Air–Ground Integrated Network, the computation offloading technology can be used to decompose the compute-intensive tasks. Selecting any computing node in the network that can take on the task offloading as the offloading site to perform tasks can effectively alleviate the computational storage pressure of network nodes and improve the efficiency of huge and complex tasks. The general process for task computation offloading in the Space–Air–Ground Integrated Network is [7]: (1) Analysis whether any terminal located in the space-based backbone network, the edge access network or the ground-based backbone network requires task computation offloading. (2) Get the resources of the network. (3) Analyze the state of the wireless channel, and determine whether the task is executed locally or needs to be offloaded to other service devices for execution by the computation offloading decision. (4) According to the decision instructions, the tasks that need to be offloaded are split into several separate subtasks, the subtasks that do not support offload are executed by the local terminal, and the subtasks that can be offloaded are offloaded to other network nodes in the Space–Air–Ground Integrated Network.

The design of computation offloading mainly includes offloading decisions and optimization of algorithms [8]. We consider the different types of computing nodes capabilities of the space-based backbone network, the edge access network and the ground-based backbone network in the Space–Air–Ground Integrated Network, and in-depth study of computation offloading decisions and algorithms for optimization decisions that are suitable for the integrated network environment, in order to improve the execution efficiency of huge and complex tasks in complex network environments.

2. Materials and Methods

2.1. Related Research

In recent years, computation offloading decisions [9] which aim at reducing delay and energy consumption have become one of the hot topics of computation offloading technology research, mainly considering whether user terminals need to be offloaded, the number of offloads, and the choice of offloading location. Zhang et al. [10] combined the auction theory to transform the computation offloading problem into a matching problem, and proposed a multi-round sealed sequence combined auction mechanism to reduce the delay in the computation offloading process. Kao and Krishnamachari [11] minimized the application delay under the constraint of resource utilization, at the same time transformed computation offloading into NP-hard problems, and proposed FPTAS online learning algorithm to solve the problem of computation offloading delay in the dynamic environment. Kamoun et al. [12] proposed a pre-computed offline solution to optimize the computation offloading process so that the terminal energy consumption could be minimized.

Zhao et al. [13] proposed an optional offloading scheme based on the lightweight request and access framework to reduce the energy consumption of devices. In reference [14], a computation offloading decision based on the Lyapunov function was proposed. However, considering the topological dynamic characteristics of the network, this method is not suitable for the wireless channel model with randomly changing. Du et al. [15] proposed a low-complexity suboptimal algorithm, which used semi-deterministic relaxation and randomization to obtain the optimal offloading decision to solve the computation offloading problems in mixed fog/cloud computing systems. Liu et al. [16] transformed the problem of computation offloading into the convex optimization problem, and proposed an integrated fog and cloud computation offloading strategy, which offloaded a series of applications to nearby fog nodes or cloud center and reduced the system delay. Ren et al. [17] proposed an optimal joint communication and computing resource allocation algorithm for the problem of resource allocation in computation offloading. Further, the method of segmentation optimized partial offloading model was used to segment the data to achieve the optimal resource matching under low delay. Chen et al. [18] proposed a mixed-integer nonlinear computing task offloading decision, which transformed the problem of low-delay computation offloading into the problem of task offloading placement and task resource allocation. This scheme effectively reduced the delay of the computation offloading process in ultra-intensive networks. Yu et al. [19] aimed to reduce the delay of the computation offloading process in the actual Internet of things and proposed a completely polynomial-time approximate computation offloading strategy to rationally allocate resources for applications and shorten the computation processing time. In Reference [20], a task priority algorithm of maximum energy saving was proposed, which divided the tasks that need to be offloaded and applied the greedy algorithm to reduce energy consumption in the computation offloading process. Chen et al. [21] considered the computation offloading problem of mobile edge servers with energy collection devices and proposed a task offloading strategy based on the Lyapunov algorithm and greedy scheduling algorithm. When the server received a task offloading request, a centralized and distributed resource scheduling strategy was adopted to reduce energy consumption. Pham et al. [22] equalized the delay and energy consumption cost, and used the dichotomy method to allocate computational resources to the mobile terminals that need to be offloaded, while maximizing the number of offload tasks and effectively reducing the total system cost. In Reference [23], the goal of minimizing delay and energy consumption was achieved by combining OREO online algorithm and the Lyapunov algorithm to dynamically optimize the server cache and the task computation offloading process in the unknown system state scenario. In Reference [24], a task offloading algorithm for distributed optimization of mobile edge computing was proposed to optimize the selection of offloading terminals and the transmission power of CPU frequency, to reduce system delay and energy consumption. Among the latest computation offloading techniques, Naouri et al. [25] proposed a three-layer task offloading framework named DCC applied to mobile terminals that distinguishes between types of application tasks and reduces delay by shunting tasks with higher computational requirements to the cloudlet and cloud layers, and placing low computational and high communication cost tasks to be executed on the device layer. Zhang et al. [26] investigated task offloading in virtual vehicle servers and proposed a fully distributed and infrastructure-less congestion avoidance and traffic optimization system for vehicular ad-hoc networks in urban environments, called DIFTOS, which divides the city map into a server with a hierarchical structure and has been demonstrated in simulations under different traffic conditions its effectiveness and scalability. Most of the existing computation offloading decision schemes have been applied in mobile edge computing scenarios for mobile terminals or in-vehicle networks, but without considering the time-varying factors of the network, the communication quality of the channel, and the tasks' attributes. For tasks of excessive magnitude, they are still unable to effectively reduce delay and energy consumption in the task offloading process.

The Space–Air–Ground Integrated Network has a complicated and changeable environment, and the amount of offloading tasks is huge so that the existing computation offloading decision schemes are not applicable. Considering the Space–Air–Ground Integrated Network structure, network channel states, task characteristics, and terminal devices performance, combining the Markov State Transition Principle [27,28] and neural networks, we propose a computation offloading decision method based on Markov and DQN. This method performs data segmentation processing on the task, and abstracts the computation offloading decision process of the task into the state transition process of Markov model, and then uses the improved DQN algorithm to update the model, and finally selects the most optimal offloading decision according to the attributes of the tasks and the performance of the offloading sites.

2.2. Computation Offloading Decision Process

In this section, combining with Markov Decision Process Principle, the tasks are offloaded to the ground-based backbone nodes, the edge access nodes, or the space-based backbone nodes according to their own locations and requirements. Figure 2 shows the model of the computation offloading decision process based on Markov Decision.

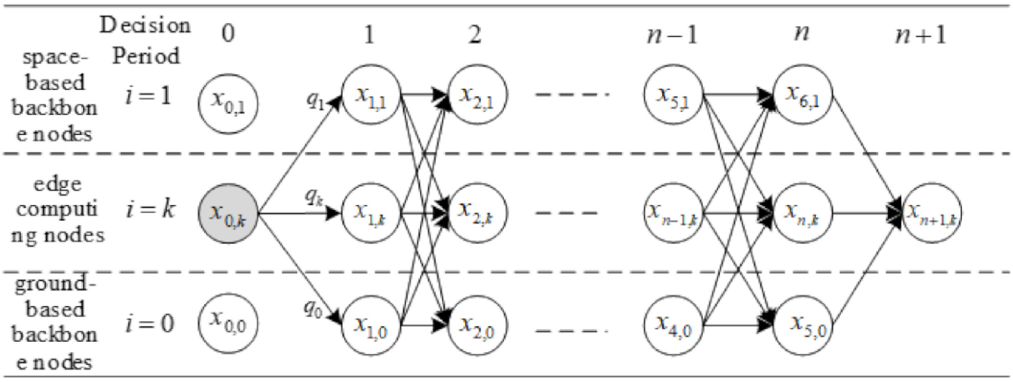

Figure 2. Markov State Transition Model. The Space–Air–Ground Integrated Network is abstracted into an offloading network model represented by Markov states, with i denoting the task offloading location and circles de-noting the Markov states in each decision period. Taking the edge access network in the middle layer as an example, the whole Markovian decision process is given, which is represented by gray circles.

