

Article

Not peer-reviewed version

Joint Latency-Oriented, Energy Consumption, and Carbon Emission for Space-Air-Ground Integrated Network with New Designed Power Technology

[Yonghao Wang](#), [Bo Li](#)^{*}, Jiahao He, Jiaxing Dai, Yuxin Yang, Yidong Liu

Posted Date: 4 July 2023

doi: 10.20944/preprints202307.0219.v1

Keywords: space-air-ground integrated network; renewable energy; twin delayed deep deterministic policy gradient; latency; energy consumption



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

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

Article

Joint Latency-Oriented, Energy Consumption, and Carbon Emission for Space-Air-Ground Integrated Network with New Designed Power Technology

Yonghao Wang¹, Bo Li^{1,*}, Jiahao He¹, Jiaxing Dai¹, Yidong Liu¹, and Yuxin Yang¹

¹ School of Electronic-Electrical Engineering, Ningxia University 750021, China; 2873495240@qq.com (Y.W.); hjh2000a@163.com (J.H.); 1534792366@qq.com (J.D.); 705572904@qq.com (Y.Y.); lyd85084313@163.com (Y.L.)

* Correspondence: libo@nxu.edu.cn

Abstract: The ubiquitous connectivity for the space-air-ground integrated network (SAGIN) of the beyond fifth generation of communication and sixth generation of communication (B5G/6G) is envisaged to meet the needs for the demanded quality of service (QoS), green communication, and "dual carbon" target. However, the offloading and computation of massive latency-sensitive tasks dramatically increases the energy consumption of the network. Furthermore, the traditional power supply technology of the network base stations (BSs) enhances the carbon emission. To address these issues, we first propose a SAGIN architecture with energy harvesting devices, where the BS is powered by both renewable energy (RE) and the conventional grid. The BS explores wireless power transfer (WPT) technology to power the unmanned aerial vehicle (UAV) for stable network operation. RE sharing between neighbouring BSs is designed to fully utilize RE for reduce carbon emission. Secondly, on the basis of task offloading decision, UAV trajectory, and RE sharing ratio, we construct cost functions with joint latency-oriented, energy consumption, and carbon emission. Then, we develop a twin delayed deep deterministic policy gradient (TD3PG) algorithm based on deep reinforcement learning to minimize the cost function. Finally, simulation results demonstrate that the proposed algorithm outperforms the benchmark algorithm in terms of reducing latency, energy saving, and lower carbon emission.

Keywords: space-air-ground integrated network; renewable energy; twin delayed deep deterministic policy gradient; latency; energy consumption

1. Introduction

To accommodate the diverse quality-of-service (QoS) requirements of a wide range of applications in various scenarios, future communication networks are envisioned to offer low latency and ubiquitous connectivity [1]. In particular, the space-air-ground integrated network (SAGIN) is considered a promising paradigm for future communication to provide large-scale coverage and network performance enhancement [2].

On the other hand, mobile edge computing (MEC) is considered one of the key technologies for future communication network service for latency-sensitive tasks [3,4]. Meanwhile, the execution of tasks would consume a considerable amount of energy. Green communication in 6G should be manifested in reducing the total energy consumption in order to achieve energy efficiency goal [5]. In addition, The dual carbon goal of future communication networks is to reduce carbon emissions by 50% [6]. Thereby, joint latency-oriented for QoS guarantee, energy consumption for green communication and carbon reduction for dual carbon goal in SAGIN has attracted wide attention from both academia and industry.

Several studies have been put into reducing network latency for QoS guarantee. Peng et al. [7] consider two typical MEC architectures in a vehicular network and formulate multi-dimensional resource optimization problems, exploiting reinforcement learning (RL) to obtain the appropriate resource allocation decisions to achieve high delay/QoS satisfaction ratios. Through jointly optimizing

computation offloading decision and computation resource allocation, Zhao et al. [8] effectively improve the system utility and computation time. Abderrahim et al. [9] propose an offloading scheme that improves network availability, while reducing the transmission delay of ultra-reliable low-latency communication packets in the terrestrial backhaul and satisfying different QoS requirements.

Furthermore, a few other studies have considered the issue of energy consumption in networks. Literature [10,11] consider a multi-user MEC system with the goal of minimizing the total system energy consumption over a finite time horizon. Chen et al. [12] investigate the task offloading problem in ultra-dense network and formulate it as a mixed integer nonlinear program which is NP-hard, which can reduce 20% of the task duration with 30% energy saving, compared with random and uniform task offloading schemes.

Some scholars have concentrated their attention on latency and energy consumption for the SAGIN network architecture. Guo et al. [13] investigate service coordination to guarantee the QoS in SAGIN service computing and propose an orchestration approach to achieve low cost reduction in service delay. To achieve a better latency/QoS for different 5G slices with heterogeneous requirements, Zhang et al. [14] offload smartly the traffic to the appropriate segment of SAGIN. In order to ensure QoS for airborne users as well as to reduce energy consumption, Chen et al. [15] design an energy-efficient data saving scheme with extensive simulations confirming the effectiveness of the proposed scheme in terms of both the energy consumption and the processing delay in SAGIN.

Although the aforementioned studies have done a considerable amount of exploration on the issues of latency reduction and energy consumption reduction, research in [16] highlights that the dense deployment of higher energy-demanding communication devices will keep pumping up the energy consumption of wireless access networks. Moreover, the future networks have to reduce carbon emission. Fortunately, energy harvesting technology converts ambient energy to electric energy, which can be used in cellular networks to help reduce the carbon footprint of wireless networks [17]. In particular, the approach to green cellular BS has been proposed, which involves the adoption of renewable energy (RE) resources [18].

Consequently, the subject of utilizing RE to power BS reduce the carbon footprint of the network attracted academic attention. Yuan et al. [19] propose an energy storage assisted RE supply solution to power the BS, in which a deep reinforcement learning (DRL) based regulating policy is utilized to flexibly regulate the battery's discharging/charging. On the basis of off-grid cellular BSs powered by integrated RE, JAHID et al. [20] formulate a hybrid energy cooperation framework, which optimally determines the quantities of RE exchanged among BSs to minimize both related costs and greenhouse gas emissions. Despite the more advantages, RE harvesting technologies are still variable and intermittent compared to traditional grid power. An aggregate technology, which combines RE with traditional grid power, is the most promising option for reliably powering cellular infrastructure [21].