The offloading terminals (offloader) of the Space–Air–Ground Integrated Network are the terminal devices at various positions in the network, and the offloading sites (offloadee) are offloaded sets that can provide a variety of computing nodes choices for tasks. The terminal devices in the network are in high-speed movement states. When processing high-intensity tasks, the signal strength of the wireless network coverage of the offloading sites will be weakened due to the prolonged processing time. Therefore, the huge and complex task is divided into several independent subtasks using the divide and conquer thinking, and the continuous task processing process is abstracted into the state transition process of the Markov model to reduce the series of effects caused by the dynamic change of the position of the terminal devices. Taking the offloading tasks (offloader tasks) in the edge access network as an example, the entire Markov decision process is constructed. Define V_{in} as the i-th task of the current terminal device n.

The location of the offloading terminals and the offloading execution terminals are not fixed. During the task offloading execution, the wireless channel performance from the terminals to the servers in the space-based backbone network, the ground-based backbone network, or edge access network is constantly changing. And the higher the transmission delay, the more difficult it is to guarantee the transmission quality of the wireless channel. Therefore, it is necessary to fully consider the factors that affect the network status during the task offloading execution. In the process of task offloading, to minimize the delay and energy consumption of task computation offloading, it is necessary to keep the current task offloading network as the most optimal network. When the terminal devices leave the scope of the offloading network which is currently executed by the offloading task, or a new offloading network that meets the offloading conditions is added, the original offloading network will change. At this time, switching through the network should be considered to complete the task offloading. Figure 3 shows the flow chart of network switching in the process of task computation offloading of the Space–Air–Ground Integrated Network.

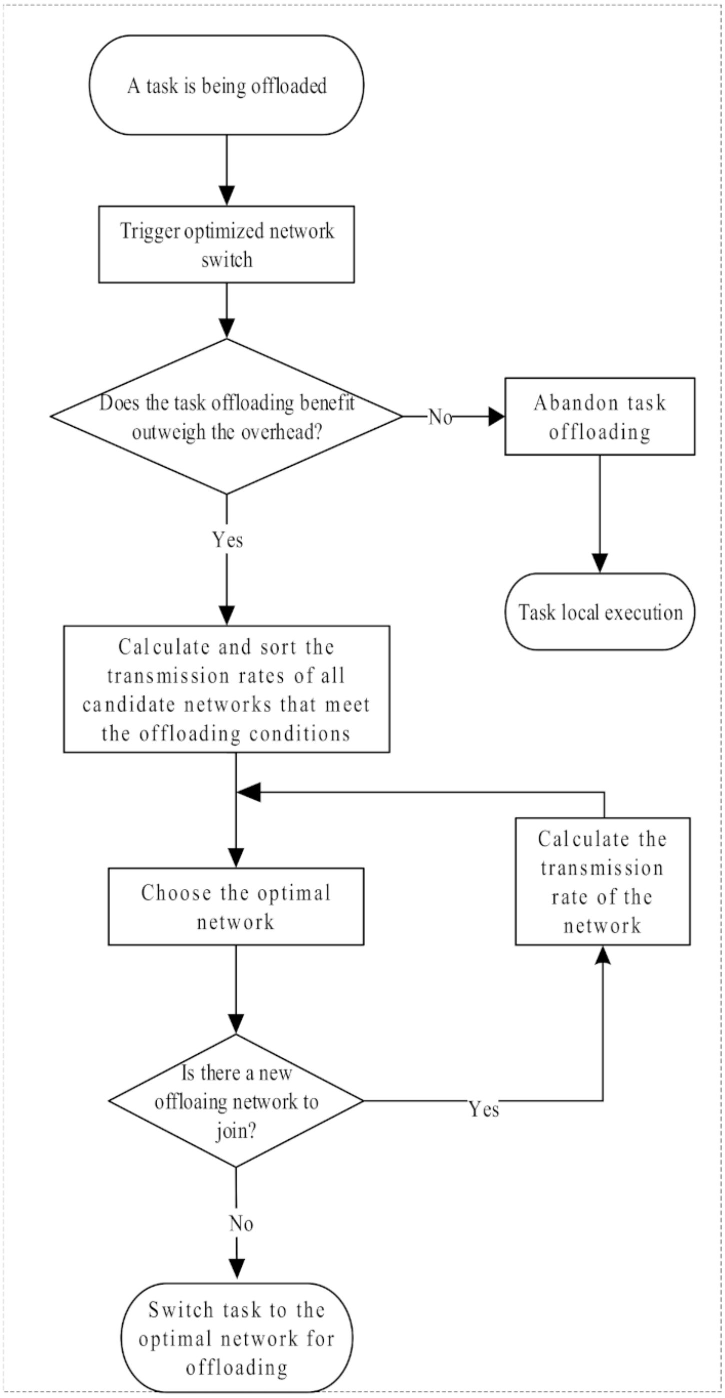
A task is being offloaded
Trigger optimized network switch
Does the task offloading benefit outweigh the overhead?
No
Abandon task offloading
Task local execution
Yes
Calculate and sort the transmission rates of all candidate networks that meet the offloading conditions
Calculate the transmission rate of the network
Choose the optimal network
Is there a new offloaing network to join?
Yes
No
Switch task to the optimal network for offloading

Figure 3. Network Switching Flowchart. This figure represents the network selection and switching process during task offloading. Network switching is always present during the whole task offloading process, always keeping the network where the current task is offloaded as the optimal network, so that the delay and energy cost of task offloading are minimized.

When the computational resources of the offloading sites do not meet the offloading conditions, it's essential to select new offloading sites or perform network switching or abandon the task computation offloading and change to execute at local terminal devices. However, due to the complex environment of the Space–Air–Ground Integrated Network, the offloading network is prone to network interruption during the switching process. To avoid network interruption caused by the change of offloading network, it is required to select the most optimal offloading sites before computation offloading process, construct the most optimal offloading network, and reduce the probability of network switching in the decision process. Figure 4 shows the mobile model of the task offloading process.

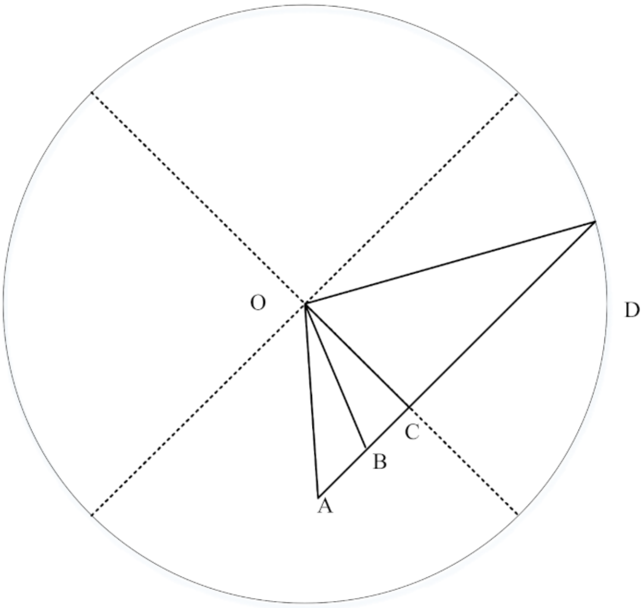

Figure 4. Task Offloading Process Movement Model. The connection boundary of the wireless network is a circular shape. Its center, O, is the transmitting and receiving antenna of the signal of the offload terminal device, which represents the terminal location of the current task offload of the network. The radius of the boundary circle (indicated by the dotted line) is the connection range of the infinite network, and the solid line indicates the moving path of the terminal equipment within the circle, where Point A represents the initial position of the offloading site when the task is unloaded, and Points B, C and D are the positions of the offloading site in the process of moving.

Among the model, Point O represents the terminal position where the current task of the network is offloaded, Point A represents the initial position of the offloading sites when task offloading, and Points B, C, and D are the positions during the movement of the offloading sites. When the offloading sites move to the boundary of D, but the task offloading calculation has not been completed, the task offloading may be failing. Next, we choose the offloading network by calculating the residence time of the offloading sites within the network. From the Friss equation for calculating the turbulence transmission loss in free space [29] and the wireless signal strength attenuation model based on cubic spline interpolation [30], the signal strength P_A of the offloading terminal at A point can be obtained, as shown in Equation (1):

$$P_A = P_\xi - 10\phi \times \log_{10}(d) \quad (1)$$

where P_ξ represents the signal strength reference value, and ϕ represents the wireless network signal attenuation factor.

The distance d_{OA} between the offloading terminal O and the initial position A of the offloading site is shown in Equation (2):

$$d_{OA} = \frac{P_{\xi} - P_A}{10\phi} \tag{2}$$

In the same way, the distance d_{OB} from Point O to Point B can be obtained. From the known moving speed v of the offloading terminal O and the time t_{AB} required for the offloading terminal to move in the AB segment, calculate the distance d_{AB} between AB, as shown in Equation (3):

$$d_{AB} = t_{AB} \times \nu \tag{3}$$

OC is the vertical line of the offloading terminal's moving path. According to the triangle side length equation, triangle similarity theorem, and the known triangle side length, the distance between AD can be calculated finally. The resident time $T_j^{resident}$ of the terminal device in the wireless network is as shown in Equation (4):

$$T_j^{resident} = \frac{d_{AD}}{\nu} \tag{4}$$

A task i of the offloading terminal is offloaded to the offloading site j for execution, and the maximum calculation amount $f_{maxgive}$ provided by the offloading site j to the terminal device in the network is calculated as Equation (5):

$$f_{maxgive} = (T_j^{resident} - T_i^s - T_i^r) \times f_j \tag{5}$$

where T_i^s represents the transmission time of task i data offloading to the specified offloading site, T_i^r represents the data return time after the task is calculated and processed at the offloading site, and f_j represents the computing capacity of the offloading node j.

In the task computation offloading decision process, the computing capacity of all offloading sites in the current network is judged according to $T_j^{resident}$ and $f_{maxgive}$, the $f_{maxgive}$ s are sorted according to the value, and then select the most optimal offloading network. Use control parameters to represent the system state of the Markov decision process model in the offloading network. The entire system state is composed of the Markov decision process set. A single Markov decision process includes the system state set, action set, state transition probability, cost function and strategy space. At each decision moment of task offloading, the control parameters of the system correspond to the actions in the Markov decision process. The state transition probability of the system controls the dynamic changes of the state and control parameters at each stage. Further, the evaluation metrics for the final task computation offloading are presented by the cost function.

In the three-layer network architecture of Space–Air–Ground Integrated Network, select the offloading sites that meets $f_{maxgive} \geq w_{in}$ to constitute the target state set $X_{target} = \{0, 1, 2, \cdots, n, n+1\}$ in the Markov decision process, and w_{in} represents the computing power required for the offloading task. Among them, the newly added state is the initial state $x_{0,0}$ of task computation offloading and the state $x_{n+1,0}$ of task computation offloading completed. The specific Markov decision process of the model in Figure 2 is $M = (S, A, P_{sa}, R)$. During the decision period t, the offloading terminal selects the action a ($a \subset A$) according to the system state information $x_{t,i} = (t, i, \Upsilon_t)$, and transitions from the current state s to the next state s'. P_{sa} represents the state transition probability and the reward function R represented as $r(s'|s, a)$, where s ($s \subset S$) represents the set of all states in a single Markov process. In the system status information, Υ_t represents the channel status between the current offloading terminal and the offloading site; $i \in [0, k+6+5]$ represents the set of task offloading locations, where k represents the optional offloading sites in the edge access network, and 6 represents the offloading sites of the space-based backbone nodes, 5 represents the offloading sites of the ground-based backbone nodes. The state s is determined by the channel state and the location of the task offloading site. Assuming that the initial channel state is good, the initial state of the system is expressed as $x_{0,k} = (0, k, g)$, and the final state is expressed as $x_{n+1,k} = (n+1, k, \Upsilon_{n+1})$. When the target state of the offloading task transitions to the state $x_{n+1,0} \in X_{target}$, the task computation offloading is completed.

2.3. DQN-Based Computation Offloading Decision Algorithm

In the Markov computation offloading decision process, the strategy space contains countless kinds of strategies, and it is necessary to find the best state-action path as the final optimal decision, and then the offloading environment performs state transition according to the actions of the terminal selecting a certain offloading site in the network and calculates the reward value through the reward function. If the reward value is positive, the terminal is less likely to select that offloading site thereafter; if the reward value is negative, the trend is reversed. In this process, a large number of state-action pairs are generated. In this section, the DQN algorithm is used to process these state-action pairs, and the convolutional neural network and Q-learning algorithm are fused to solve the task offloading decision process model to find the best action corresponding to each state and complete the optimal task computation offloading.

Using the Q-learning algorithm, the intelligent terminal transfers the state of the offloading tasks through the real-time action value and the cumulative reward value. During the task offloading process, the channel state in the network is estimated in advance and the computing capacity of the terminal is monitored. During the Q-learning process, the processing status of the task is represented as (O_i, w_{total}), where O_i represents the offloading site's number of the current processing task, and w_{total} represents the total amount of task data processed up to the current state. The next action of the task is represented as (O_i, w_i), where w_i represents the amount of data to be offloaded when the next action is performed. Suppose the initial reward function is 0, and the instant reward function of each step in the offloading task is calculated by the reciprocal of the linear expression of total delay and total energy consumption. The expressions of delay C_{T_i} and energy consumption C_{E_i} are as follows:

$$T_{t_i} = FT^{l}_{t_i} = T^{l}_{t_i} + RT^{l}_{t_i} \qquad (executed\ local) \tag{6}$$

$$T_{t_i} = FT^{ed}_{t_i} = T^{ed}_{t_i} + RT^{ed}_{t_i} \qquad (offloaded\ to\ edge\ compute\ node) \tag{7}$$

$$T_{t_i} = FT^{b}_{t_i} = T^{b}_{t_i} + RT^{b}_{t_i} \qquad (offloaded\ to\ bone\ cpmpute\ node) \tag{8}$$

$$E_{t_i} = E^{l}_{t_i} = \gamma_{in} \cdot w_{in} \qquad (executed\ local) \tag{9}$$

$$E_{t_i} = E^{ed}_{t_i} = T^{\mathrm{eds}}_{t_i} p_s + T^{ed}_{t_i} p_{\mathrm{c}} \qquad (offloaded\ to\ edge\ compute\ node) \tag{10}$$

$$E_{t_i} = E^{b}_{t_i} = T^{bs}_{t_i} p_s + T^{br}_{t_i} p_r \qquad (offloaded\ to\ bone\ cpmpute\ node) \tag{11}$$