However, existing research efforts are limited to one or two aspects of the network in terms of reducing latency, energy saving, or access to new power supply technology. Meanwhile, few attention has been paid to the entire SAGIN with integrated power energy access technology. Apart from considered above, there are still a number of challenges facing SAGIN. First of all, the long-term performance of the network needs to be considered as the arrival, transmission, and processing of tasks is a stochastic and dynamic process over a period of time. DRL in artificial intelligence techniques can address problems with stochasticity and complexity, as well as dynamic problems. Secondly, due to the fact that the future communication is probably a cellular network capable of providing energy [22], it is extremely critical to power the UAV in order to maintain the sustainability of the network when device is powered. Fortunately, radio-frequency signal based wireless power transfer (WPT) can provide battery-equipped equipment with sustainable energy supply. For instance, literature [23–26] consider a wireless-powered multi-access network, where a hybrid access point powers the wireless devices via WPT.

Motivated by the above limitations and challenges, we inquire into two issues in SAGIN. One is how to combine latency-oriented, energy consumption, and carbon emission to model the objective function in SAGIN with new power access technology. The other is how the relevant influence factors affect latency, energy consumption and carbon emissions.

In this article, we first develop a SAGIN with a new designed power supply technology. In order to maintain the sustainability of the network, the BS explores the WPT to power the UAV. The UAV is able to dynamically adjust its trajectory to cope with task processing and charging. Joint the latency, energy consumption, and carbon emission is formulated the cost function. Obviously, our research target is to minimize the cost function.

The main contributions of this article can be summarized as follows:

- We propose an architecture for SAGIN with a new designed power supply technology. This network offers ubiquitous connectivity while adapting to the requirements of high reliability and green communication.
- We develop a cost function with a joint latency-oriented, energy consumption, and carbon emission, which facilitates to decrease the latency of task processing, the energy consumption and carbon footprint of the network through optimization.
- We put forward a TD3PG method. This DRL-based algorithm is capable of sensing parameter changes in the network and dynamically updates the offloading decision.
- We conduct experimental evaluations and comparisons. The proposed algorithm is compared with the other three benchmark algorithms. The results show that the proposed algorithm is outstanding in terms of reducing task latency, network energy saving, and lower carbon emission.

The rest of this article is organized as follows. In Section 2, the system model with SAGIN is presented. Section 3 provides the problem formulation. In Section 4, a TD3PG algorithm is designed to obtain the optimal values. Section 5 demonstrates the simulation results and performance analysis. Finally, Section 6 devotes to the conclusions.

2. System Model

A SAGIN is considered, which consists of M ground user devices, N BSs, K UAVs, and a low earth orbit (LEO) satellite. As shown in Figure 1, each BS, each UAV, and the LEO satellite is separately equipped with a server for providing the devices with computation services. Devices are randomly distributed on the ground, which satisfy $m \in \mathcal{M} = \{1, 2, \dots, M\}$. BSs are deployed on buildings of height H_{BS} and satisfy $n \in \mathcal{N} = \{1, 2, \dots, N\}$. The flight altitude of the UAV k is fixed as H . The communication range covers the preset range around itself, which satisfies $k \in \mathcal{K} = \{1, 2, \dots, K\}$. It is assumed that a LEO satellite is available within the communication coverage at all times and the meteorological environment remains constant. The set of the time slot is indicated as $t \in \mathcal{T} = \{1, 2, \dots, T\}$ with the size of each time slot is τ .

Each BS is independently equipped with solar panels, wind turbine, backup battery and electricity converter to provide uninterrupted energy supply. Adjacent BSs are able to share energy by energy sharing line. In order to maintain the stable operation of the UAV, the BS is able to supply the UAV with energy via WPT.

Given that virtual reality, augmented reality, and mixed reality are the typical applications based on the 6G [27], it is assumed that each device has one total task which can be divided into a number of subtasks. The subtasks from device m may be respresented as $\langle s_m(t), e_m(t), t_m(t) \rangle$, where $s_m(t)$ and $e_m(t)$ denote the subtask's size (bit) and required CPU cycles per unit of data (cycles/bit) in time slot t , respectively. $t_m(t)$ indicates the latency constraint for executing the subtask, which contains the latency for the transmission, computation, and waiting of the subtask. Subtasks follow a Poisson distribution with arrival rate $\lambda_m(t)$.

The relevant system models in SAGIN will be enumerated in the following.

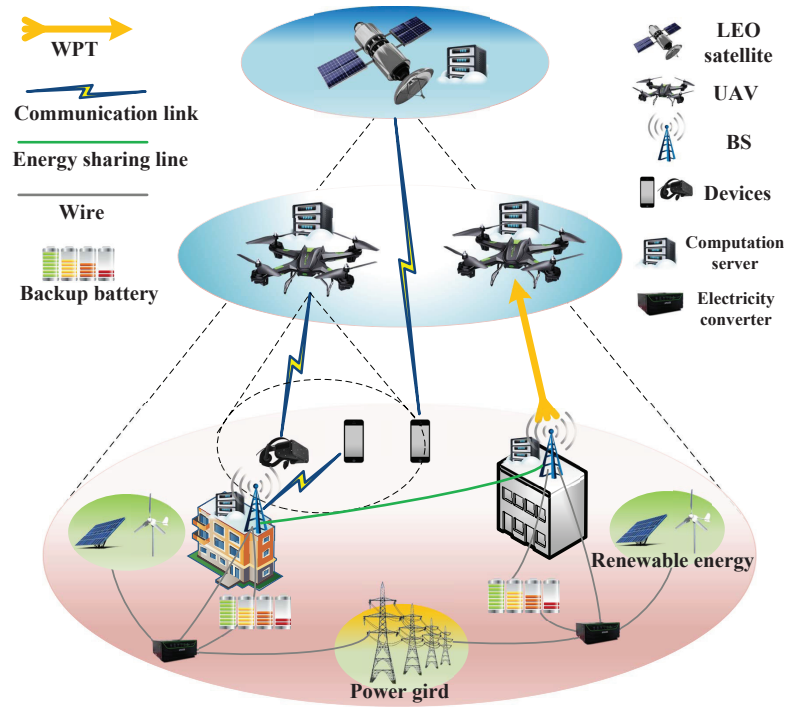


Figure 1. SAGIN scenario with new designed power supply.

2.1. Backup battery Model for BS

Denote $L_n(t)$ as the electrical load for BS n in time slot t , which is related to the communication traffic $T_n(t)$ [28]. Hence, $L_n(t)$ can be expressed as

$$L_n(t) = \delta_0 + \delta_1 T_n(t), \quad (1)$$

where δ_0 and δ_1 are coefficients.