Considering that in the Space–Air–Ground Integrated Network offloading environment, both the offloading terminals and the offloading sites have mobility. In Equations (4) and (5), the communication duration $T^{resident}_{j}$ of the offloading terminals and the offloading sites, and the maximum computing capacity $f_{maxgive}$ which the offloading sites can provide are essential constraint conditions. According to the conditions of $f_{maxgive} \geq w_{in}$ in the Markov computation offloading decision process, the offloading sites that can theoretically complete the calculation task are obtained. Among these offloading sites that can complete the calculation task, select the offloading site with the smallest sum of delay and energy consumption cost, and the computation offloading decision taken at this time is optimal. The traditional measure of the performance of the offloading decision uses delay or energy consumption as a single constraint and seeks another minimum value to select the optimal offloading decision. To be able to balance the cost of delay and energy consumption, while taking into account the characteristics of different offloading tasks, weight parameters [31] α and β that the maximum value is 1 are set for the cost of the task, respectively representing the weight of delay and energy consumption. Each task that executes offloading selects the size of the parameter value α and β according to its characteristic attributes. For example, for military emergency tasks, the parameter value α can be set bigger. The task computation offloading optimization goals are as follows:

$$minimize\left\{ Totalcost(C) = \alpha \sum_{v_i \in V} \sum_{i=1}^{m} C_{T_{in}} + \beta \sum_{v_i \in V} \sum_{i=1}^{m} C_{E_{in}} \right\} \tag{12}$$

$$st. \quad \forall : f_{maxgive} \geq w_{in} \tag{13}$$

$$\alpha + \beta = 1 \tag{14}$$

where m represents the number of subtasks of the offloading task v of the terminal device n, i represents the number of currently executed subtasks $0 \leq i \leq m$, and C represents the weighted total cost of delay and energy consumption. After obtaining the above-mentioned optimal strategy, it is also necessary to compare its total cost with the total cost of the local execution of the task. If the offloading cost is greater than the local execution cost, the task is executed locally, otherwise, the execution is offloaded.

For edge computing nodes, space-based backbone nodes, and ground-based backbone nodes in the Space–Air–Ground Integrated Network, there are differences in the computing storage performance of offloading sites. Therefore, the calculation equations for delay and energy consumption of different offloading sites are given, as shown in Equations (6) to (11), the reward function is shown in Equation (15).

$$r_t = 1/[\alpha T_{t_i} + \beta E_{t_i}] \tag{15}$$

Suppose the terminal takes action a_t at time t, the environment transitions from state s_t to s_{t+1}, and the winning case is $r_t = r(s_t, a_t, s_{t+1})$;$s_t$ and a_t determine the probability distribution of s_{t+1} and r_t, and Q_t is the action estimate of the initial state action pair at time t. The optimization process of action-value function in the Q-learning process is shown in Equations (16) and (17).

$$Q(s, a) = r + \gamma \max_{a \in A} Q(s', a') \tag{16}$$

$$Q(s, a) = (1 - \lambda_t)Q(s, a) + \lambda_t \left\{ r + \gamma \max_{a \in A} Q(s', a') \right\} \tag{17}$$

where λ_t represents the learning rate. The larger the value of λ_t is, the faster the convergence speed will be. However, the value should not be too large, otherwise, the convergence will be unstable and the optimal computation offloading decision will be affected.

Since Q-learning cannot store all possible states that in the complex Space–Air–Ground Integrated offloading network in a Q table, introducing the neural network to estimate the $Q(s, a)$ generated in Q-learning can reduce the state space in the process of computation offloading decision and accelerate the convergence rate of the offloading process. Equations (16) and (17) is used to calculate the partial action $Q(s, a)$ value of a single task during the task offloading decision period t, which is stored as the experience value in the memory bank Ω and used as the training data set of the convolutional neural network. Then the vector containing the value of each action $Q(s, a)$ is output through two fully connected convolutional network layers. The neural network is used to convert a single Q value into a Q network, and then the backpropagation and gradient descent methods are used to minimize the loss function, as shown in Equation (18):

$$L = (r + \gamma \underset{a'}{max}(s', a') - Q(s, a))^2 \tag{18}$$

Finally, Equations (8) and (11) choose the optimal offloading decision to execute the task.

The DQN computation offloading algorithm uses the deep convolutional neural network to nonlinearly approximate the action-value function in Q-learning during the computation offloading decision process, adopts the experience playback method to train the Q-learning process, and randomly extracts experience data from the state space to learn. This method breaks the correlation between past experiences so that it can quickly find the best task computation offloading decision path in the strategy space of the Space–Air–Ground Integrated Network. The specific computation offloading algorithm process is shown in Algorithm 1:

Algorithm 1 Computation Offloading Algorithm Based on DQN

```
Input:task amount; training cycle; the offloading terminal state-action value; location of the
    offloading terminals and the offloading sites; movement speed of the offloading terminals;
Output: total cost of task offloading C
initial the computation offloading service request of the offloading terminals; initial the
    offloading terminal state-action value and read the task amount;
calculate the resident time of the offloading terminal inside the network. $T_j^{resident} = \frac{d_{AD}}{v}$;
obtain information about offloading sites in the network;
calculate the maximum calculation amount that each offloading site can provide
    $f_{maxgive} = (T_j^{resident} - T_i^s - T_i^r) \times f_j$;
Sort $f_{maxgive}$ according to the size of the value and select the optimal offloading network;
repeat
  detect the performance and amount of offloading sites;
  perform data cutting based on the performance of each offloading site;
  repeat
    calculate the state of each state after selecting an action;
    select the corresponding action $a_t$ from the state $s_t$;
    execute action $a_t$ and return the excitation value $r_t$ and state $s_t$;
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t)]$
    for $t = 1; t < n + 1; t + +$ do
        store experience value $< s_t, a_t, r_t, s_{t+1} >$ in memory $\Omega$.
        train the Q network;
        calculate the minimum value of the loss function
$L = (r + \gamma \max_{a'} x(s', a') - Q(s, a))^2$
    end for
    choose the best offloading site, corresponding to state $s_t \leftarrow s_{t+1}$
  until end of each training cycle;
  if the task status in $(O_i, w_{total})$ is changed to $x_{n+1,0}$ then
        complete the task computation offloading;
        Send the data to the offloading terminal;
  end if
until Over task amount;
All task data processing completed.
```

3. Results

The Space–Air–Ground Integrated Network simulation scene is set as follows: According to the 6 + 5 + x structure of the Space–Air–Ground Integrated Network, two space-based backbone nodes, two ground-based backbone nodes, and six edge access nodes were selected for simulation. That is, there are one offloading terminal and ten computation offloading sites, among which four are used to simulate the computation offloading sites of space-based or ground-based backbone nodes (B_offloadee), and the rest are computation offloading sites of edge computing nodes (E_offloadee), which are randomly distributed in different latitudes, and all nodes are connected through the wireless network. The factors that affect the task computation offloading decision include the offloading terminal, offloading sites, and attributes of the task itself, as shown in Table 1. Considering the diversity of offloading sites of space-based backbone nodes, ground-based backbone nodes, and edge access nodes in the network, the performance of ten offloading sites are set, as shown in Table 2, where $B1 \cdots B4$ represents the performance of backbone nodes and $E5 \cdots E10$ represents the performance of six edge offloading sites.