The BS backup battery is required to reserve capacity $b_n^{res}(t)$ to meet the energy consumption for BS n electrical load during the fixed standby time $D_n^{res}(t)$ in time slot t . $b_n^{res}(t)$ can be represented as

$$b_n^{res}(t) = \int_0^{D_n^{res}(t)} L_n(\tau) d\tau. \quad (2)$$

During the time slot t , the harvested energy from RE by N BS backup batteries is $\mathbf{H}(t) = [h_1(t), h_2(t), \dots, h_N(t)]$. The energy sharing to the adjacent BS is denoted as $\mathbf{E}^{share}(t) = [e_1^{share}(t), e_2^{share}(t), \dots, e_N^{share}(t)]$. Let $\mathbf{B}^{res}(t) = [b_1^{res}(t), b_2^{res}(t), \dots, b_N^{res}(t)]$ denote the energy reserved by the backup battery to meet a fixed standby time at the current communication traffic of the BS. $\mathbf{E}^{bs}(t) = [e_1^{bs}(t), e_2^{bs}(t), \dots, e_N^{bs}(t)]$ is the energy consumption of each BS. Suppose the energy remaining in the backup battery is $\mathbf{B}^{bs}(t) = [b_1^{bs}(t), b_2^{bs}(t), \dots, b_N^{bs}(t)]$. Then backup battery during the next time slot can be derived as

$$\mathbf{B}^{bs}(t+1) = \min\{\mathbf{E}^{gird}(t) + \mathbf{H}(t) + \mathbf{B}^{bs}(t) - \mathbf{E}^{share}(t) - \mathbf{E}^{bs}(t), B^{battery}\}, \quad (3)$$

where $B^{battery}$ is the backup battery capacity. $\mathbf{E}^{gird}(t)$ is the energy supplied by power grid when RE harvesting is unable to maintain the BS energy consumption, which can be constructed as

$$\mathbf{E}^{gird}(t) = [\mathbf{B}^{res}(t) + \mathbf{E}^{bs}(t) - \mathbf{H}(t) + \mathbf{E}^{share}(t) - \mathbf{B}^{bs}(t)]^+, \quad (4)$$

where $[x]^+ = \max\{x, 0\}$. Considering the cost and size of BS energy harvesting equipment, the optimal unit size is finitely large. On the other hand, battery lifetime B^{life} is another factor to be considered,

which is related to the number of charge/discharge cycles. In this case, the following constraints are required to be satisfied as

$$B^{battery} \geq \int_0^{D_n^{res}(t)} L^{max} d\tau, \quad (5a)$$

$$CPR = \frac{B^{life} B^{battery}}{C^{battery}}, \quad (5b)$$

$$C^{battery} \leq C^{expect}, \quad (5c)$$

where L^{max} is the maximum electrical load for the BS. Define CPR as the cost-performance ratio of the backup battery, which is as large as possible. $C^{battery}$ and C^{expect} denote the actual and the maximum expectation cost of the backup battery, respectively.

Furthermore, adjacent BSs are able to share energy by physical resistive power line connections to maximize RE utilization. This approach may be reasonable for energy sharing among BSs within small zones [20]. The remaining energy of the RE may be shared with other BSs. Then the sharing energy $E^{share}(t)$ can be expressed as

$$E^{share}(t) = v_n \cdot [H(t) + B^{bs}(t) - B^{res}(t) - E^{bs}(t)]^+, \quad (6)$$

where v_n is the RE transfer ratio with $v_n \in [0, 1]$. When the backup batteries are fully charged, the excess RE $E^{exc}(t) = [e_1^{exc}(t), e_2^{exc}(t), \dots, e_N^{exc}(t)]$ will be discarded. Supposing that the loss factor for energy transfer between BSs is η , which satisfies $\eta \in [0, 1]$. Thus, the energy loss $E^{loss}(t) = [e_1^{loss}(t), e_2^{loss}(t), \dots, e_N^{loss}(t)]$ can be expressed as

$$E^{loss}(t) = \eta E^{share}(t) + E^{exc}(t). \quad (7)$$

It is crucial to reduce energy loss $E^{loss}(t)$ in evaluating the reduction of carbon emission.

2.2. Communication Model

Each subtask can be offloaded to BS, UAV, or LEO satellites for execution in one of these ways in time slot t . Thereafter, the task communication model is described as follows.

2.2.1. Offloading to the BSs

Let $R_{m,n}(t)$ denotes the wireless transmission rate between device m and BS n in time slot t , which can be expressed as

$$R_{m,n}(t) = B_{m,n} \log_2 \left[1 + \frac{p_m G_{m,n}(t) g_{m,n}(t)}{\sigma^2 + \zeta_{m',n'}(t)} \right], \quad (8)$$

where $B_{m,n}$ represents the channel bandwidth between device m and BS n . p_m is the transmission power of device m . $G_{m,n}(t)$ is the path loss that is related to the distance $d_{m,n}(t)$ between device m and BS n in time slot t . $g_{m,n}(t)$ is the small-scale channel fading subject to Rayleigh fading. σ^2 denotes the noise power. $\zeta_{m',n'}(t)$ is the interference from the other BSs.

2.2.2. Offloading to the UAVs

If device m is in the communication coverage of UAV k in time slot t , $a_{m,k}(t) = 1$ is assumed. Otherwise, $a_{m,k}(t) = 0$. As the altitude of the UAV is much higher than that of the devices, the line-of-sight channels of the UAV communication links are much more predominant [29]. The wireless channel of the UAV is considered as a free-space path loss model. Thereby the channel gain $g_{m,k}(t)$ is $\beta_0(d_{m,k}(t))^{-\alpha_0}$, where β_0 denotes the channel gain at unit distance. α_0 represents the path loss index.

$d_{m,k}(t)$ is the Euclidean distance between the m -th device and the k -th UAV. Hence, the transmission rate $R_{m,k}(t)$ can be formulated as

$$R_{m,k}(t) = B_{m,k} \log_2 \left[1 + \frac{a_{m,k}(t) p_m g_{m,k}(t)}{\sigma^2 + \zeta_{m',k'}(t)} \right], \quad (9)$$

where $B_{m,k}$ is the channel bandwidth. The $\zeta_{m',k'}(t)$ represents the co-channel interference from all the other UAVs.

2.2.3. Offloading to the LEO satellite

The channel conditions between the LEO satellite and ground devices are mainly affected by the communication distance and rainfall attenuation. Since the distance between LEO satellite and device remains almost unchanged, a fixed transmission rate $R_{m,sat}(t)$ between devices and the LEO satellite is adopted. It is usually lower than the transmission rate between devices and UAV communication [30,31].

2.3. Computation Model

The task can be executed in four ways, executed locally, offloaded to BSs, offloaded to UAVs, or offloaded to the LEO satellite. Due to the small data size of the computation result, the delay and energy consumption for transmitting the computation result can be omitted [32].