Table 1. Factors Affecting Computation Task Offloading Decisions.

Type	Factors Affecting Decision
Terminal (Offloader)	Data privacy and location
Offloading sites (Offloadee)	CPU performance, energy, storage space, memory size
Task characteristics	Calculate the data size and execution delay of offloading tasks

Table 2. Performance of Each Offloadee.

Type	Computing Frequency
B1	3 GHz
B2	3.5 GHz
B3	4 GHz
B4	5 GHz
E1	2 MHz
E2	3 MHz
E3	5 MHz
E4	1 MHz
E5	2.5 MHz
E6	4MHz

The simulation in this paper utilized Python 3.7 for algorithm simulation and algorithm analysis, and the code run dependency libraries including Tensorflow, Keras, Numpy, Scipy, Matplotlib, CUDA, etc. We used Spyder which the integrated development environment (IDE) that comes with Anaconda, perform the simulation verification, and the specific results are as follows.

Firstly, taking the weighted total cost of delay and energy consumption as indicators, the optimization performance of the DQN computation offloading algorithm is compared with that of traditional computation offloading algorithm including Lyapunov-based methods and classical Q-learning-based methods. Figure 5 shows the comparison of the total cost for delay and energy consumption weight of 0.5 each, indicating the need to balance delay and energy consumption when offloading tasks. Figure 6 compares the total delay and energy cost of the three computational offloading methods for delay-sensitive offloading tasks. The curves obtained from the three computation offloading methods in Figures 5 and 6 show an increasing trend, indicating that the total delay and energy consumption cost for task offloading in the integrated network increases with the amount of tasks to be offloaded, while the offloading strategy based on the DQN computational offloading algorithm proposed in this paper can significantly reduce the delay and energy consumption cost.

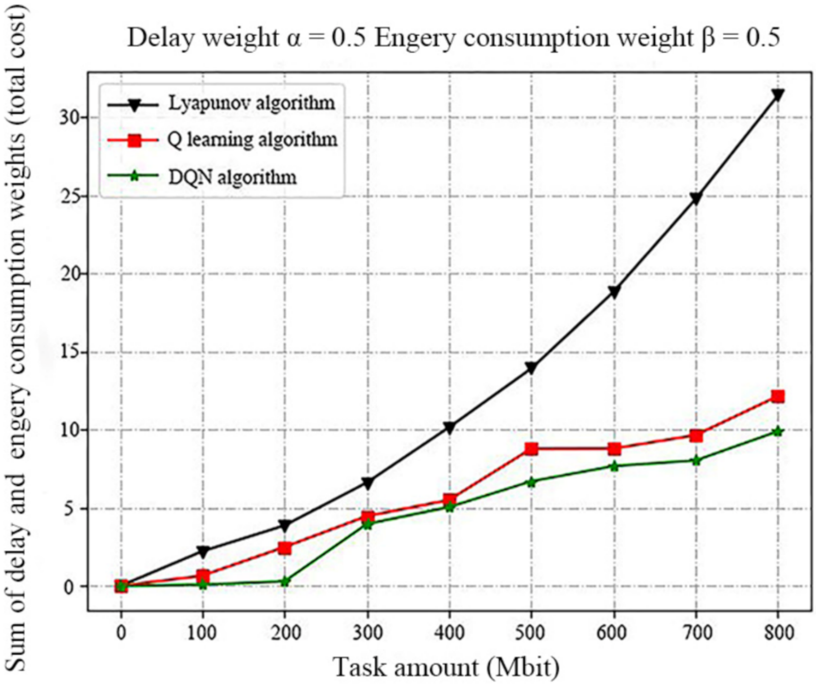

Figure 5. Comparison of total cost of delay and energy consumption weight at 0.5. Black indicates the computation offloading method based on Lyapunov algorithm, red indicates the computation offloading method based on classical Q-learning algorithm, and green indicates the computation offloading method of DQN algorithm proposed in this paper, comparing the cost of the three approaches when the delay and energy consumption are equally important.

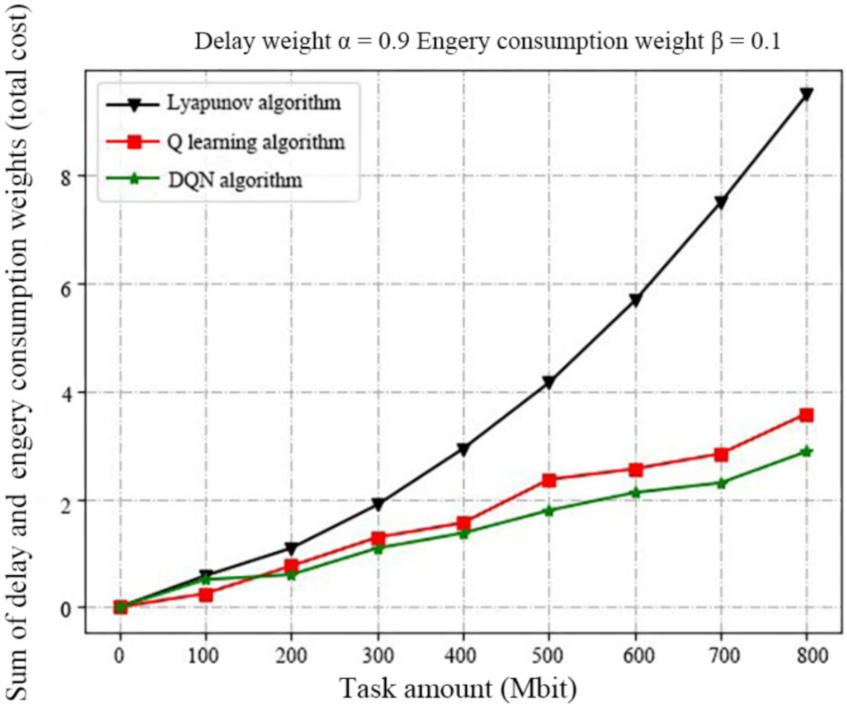

Figure 6. Comparison of total cost when the delay weight is 0.9. Black indicates the computation offloading method based on Lyapunov algorithm, red indicates the computation offloading method based on classical Q-learning algorithm, and green indicates the computation offloading method of DQN algorithm proposed in this paper, comparing the time delay and energy consumption cost of the three approaches under experimentally sensitive task requirements.

Secondly, using the number of times of network switching as an indicator, we compare and analyze the impact of the DQN computation offloading method, Lyapunov algorithm, and Q-learning algorithm on the task offloading decision process under the Space–Air–Ground Integrated Network environment. In the process of offloading, network switching will increase the cost of task execution. To reduce the cost of delay and energy consumption, the number of network switching needs to be offloaded as few as possible. The results of this comparison are shown in Figure 7. The graphs of all three strategies show an increasing trend, i.e., it shows that the number of network switches performed by all three strategies gradually increases as the number of tasks increases in the Space–Air–Ground Integrated Network. The Lyapunov algorithm-based computation offloading strategy always has the highest number of network switches, and the DQN algorithm based computational offloading strategy proposed in this paper has significant advantages when the number of offloading tasks is high, with the lowest number of network switches and thus the lowest additional cost during the computational offloading of tasks.