2.3.1. Local Computation

The device is capable of varying CPU cycle frequency $f_m(t)$ dynamically to obtain better system performance. Consequently, the latency of local task execution $t_m^l(t)$ can be obtained as

$$t_m^l(t) = \frac{s_m(t) e_m(t)}{f_m(t)} + t_m^{l,wait}(t), \quad (10)$$

where $t_m^{l,wait}(t)$ indicates the queuing delay of the task for the device. Assuming the energy consumption of per CPU cycle as $\kappa f_m^2(t)$, where κ is the CPU chip related switching capacitor [33], the energy consumption $E_m^l(t)$ of the task execution at the device can be expressed as

$$E_m^l(t) = \kappa f_m^2(t) s_m(t) e_m(t). \quad (11)$$

2.3.2. Offloading to the BSs

The BS has a relatively stable energy supply and a powerful CPU. If tasks are offloaded to the BS, they will be executed instantly without queueing. Denote that $f_{m,n}(t)$ is the allocated computation resource for the n -th BS. The task execution latency $t_n^{bs}(t)$ for offloading to the BS can be given as

$$t_n^{bs}(t) = \frac{s_m(t)}{R_{m,n}(t)} + \frac{s_m(t) e_m(t)}{f_{m,n}(t)}. \quad (12)$$

The energy consumption $E_n^{bs}(t)$ of the task execution at the BS can be obtained as

$$E_n^{bs}(t) = \delta_0 + \delta_1 s_m(t). \quad (13)$$

2.3.3. Offloading to the UAVs

The task execution latency $t_k^{uav}(t)$ for offloading to the UAV can be expressed as

$$t_k^{uav}(t) = a_{m,k}(t) \left[\frac{s_m(t)}{R_{m,k}(t)} + \frac{s_m(t) e_m(t)}{f_{m,k}(t)} + t_k^{u,wait}(t) \right], \quad (14)$$

where $f_{m,k}(t)$ is the allocated computation resource for the k -th UAV.

Define e_{uav} as the energy consumption per unit of computation resources of the UAV. The energy consumption $E_k^{uav}(t)$ of the task execution at the UAV can be given as

$$E_k^{uav}(t) = a_{m,k}(t) \left[\frac{p_m s_m(t)}{R_{m,k}(t)} + e_{uav} s_m(t) e_m(t) \right]. \quad (15)$$

2.3.4. Offloading to the LEO satellite

Denote that $f_{m,sat}(t)$ is the allocated computation resource for the LEO satellite. The task execution latency $t_m^{sat}(t)$ for offloading to the LEO satellite can be expressed as

$$t_m^{sat}(t) = \frac{s_m(t)}{R_{m,sat}(t)} + \frac{s_m(t) e_m(t)}{f_{m,sat}(t)} + t^{s,wait}(t). \quad (16)$$

The energy consumption $E_m^{sat}(t)$ of the task execution at the LEO satellite can be obtained as

$$E_m^{sat}(t) = \frac{p_m s_m(t)}{R_{m,sat}(t)} + e_{sat} s_m(t) e_m(t), \quad (17)$$

where e_{sat} is the energy consumption per unit of computation resources of the LEO satellite.

2.4. WPT Model

In the case that UAV k is within the BS n energy transmission coverage in time slot t , $o_{k,n}(t) = 1$. Otherwise, $o_{k,n}(t) = 0$. In order to maintain the sustainability of the network, the BS explores the WPT to power the UAV. Linear energy harvesting model is adopted. Then the harvested energy of the UAV k can be expressed as

$$B_k^H(t) = \sum_{k=1}^K o_{k,n}(t) \chi P_n G_n^t G_k^r h_{n,k}(t) \cdot \tau, \quad (18)$$

where χ denotes the energy conservation efficiency, which satisfies $\chi \in (0, 1]$. P_n is the BS transmitting power. G_n^t and G_k^r are the antenna power gains of the BS transmitter and the UAV receiver, respectively. $h_{n,k}(t)$ is the channel gain between the n -th BS and the k -th UAV in time slot t .

2.5. Flight Model

Continuous variables such as the average velocity $v_k(t)$ and the directional angle $\theta_k(t)$ are considered to adjust the trajectory of the UAV k . Assuming that the horizontal position coordinates and the moving distance of UAV k in current time slot are $\Psi_k(t) = [x_k(t), y_k(t)]^T$ and $d_k(t)$, respectively. The position coordinates of UAV k in the next time slot can be expressed as

$$\begin{cases} x_k(t+1) = x_k(t) + d_k(t) \cos \theta_k(t), \\ y_k(t+1) = y_k(t) + d_k(t) \sin \theta_k(t). \end{cases} \quad (19)$$

Referring to [34], the propulsion power consumption of UAV can be modeled as

$$P[v(t)] = P_0 \left(1 + 3v^2(t) / U_{tip}^2 \right) + \frac{1}{2} d_0 \rho s A v^3(t) + P_1 \left[\sqrt{1 + v^4(t) / 4v_0^4} - v^2(t) / 2v_0^2 \right]^{\frac{1}{2}}, \quad (20)$$

where P_0 and P_1 are constants, representing the blade profile power and induced power in hovering status, respectively. U_{tip} denotes the tip speed of the rotor blade. v_0 is known as the mean rotor induced velocity in hover. d_0 and s are the fuselage drag ratio and rotor solidity, respectively. ρ and A

denote the air density and rotor disc area, respectively. The flight energy consumption of UAV k in time slot t can be given as

$$E_k^F(t) = \int_t^{t+\tau} P[v_k(t)]dt. \quad (21)$$

The system model in SAGIN are enumerated in this section. In the following section, the cost function will be constructed.

3. Problem Formulation

How to integrate latency-oriented, energy consumption, and carbon emission to model the objective function in SAGIN is the key to solve the first issue. According to the system model mentioned above, the cost function is first constructed on the basis of system model and then the formulation of the problem is demonstrated in this section.

The network cost function $U(t)$ is constructed by joint the total latency $T(t)$, energy consumption $E^{ec}(t)$, and carbon emission $C^{RE}(t)$, which can be represented as

$$U(t) = \vartheta_0 T(t) + \vartheta_1 E^{ec}(t) + \vartheta_2 C^{RE}(t), \quad (22)$$

where ϑ_0 , ϑ_1 , and ϑ_2 represent the preference on latency, energy consumption, and carbon emission, respectively. And satisfy $\vartheta_0 + \vartheta_1 + \vartheta_2 = 1$. $T(t)$, $E^{ec}(t)$, and $C^{RE}(t)$ are expressed as

$$T(t) = \sum_{\substack{m \in \mathcal{M}, \\ n \in \mathcal{N}, k \in \mathcal{K}}} [t_m^l(t) + t_n^{bs}(t) + t_k^{uav}(t) + t_m^{sat}(t)], \quad (23)$$

$$E^{ec}(t) = \sum_{\substack{m \in \mathcal{M}, \\ n \in \mathcal{N}, k \in \mathcal{K}}} [E_m^l(t) + E_n^{bs}(t) + E_k^{uav}(t) + E_m^{sat}(t) + E_k^F(t)], \quad (24)$$

$$C^{RE}(t) = \sum_{n \in \mathcal{N}} \xi [h_n(t) - e_n^{loss}(t)], \quad (25)$$

where ξ is the conversion factor of carbon emission per unit of energy and is related to the type of energy source. $t_m^l(t)$, $t_n^{bs}(t)$, $t_k^{uav}(t)$, and $t_m^{sat}(t)$ represent the time when the tasks are executed locally or offloaded to the computation server in time slot t , respectively. In particular, each subtask can only be executed in one of these ways. $E^{ec}(t)$ includes energy consumption for tasks execution and UAV flight in time slot t . $C^{RE}(t)$ is the reduction in carbon emissions through reducing traditional power grid energy consumption.

The network cost function is formulated into a minimization problem as the following.

Constraint (26a) and (26b) imply that the task is indivisible and can be executed locally or remotely at the computation server. (26c) constrains the latency limit of the task. Constraint (26d) defines the range of angles and velocities that control the UAV's flight trajectory. (26e) ensures that the safety distance among different UAVs, where $\|\cdot\|$ is the Euclidean norm. Constraint (26f) indicates that the minimum and maximum thresholds for the UAV and BS backup batteries, respectively. (26g) guarantees that the computation resources allocated to the task by the device and other computation servers do not exceed the maximum computation capacity. (26h) specifies the range of variables.

It is a challenge to solve the above problem, since the dynamic and complex characteristics of the network environment. It is crucial to design a satisfactory solution that is real-time and capable of handling high-dimensional dynamic parameters. Therefore, in the following section, DRL is used to learn network and to deal with optimization problem with various constraints.

$$\min \sum_{t \in \mathcal{T}} U(t) \quad (26)$$

$$\text{s.t. } o_m^L(t), o_{m,n}^N(t), o_{m,k}^K(t), o_m^{\text{sat}}(t) \in \{0, 1\}, \quad (26a)$$

$$o_m^L(t) + \sum_{n=1}^N o_{m,n}^N(t) + \sum_{k=1}^K o_{m,k}^K(t) + o_m^{\text{sat}}(t) = 1, \quad (26b)$$

$$t_m^l(t), t_n^{bs}(t), t_k^{uav}(t), t_m^{\text{sat}}(t) \leq t_m(t), \quad (26c)$$

$$0 \leq \theta_k(t) \leq 2\pi, 0 \leq v_k(t) \leq v_{\max}, \quad (26d)$$

$$\|\Psi_{k1}(t) - \Psi_{k2}(t)\| \geq D_{\min}, k1 \neq k2, \quad (26e)$$

$$B_k^{\text{bot}} \leq B_k^K(t) \leq B^{\text{capacity}}, \quad (26f)$$

$$B^{\text{res}} \leq B_n^N(t) \leq B^{\text{battery}}, \quad (26f)$$

$$f_m(t) \leq f_m^{\max}, \sum_{m=1}^M f_{m,k}(t) \leq f_k^{\max}, \quad (26g)$$

$$\sum_{m=1}^M f_{m,n}(t) \leq f_n^{\max}, \sum_{m=1}^M f_{m,\text{sat}}(t) \leq f_{\text{sat}}^{\max}, \quad (26g)$$

$$\forall m \in \mathcal{M}, n \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}. \quad (26h)$$

4. Markov Chain Model based TD3PG Algorithm in SAGIN

DRL aims to learn award maximizing strategy by the interaction of intelligent agents with the environment. In order to solve the second issue with DRL, the cost function is reformulated as a Markov Decision Process (MDP). TD3PG, a DRL-based strategy, is proposed to optimize the objective problem by training the nearoptimal model.

4.1. MDP Formulation

A typical MDP with 4 elements tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{M}, \mathbf{R} \rangle$, where \mathbf{S} is the state space. \mathbf{A} is the action space. \mathbf{M} denotes the state transition function, which indicates the probability of the transition to next state s_{t+1} after performing (s_t, a_t) . \mathbf{R} is the reward function. The detail descriptions of the elements in the tuple are as follows.

- *State space*: State space is designed to store the DRL agent's observations in the environment and to guide the generation of actions. The state space includes the remaining energy of the UAV $B_k^K(t)$, the BS $B_n^{bs}(t)$, the reserved energy for BS backup battery $B_n^{\text{res}}(t)$, the current horizontal location of the UAV $\Psi_k(t)$, and the remaining amount of the current task $s_m(t)$. Then the system state space can be defined as

$$s_t = \{B_k^K(t), B_n^{bs}(t), B_n^{\text{res}}(t), \Psi_k(t), s_m(t)\}. \quad (27)$$

- *Action space*: Action space is designed to store the actions performed by the DRL agent in the environment in order to obtain feedback from the environment and the next state. The action space contains the offloading decisions $o_m^L(t), o_{m,n}^N(t), o_{m,k}^K(t), o_m^{\text{sat}}(t)$ (i.e., offloading task to local device, BS, UAV, or the LEO satellite). The navigation speed $v_k(t)$ and the rotation angle $\theta_k(t)$ for controlling UAV trajectory are continuous value in the interval $[0, v_{\max}]$ and $[0, 2\pi]$, respectively. $f_m(t) \in [f_m^{\max}/2, f_m^{\max}]$ denotes the dynamic computation resources of the device. v_n is the proportion of shared RE harvested by BS. Therefore, the system action space can be expressed as

$$a_t = \{o_m^L(t), o_{m,n}^N(t), o_{m,k}^K(t), o_m^{\text{sat}}(t), v_k(t), \theta_k(t), f_m(t), v_n(t)\} \quad (28)$$

- *Reward*: Reward is the feedback from the environment to the DRL agent after performing the current action. The research goal is to minimize the cost function. The algorithm depends on the reward to evaluate its decisions by learning the SAGIN environment. Due to the fact that the goal of RL is a cumulative overall reward over time to maximize reward, the reward function is

reformulated to be the negative value of the optimization cost function. Then the reward can be expressed as

$$R_t = -U(t), \quad (29)$$

where the coefficients ϑ_0, ϑ_1 , and ϑ_2 in U are required to satisfy $\vartheta_0, \vartheta_1 > 0$ and $\vartheta_2 < 0$, respectively.