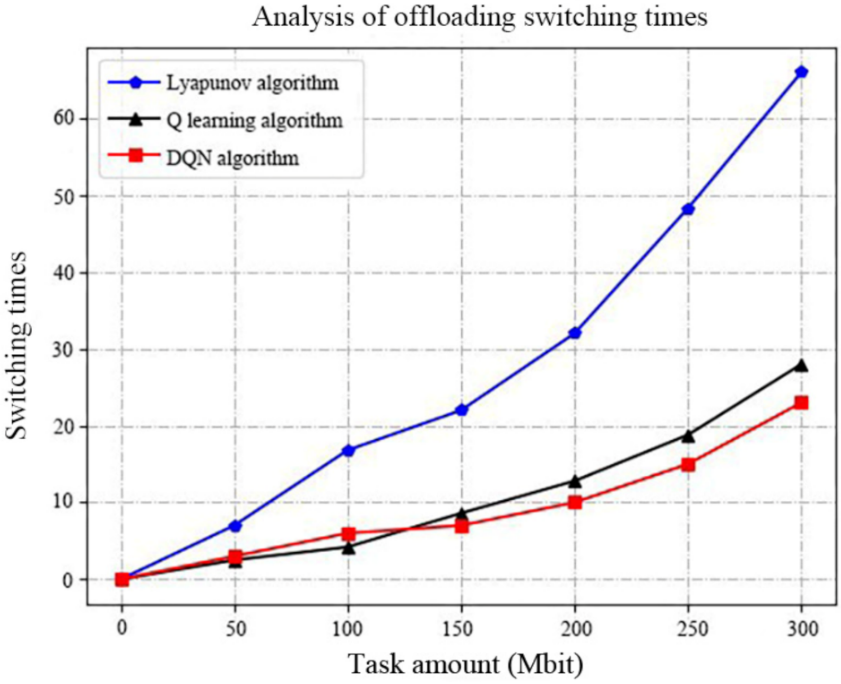

Figure 7. The change in the number of subtasks switching frequency changes in offloading. Blue indicates the computation offloading based on Lyapunov algorithm, black indicates the computation offloading based on classical Q-learning algorithm, and red indicates the computation offloading method of DQN algorithm proposed in this paper. Compare the number of network switching with task amount during task offloading of the three approaches.

Then, using the delay and energy consumption as indicators, according to the characteristics of the three-layer architecture of the Space–Air–Ground Integrated Network and the terminal's dynamic movement, the effectiveness of the Markov computation offloading decision process to maximize the use of the three types of computing nodes in the network is verified. Classify task offloading into all offloading to edge computing nodes, all offloading to backbone nodes, and mixed computation offloading at edge backbone nodes. Figure 8 compare the energy consumption cost of the mixed computation offloading under a certain work request volume with that of the mode in which the particular offloading sites are selected separately, while Figure 9 compares the delay cost. The graphs for all three task offloading modes show an upward trend as the workload (i.e., the number of task requests) increases. The rising trend of the mixed compute offloading mode proposed in this paper rises slowly compared to the computation offloading mode that offloads all to the backbone nodes, while the computation offloading mode that offloads all to the edge nodes tends to saturate when the workload is high. Therefore, the mixed compute offload mode for edge backbone nodes in this paper has lower energy and delay cost.

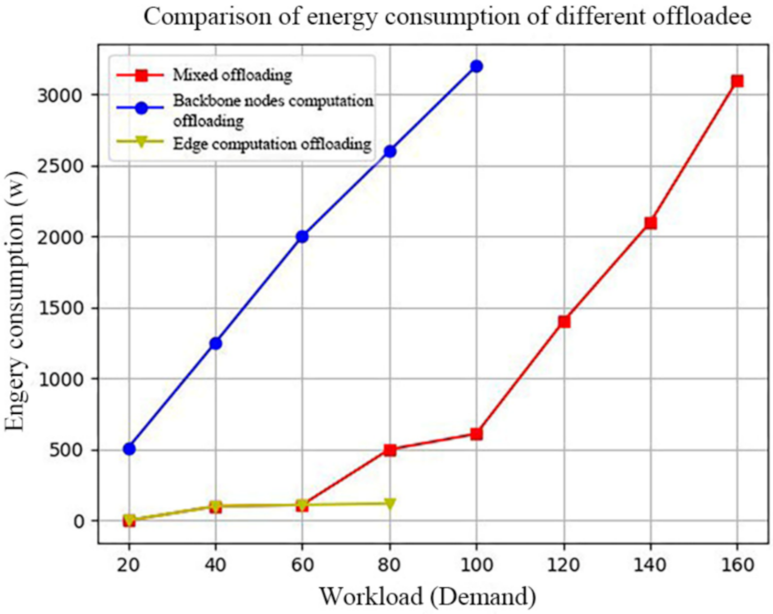

Figure 8. Comparison of energy consumption costs of different offloading sites. Blue indicates that all tasks are offloaded to the backbone nodes of the network, yellow indicates that all tasks are offloaded to the edge nodes of the network, and red indicates the mixed offloading mode proposed in this paper, where some tasks are offloaded to the backbone nodes and the others are offloaded to the edge nodes. Compare the energy consumption cost of the mixed computation offloading under a certain work request volume with that of the mode in which the particular offloading sites are selected separately.

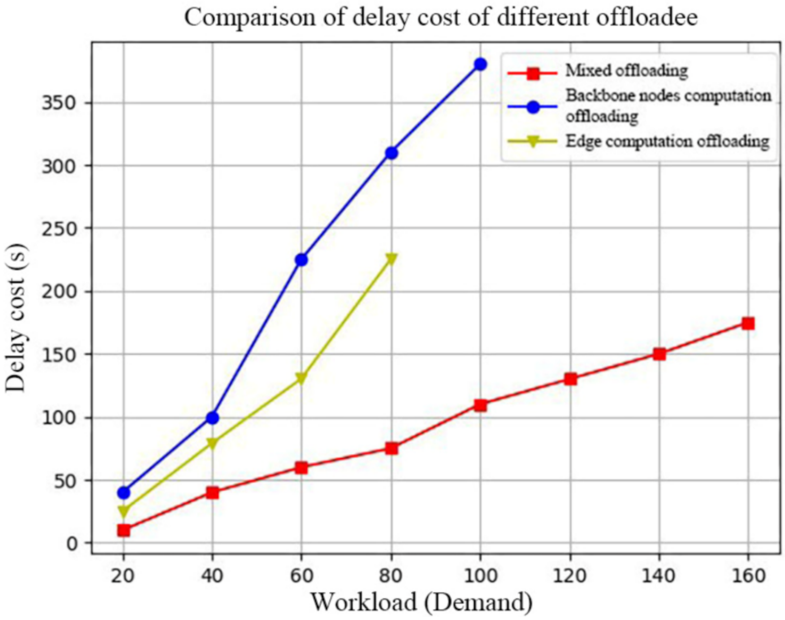

Figure 9. Comparison of delay costs of different offloading sites. Blue indicates that all tasks are offloaded to the backbone nodes of the network, yellow indicates that all tasks are offloaded to the edge nodes of the network, and red indicates the mixed offloading mode proposed in this paper, where some tasks are offloaded to the backbone nodes and the others are offloaded to the edge nodes. Compare the delay cost of the mixed computation offloading under a certain work request volume with that of the mode in which a particular offloading sites are selected separately.