4.2. TD3PG Algorithm

Owing to the fact that the action space contains continuous variables, TD3PG is adopted, which is an off-policy algorithm based on the actor-critic structure for dealing with the optimization problems with the high-dimension action space [35].

Compared to the conventional actor-critic based deep deterministic policy gradient (DDPG), which would result in cumulative errors. TD3PG solves the former overestimation problem. It has two pairs of critic networks in which the smaller Q value of the two Q-functions is chosen to form the loss function in the Bellman equation. Hence, there is less overestimation in learning process. Moreover, the application of delayed policy technique enables TD3PG to less frequently update the actor network and the target network than the critic networks. In other words, the model does not update policy unless the model's value function is updated sufficiently. The less frequent policy updates will bring a value evaluation with lower variance and produce a better policy [36].

Apart from the above, TD3PG is also able to smooth the value estimate by applying a smoothing technique for target policy that adds noise (clipped random noise) to the target actions and average over mini-batches, as shown in

$$\epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\omega}), -c, c). \quad (30)$$

According to the structure of the TD3PG algorithm depicted in Figure 2, the TD3PG architecture contains one actor network and two critic networks. The actor network is parameterized by θ_π , and two critic networks by θ_{Q1} and θ_{Q2} , respectively.

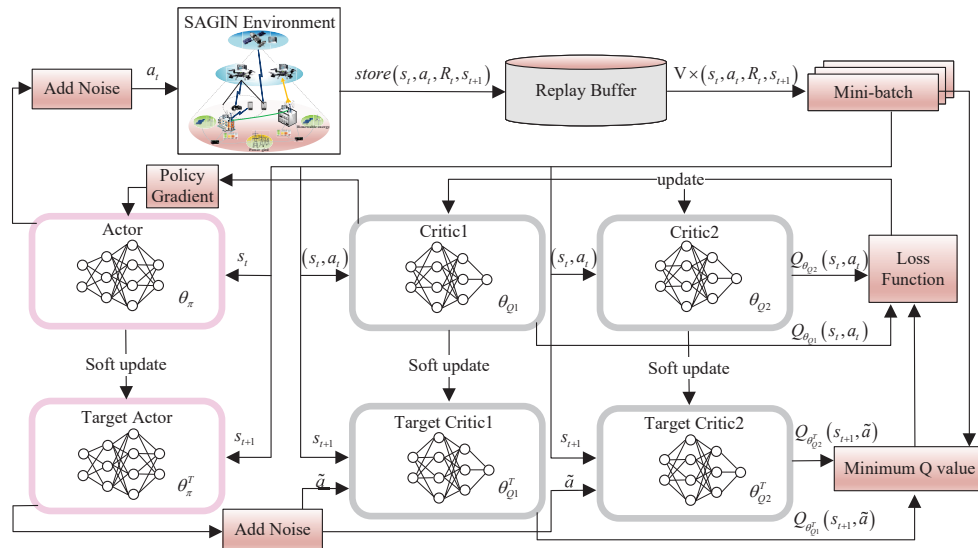


Figure 2. Structure of TD3PG algorithm.

The agent firstly observes the current state s_t and picks an action a_t via the actor network. After performing a_t in the environment, the agent is able to observe the immediate reward R_t and the next state s_{t+1} . Then, the agent will select the policy that maximizes the cumulative reward, which converts the state s_t to the action a_t . At this point, the transition $\langle s_t, a_t, R_t, s_{t+1} \rangle$ is obtained and stored in the replay memory for further sampling.

The training process of the TD3PG algorithm is summarized as Algorithm 1.

Algorithm 1 TD3PG algorithm.**Require:** Training episode length: $max\text{-}episode$. discount factor: γ .**Ensure:** Loss, gradients.

```

1: Initialize the actor network  $\pi_{\theta_\pi}$  with parameter  $\theta_\pi$ . Initialize the critic network  $Q_{\theta_{Q1}}, Q_{\theta_{Q2}}$  with
   parameters  $\theta_{Q1}$  and  $\theta_{Q2}$ , respectively.
2: Initialize the target network by:  $\theta_\pi^T \leftarrow \theta_\pi, \theta_{Q1}^T \leftarrow \theta_{Q1}, \theta_{Q2}^T \leftarrow \theta_{Q2}$ .
3: Initialize replay buffer  $\mathcal{R}$ .
4: for episode = 1 to  $max\text{-}episode$  do
5:   Reset simulation parameters and obtain initial observation state  $s$ .
6:   for  $t = 1$  to  $T$  do
7:     Get action after adding exploration noise  $a_t \sim \pi(s_t|\theta_\pi) + N(0, \delta)$ .
8:     Obtain the reward  $R_t$  and update  $s_{t+1}$ .
9:     Store the transition  $(s_t, a_t, R_t, s_{t+1})$  in replay buffer  $\mathcal{R}$ .
10:    Sample a random mini-batch of  $V$  tuples from replay buffer  $\mathcal{R}$ .
11:    Compute greedy actions for next states through  $\tilde{a} \leftarrow \pi(s_{t+1}|\theta_\pi^T) + \epsilon$ , where  $\epsilon \sim$ 
       clip( $N(0, \tilde{\omega}), -c, c$ ).
12:    Compute target value  $y_t$  based on Eq. (31).
13:    Update the critics by minimizing the loss function Eq. (32).
14:    Update the actor network parameter  $\theta_\pi$  by using the sampled policy gradient Eq. (34).
15:    Update the target networks Eq. (35).
16:   end for
17: end for

```

The minimum Q-value estimate of these two critic networks is used to calculate the target value y_t , which can be obtained by

$$y_t = R_t + \gamma \cdot \min_{i=1,2} Q_{Q_i}(s_{t+1}, \pi(s_{t+1}|\theta_\pi^T)|\theta_{Q_i}^T), \quad (31)$$

where γ is a discount factor, ranging from 0 to 1.

The loss function of two critic networks can be defined as

$$Ls(\theta_{Q_i}) = E[(y_t - Q_{Q_i}(s_t, a_t|\theta_{Q_i}))^2], i = 1, 2. \quad (32)$$

Then the gradient of loss function is calculated as

$$\nabla_{\theta_{Q_i}} Ls(\theta_{Q_i}) = E[y_t - (Q(s_t, a_t|\theta_\mu)) \nabla_{\theta_{Q_i}} Q(s_t, a_t|\theta_{Q_i})]. \quad (33)$$

The policy gradient of the actor network can be given as

$$\nabla_{\theta_\pi} J \approx E[\nabla_a \min_{i=1,2} Q(s_t, a_t|\theta_{Q_i})|_{a_t=\pi(s_t)} \nabla_{\theta_\pi} \pi(s_t|\theta_\pi)]. \quad (34)$$

The parameters of the target networks will be updated by means of

$$\begin{aligned} \theta_\pi^T &\leftarrow \omega \theta_\pi + (1 - \omega) \theta_\pi^T \\ \theta_{Q_i}^T &\leftarrow \omega \theta_{Q_i} + (1 - \omega) \theta_{Q_i}^T, i = 1, 2, \end{aligned} \quad (35)$$

where ω is a step size coefficient and satisfies $\omega \in [0, 1]$.

4.3. Complexity Analysis

The proposed TD3PG algorithm contains both actor network and critic network. Therefore, the complexities of them are evaluated separately. Assuming that the actor network has L^a layers with n_l neuron nodes in the l -th layer. Then the complexity of the l -th layer is $\mathcal{O}(n_{l-1}n_l + n_l n_{l+1})$ [37]. The complexity of the actor network is $\mathcal{O}(\sum_{l=2}^{L^a-1} (n_{l-1}n_l + n_l n_{l+1}))$. Assuming that the critic network has L^c layers with m_l neuron nodes in the j -th layer, the complexity of the j -th layer is $\mathcal{O}(m_{j-1}m_j + m_j m_{j+1})$ [37]. Similarly, the complexity of the critic network is $\mathcal{O}(\sum_{j=2}^{L^c-1} (m_{j-1}m_j + m_j m_{j+1}))$. Hence, the overall computational complexity of the TD3PG algorithm is $\mathcal{O}(\sum_{l=2}^{L^a-1} (n_{l-1}n_l + n_l n_{l+1}) +$

$\sum_{j=2}^{L^c-1} (m_{j-1}m_j + m_jm_{j+1}))$, which is similar to that of the DDPG algorithm. In general, TD3PG algorithm is an improvement for DDPG algorithm, but it does not increase the complexity much.

5. Results and Discussion

In this section, numerical simulations are used to evaluate the performance of the proposed TD3PG algorithm. Specifically, the simulation settings and benchmark strategies are firstly elaborated. Afterwards, the analysis of test results is conducted.

5.1. Simulation Settings

We consider an experimental area of $800\text{m} \times 800\text{m}$ in which $M = 50$ devices are uniformly distributed. The data size and latency requirements of each computation subtask obey a uniform distribution. Each device has a subtask in each time slot with the data size is randomly distributed in $U[3, 8]$ MB. The latency requirements of the subtasks are randomly distributed in $U[0.8, 1.2]$ s. There are $N = 2$ BSs, $K = 3$ UAVs and a LEO satellite in the experimental area, where the fixed altitude of the UAV is $H = 100$ m. The bandwidths of the UAV and BS are 5 and 20MHz, respectively. Corresponding simulation parameters are illustrated in Table 1.

Table 1. Simulation Parameters.

Parameter	Value
Weight of UAV (m_k)	9.65kg
Safety distance among UAVs (D_{\min})	20m
Maximum speed of UAV (v_{\max})	30m/s
Path loss index [38] (α_0)	2
Device upload power (p_m)	100mW
CPU cycles required for each bit of data (e_m)	800 cycles/bit
Effective switched capacitance (κ)	$10^{-27}\text{J/Hz}^3/\text{s}$
Energy conservation efficiency (χ)	0.9
Antenna power gains of the BS transmitter (G_n^t)	41dB
Antenna power gains of the UAV receiver (G_k^r)	41dB
Blade profile power in hovering status (P_0)	79.86W
Induced power in hovering status (P_i)	92.48W
Tip speed of the rotor blade (U_{tip})	120m/s
Mean rotor induced velocity in hover (v_0)	4.03m/s
Air density (ρ)	1.225kg/m^3
Rotor disc area (A)	0.503m^2
Fuselage drag ratio (d_0)	0.6
Rotor solidity (s)	0.05
Loss factor for energy transfer (η)	0.1
Energy consumption per unit of computation resource for LEO satellite (e_{sat})	1W/GHz
Energy consumption per unit of computation resource for UAV (e_{uav})	1.5W/GHz

To evaluate the performance of our proposed algorithm, we compare the proposed TD3PG algorithm with three conventional benchmarks.

- *DDPG Algorithm*: DDPG also contains two types of neural networks, an actor network and a critic network, but only one critic network.
- *Full Offload*: Tasks are all offloaded to the computation server for computation, and the computation server distributes computation resources evenly based on the number of connected devices.
- *Greedy on Edge Algorithm (Greedy-edge)*: Since edge computing has a low computation cost, the computation resources of the UAV can be fully utilized, resembling the “Greedy on edge”

algorithm in literature [39]. In time slot t , 60% of the total tasks are offloaded to the UAV, which distributes computation resources evenly based on the number of connected devices.

5.2. Results and Analysis

As shown in Figure 3, combinations of four learning rates are presented to evaluate the impact of different learning rates. With the increase of episodes, the normalized cumulative reward value fluctuates with different learning rates. Until around 2000 episodes, the four cases of normalized cumulative reward have different convergence trends. The performance of the proposed algorithm is the best when $\alpha_\pi = 0.0001$ and $\alpha_\mu = 0.0001$, due to highest convergence reward curve among four cases.

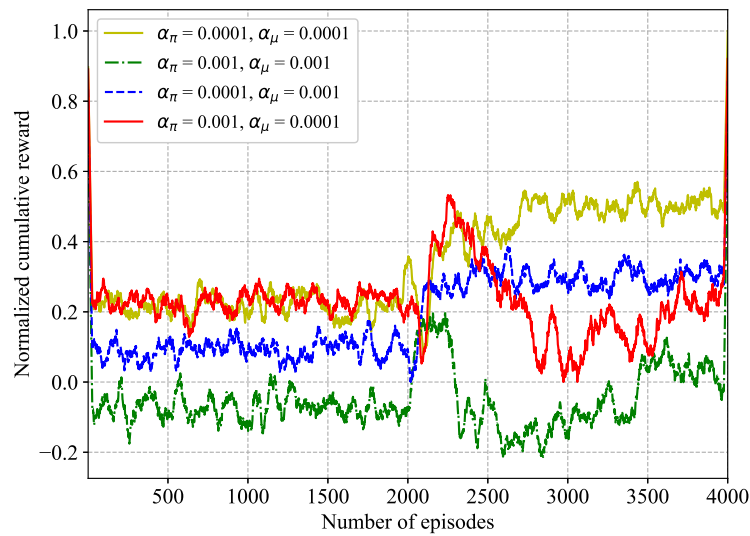


Figure 3. Impact of learning rate over training episodes.