Based on the simulation results of the algorithm, the number of network switches and the task offloading mode, the DQN-based offloading decision scheme can be applied to the Space–Air–Ground Integrated Network, which has large number of tasks to be offloaded, the complex offloading environment and the limited network resources, and it can effectively reduce the time delay and energy consumption during the task offloading process.

4. Discussion

The main objective of this paper is to solve the problems of low computational offload efficiency, high delay and energy consumption cost in Space–Air–Ground Integrated Network due to the complex and variable network offloading environment, the large amount of offloading tasks and the limited computational and storage resources of the network nodes, and to investigate a computation offloading solution that can adapt to these characteristics of the Space–Air–Ground Integrated Network. In this paper, we investigate a computation offloading solution that can accommodate these characteristics of the integrated network. In the following, we discuss and analyze the findings from the algorithms, network switching times and task offloading modes of the computation offloading application, and then explain and validate the effectiveness and applicability of our proposed DQN-based computation offloading decision scheme.

For the comparison of computational offloading algorithms, the Lyapunov algorithm is the most widely used in computation offloading technology, which can maintain the stability of the offloading system when the amount of tasks increases and prevent the system performance from deteriorating sharply. The classical Q-learning algorithm can solve the Markov decision process model through the maximum cumulative reward obtained by the state-action, but the algorithm has limited applicability and is inefficient in the large-scale state space. We have compared the system cost of these three kinds of computation offloading algorithms when dealing with different amounts of offloading tasks to verify the effectiveness of the proposed computational offloading algorithm in updating the computation offloading decision process in the Space–Air–Ground Integrated Network. We set different weights for the delay-sensitive tasks that need to be offloaded on the offloading terminal. As can be seen from Figures 5 and 6, for the task offloading requirements with equal weighting of delay and energy consumption, and for offloading task volumes above 500 Mbit, the proposed computation offloading strategy reduces on average 68.33% compared to the traditional Lyapunov-based computational offloading decision scheme, and 11.21% compared to the Markovian decision method based on the classical Q-learning algorithm. Further, for delay-sensitive offloading tasks, the three computation offloading algorithms can effectively reduce the total cost of the offloading strategy under the condition that the delay-sensitive offloading tasks have a greater weight than the weight of energy consumption. The total cost of the offloading strategy of the three computation offloading algorithms shows an upward trend with the increase in the number of offloading tasks. In contrast, the upward trend of the DQN computation offloading algorithm slows down significantly, which can effectively reduce the total delay-energy consumption cost of the task offloading strategy. In Figure 6, when the amount of offloading tasks is less than 160 Mbit, the total cost of offloading strategy under the DQN computation offloading algorithm is slightly higher than the Q-learning algorithm and close to Lyapunov algorithm. This is because the DQN computation offloading algorithm is more complex than the other two algorithms, but the number of tasks to be offloaded in the Space–Air–Ground Integrated Network is huge. Therefore, the offloading strategy based on the DQN computation offloading algorithm can effectively reduce delay and energy consumption.

For the comparison of the number of network switches during task offloading for different computation offloading methods, as shown in Figure 7, the Lyapunov-based computation offloading method has the highest number of switches and the steepest upward trend. When the number of offloading tasks is small, the number of network switching times based on the Q-learning computation offloading method is the least; when the number of offloading tasks exceeds 130 Mbit, the number of network switching times based on the DQN computation offloading method is the least. It can be seen that the Lyapunov algorithm does not have the learning characteristics of the Q-learning algorithm and the DQN algorithm, and is not able to monitor the offloading environment in real-time for policy learning and updating, resulting in the highest number of network switches during the task offloading. Therefore, compared to Q-learning and DQN algorithms, the Lyapunov algorithm is not suitable for application in the complex Space–Air–Ground Integrated Network. For the Q-learning algorithm, the learning environment of the offloading network is relatively simple with a lower algorithm complexity than the DQN algorithm when the number of offloading tasks is small. However, considering the complexity, dynamic characteristics, and the large number of tasks in the Space–Air–Ground Integrated Network, the DQN-based computation offloading method can select the optimal computation offloading decision and offloading network, ensuring fewer network switches and effectively reducing the additional cost in the computation offloading process.

For the three different mission offloading modes of the Space–Air–Ground Integrated Network, as shown in Figure 8, the task has the lowest energy consumption for edge-only computation offloading, with the workload between 20 and 80 requests, which is close to the energy consumption for edge backbone nodes mixed computation offloading but is saturated with edge computing workloads above 80 requests. Only the backbone nodes have the largest computation offloading energy consumption and the fastest increasing. In contrast, the offloading task using the mixed computation offloading mode of edge and backbone nodes has lower energy consumption and high workloads, which effectively meets the work demands of a large number of offloading tasks in the Space–Air–Ground Integrated Network. Figure 9 shows that the offloading task using a mixed-mode of edge and backbone nodes computing has the lowest delay cost, and the delay cost tends to increase more slowly than other offloading modes as the workloads increase. Specifically, when the number of the network work requests exceeds 100, the proposed mixed offloading approach is 61.54% lower in energy consumption and 71.01% lower in delay cost than the full offloading to the network backbone nodes approach, and is able to satisfy a large number of task offloading requests without saturation. Therefore, selecting the mix of edge and backbone nodes for computation offloading is more suitable for the task offloading environment of the Space–Air–Ground Integrated Network than selecting a pattern of offloading sites individually.

5. Conclusions

In the context of the Space–Air–Ground Integrated Network, due to the complex and variable offloading environment, the huge amount of offloading tasks, and the different storage resources and computing capacity of different orbital nodes, this paper theoretically analyzes the status of the offloading terminals and the offloading sites, and the selection of the offloading network and proposes a computation offloading decision scheme based on the Markov decision process and DQN algorithm. Specifically, segment the task data, select the optimal offloading network and the optimal offloading sites, transform the task processing process into the state transformation process of the Markov model, and use the DQN computation offloading algorithm to optimize the Markov decision process and seek the optimal offloading strategy. The experiment proved that the proposed computation offloading strategy reduces 68.33% on average compared to the traditional Lyapunov-based computational offloading decision scheme and 11.21% on average compared to the Markov decision method based on the classical Q-learning algorithm when the number of offloading tasks exceeds 500 Mbit and under the weighting requirements such as delay and energy consumption. When the number of offloading tasks exceeds 130 Mbit, the number of network switches required to select the optimal computation offloading decision and offload the network by the computation offloading scheme proposed in this paper is significantly lower than that of the Lyapunov-based and classical Q-learning algorithm-based computation offloading methods. Moreover, when the number of network work requests exceeds 100, the proposed mixed offloading approach can satisfy a large number of task offloading requests, and the energy cost and delay cost are 61.54% and 71.01% lower than those of the offloading to the network backbone nodes and the offloading to the edge nodes, respectively. Therefore, the proposed computational offloading scheme is suitable for Space–Air–Ground Integrated Network, reduces the number of switching of offloading networks, effectively reduces the delay and energy consumption cost of task computational offloading, and improves the network resource utilization.

The Space–Air–Ground Integrated Network is the trend of future networks as a huge system that integrates communication and aerospace telemetry to achieve full network coverage. In previous research, computation offloading techniques that enable efficient task processing have been applied to mobile phone terminals and vehicle terminals, but in the future, as various terminal devices continue to develop and advance, and human needs become more and more advanced, all terminal devices will be part of the Space–Air–Ground Integrated Network, and the rational application of computation offloading techniques to the Space–Air–Ground Integrated Network is of great theoretical importance. The application of computational offloading technology to Space–Air–Ground Integrated Network is of great theoretical significance and practical application. In this paper, the existing computation offloading techniques are improved and applied to the Space–Air–Ground Integrated Network, starting from the structural characteristics of the network. On the one hand, according to the delay and energy consumption requirements of the Space–Air–Ground Integrated Network's task offloading, the task can be matched with the offloading sites to effectively reduce delay and energy consumption while taking into account the requirements. On the other hand, the mixed consumption offloading model is able to cope with the huge volume of task processing requests in the future Inter-net of everything era with limited network resources. Theoretically, it can contribute significantly to the efficiency of task processing, thus addressing the impact and challenges of the inherent shortcomings of the Space–Air–Ground Integrated Network on the future development of the Internet.

Author Contributions: Conceptualization, J.L. and C.L.; methodology, X.L.; software, J.L.; validation, J.L., X.L. and C.L.; formal analysis, J.L.; investigation, X.L. and C.L.; resources, J.L.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L.; visualization, J.L., X.L. and C.L.; supervision, J.L.; project administration, J.L.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the National Natural Science Foundation of China under Grant NO. 61671141.

Data Availability Statement: Not applicable, the study does not report any data.

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

References

1. Liang, H.; Chen, F.C.; Ji, X.S.; Lu, P.; Gao, Y. Development status and applied research on mimic technologies for space-ground integration information network. *Sci. Sin. Inf.* **2019**, *49*, 799–818.
2. Grazia, C.A.; Klapez, M.; Patriciello, N.; Amditis, A.; Sdongos, E. Integration of satellite and LTE for disaster recovery. *IEEE Commun. Mag.* **2015**, *53*, 47–53.
3. Conti, M.; Giordano, S. Mobile ad hoc networking: Milestones, challenges, and new research directions. *IEEE Commun. Mag.* **2014**, *52*, 85–96. [CrossRef]
4. Liu, J.; Shi, Y.; Fadlullah, Z.M.; Kato, N. Space-air-ground integrated network: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2714–2741. [CrossRef]
5. Dong, S.Q.; Li, H.L.; Qu, Y.B.; Zhang, Z.; Hu, L. Survey of research on computation offloading strategy in mobile edge computing. *Comput. Sci.* **2019**, *46*, 32–40.
6. Xie, Y.D. Research on Computation Offloading Strategy in Mobile Device Cloud. Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2019.
7. Guo, S. Research on Dynamic Offloading Decision in Mobile Cloud Computing Networks. Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2019.

8. Liao, H.; Zhou, Z.; Zhao, X.; Wang, Y. Learning-based queue-aware task offloading and resource allocation for space–air–ground-integrate power IoT. *IEEE Internet Things J.* **2021**, *8*, 5250–5263. [CrossRef]
9. Xie, R.C.; Lian, X.F.; Jia, Q.M.; Huang, T.; Liu, Y. Survey on computation offloading in mobile edge computing. *J. Commun.* **2018**, *39*, 138–155.
10. Zhang, H.; Guo, F.; Ji, H.; Zhu, C. Combinational auction-based service provider selection in mobile edge computing networks. *IEEE Access* **2017**, *5*, 13455–13464. [CrossRef]
11. Kao, Y.H.; Krishnamachari, B.; Ra, M.R.; Bai, F. Hermes: Latency optimal task assignment for resource-constrained mobile computing. *IEEE Trans. Mob. Comput.* **2017**, *16*, 3056–3069. [CrossRef]
12. Kamoun, M.; Labidi, W.; Sarkiss, M. Joint resource allocation and offloading strategies in cloud enabled cellular networks. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015.
13. Zhao, P.; Tian, H.; Qin, C.; Nie, G. Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing. *IEEE Access* **2017**, *5*, 11255–11268. [CrossRef]
14. Wang, W.; Zhou, W. Computational offloading with delay and capacity constraints in mobile edge. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017.
15. Du, J.; Zhao, L.; Feng, J.; Chu, X. Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. *IEEE Trans. Commun.* **2018**, *66*, 1594–1608. [CrossRef]
16. Liu, Y.; Yu, F.R.; Li, X.; Ji, H.; Leung, V.C. Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access. *IEEE Trans. Veh. Technol.* **2018**, *67*, 12137–12151. [CrossRef]
17. Ren, J.; Yu, G.; Cai, Y.; He, Y. Latency optimization for resource allocation in mobile-edge computation offloading. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 5506–5519. [CrossRef]
18. Chen, M.; Hao, Y.X. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 587–597. [CrossRef]
19. Yu, R.Z.; Xue, G.L.; Zhang, X. Application provision in fog computing-enabled internet-of-things: A network perspective. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 15–19 April 2018.
20. Wei, F.; Chen, S.X.; Zou, W.X. A greedy algorithm for task offloading in mobile edge computing system. *China Commun.* **2018**, *15*, 149–157. [CrossRef]
21. Chen, W.W.; Wang, D.; Li, K.Q. Multi-user multi-task computation offloading in green mobile edge cloud computing. *IEEE Trans. Serv. Comput.* **2018**, *12*, 726–738. [CrossRef]
22. Pham, Q.V.; Leanh, T.; Tran, N.H.; Park, B.J.; Hong, C.S. Decentralized computation offloading and resource allocation for mobile-edge computing: A matching game approach. *IEEE Access* **2018**, *6*, 75868–75885. [CrossRef]
23. Xu, J.; Chen, L.X.; Zhou, P. Joint service catching and task offloading for mobile edge computing in dense network. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 15–19 April 2018.
24. Yu, H.Y.; Wang, Q.Y.; Guo, S.T. Energy-efficient task offloading and resource scheduling for mobile edge computing. In Proceedings of the 2018 IEEE International Conference on Networking, Architecture and Storage (NAS), Chongqing, China, 11–14 October 2018.
25. Abdenacer, N.; Wu, H.; Abdelkader, N.N.; Dhelim, S.; Ning, H. A novel framework for mobile edge computing by optimizing task offloading. *IEEE Internet Things J.* **2021**. [CrossRef]
26. Zhang, W.; Aung, N.; Dhelim, S.; Ai, Y. DIFTOS: A distributed infrastructure-free traffic optimization system based on vehicular ad hoc networks for urban environments. *Sensors* **2018**, *18*, 2567. [CrossRef] [PubMed]
27. Xu, H.J. Research on Optimization in NB-IoT Based on Fog Computing. Master's Thesis, Beijing Jiaotong University, Beijing, China, 2019.
28. Mohammad, M. AlDurgam. An integrated inventory and workforce planning markov decision process model with a variable production rate. *IFAC PapersOnLine* **2019**, *52*, 13.
29. Zhang, C.Q. Research on Path Loss of ETC System Based on Radio Frequency Identification Technology. Master's Thesis, Hefei University of Technology, Hefei, China, 2019.
30. Chu, X.J. Research on Indoor Location and Application Based on WIFI Signal Strength. Master's Thesis, Jiangsu University of Science and Technology, Zhenjiang, China, 2016.
31. Fedushko, S. Adequacy of Personal Medical Profiles Data in Medical Information Decision-Making Support System. In Proceedings of the International Conference on Rural and Elderly Health Informatics (IREHI 2018), Cotonou, Benin, 3–4 December 2018; CEUR Workshop Proceedings. 2020; Volume 2544. Available online: http://ceur-ws.org/Vol-2544/shortpaper4.pdf (accessed on 10 May 2021).