In Figure 4, the impacts of different discount factors are evaluated. From the convergence results, the normalized cumulative reward in four cases gradually become smaller as γ keeps increasing. In particular, compared with $\gamma = 0.69, 0.79$, and 0.89 , higher performance is exhibited when $\gamma = 0.99$. Hence, $\gamma = 0.99$ is the best discount factor among the proposed different discount factors.

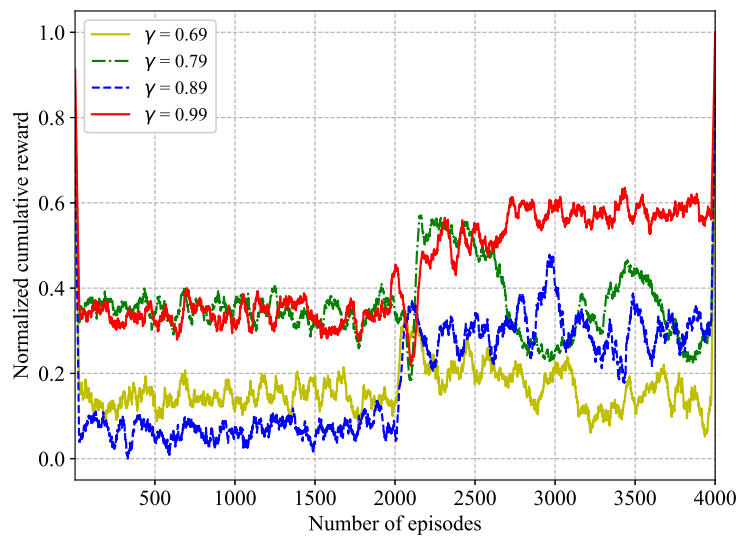


Figure 4. Impact of discount factor over training episodes.

Figure 5 shows the variation of the network cost function with the number of devices under different benchmark algorithms. Both the proposed algorithm and DDPG algorithm demonstrate the good performance in terms of joint latency-oriented, energy consumption and carbon emission. Specifically, the value of the network cost function for both algorithms are always lower than other benchmarks for different number of devices and increases at a low rate as the number of devices increases. This is due to the fact that the proposed algorithm and the DDPG algorithm dynamically adjust the offloading strategy, optimize the trajectory of the UAV and the sharing ratio of the RE in real time, ultimately obtaining a lower value of the cost function. In particular, the cost function under the TD3PG algorithm is lower and shows better performance compared to the DDPG algorithm.

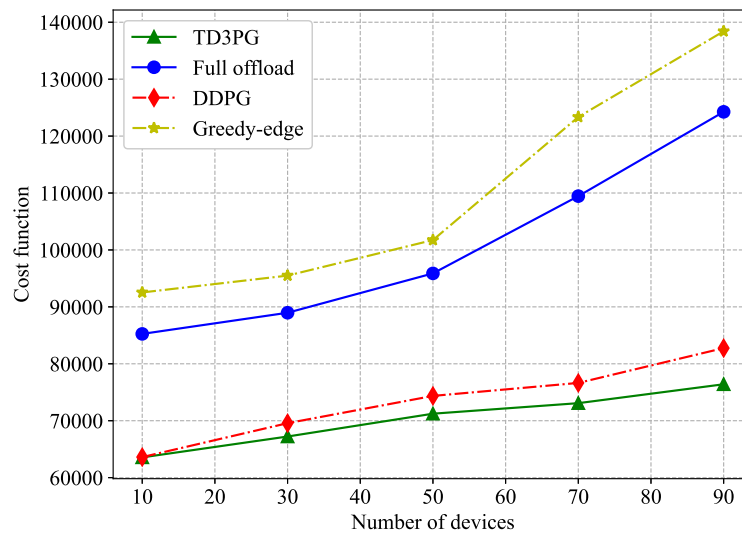


Figure 5. Comparison of cost function with respect to the number of devices.

In Figure 6, with the exception of full offload, the total energy consumption under other benchmark strategies increases with the maximum computation capacity of the device. Due to the ability of the proposed algorithm and DDPG algorithm with dynamically adjusting the local computation resources according to the reward function, the total energy consumption has great performance when f_{max} is low. In particular, the proposed algorithm shows the better performance with f_{max} between 2.0GHz and 2.5GHz. However, the full offload policy offloads all tasks to the computation server with no significant change in total energy consumption.

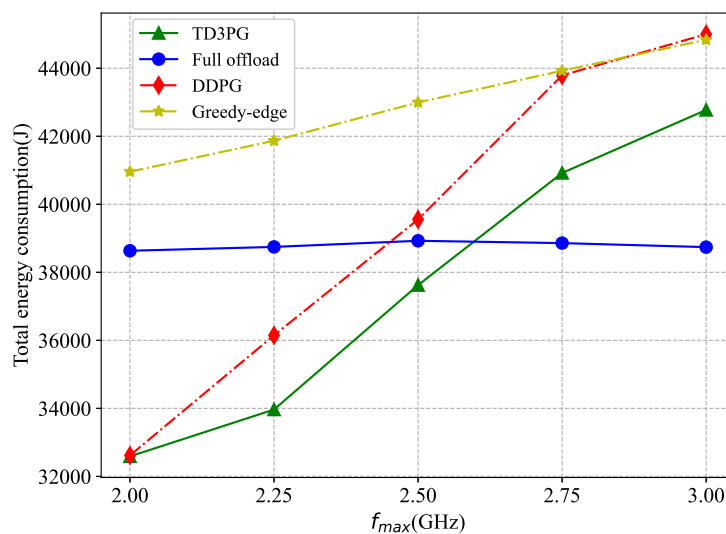


Figure 6. Comparison of total energy consumption for device with maximum computation capacity.

Figure 7 compares the variation of the total network energy consumption with different energy consumption per unit of computation resource for LEO satellite e_{sat} under different benchmark algorithms. With the increase of e_{sat} , the total energy consumption under the proposed algorithm is always the lowest. The total energy consumption of the network under the greedy-edge algorithm is the highest. On the one hand, it has no better offloading decision. On the other hand, in terms of energy consumption, the devices unable to intelligently change the computation resource, resulting in increased energy consumption for local computation. Under the full offload benchmark algorithm, more tasks are offloaded to the LEO satellite for execution and the total energy consumption grows more significantly when e_{sat} increases. The DDPG algorithm enables to dynamically adjust task offloading according to the reward value, with the rate of increase in total network energy consumption gradually decreasing as e_{sat} grows. The high altitude of LEO satellite lead to more transmission latency. So a very little tasks will be offloaded to it. The variation in total network energy consumption with the proposed algorithm is slight when e_{sat} increases. Similarly, the proposed algorithm demonstrates excellent performance as the energy consumption per unit computation resource for UAV e_{uav} increases.

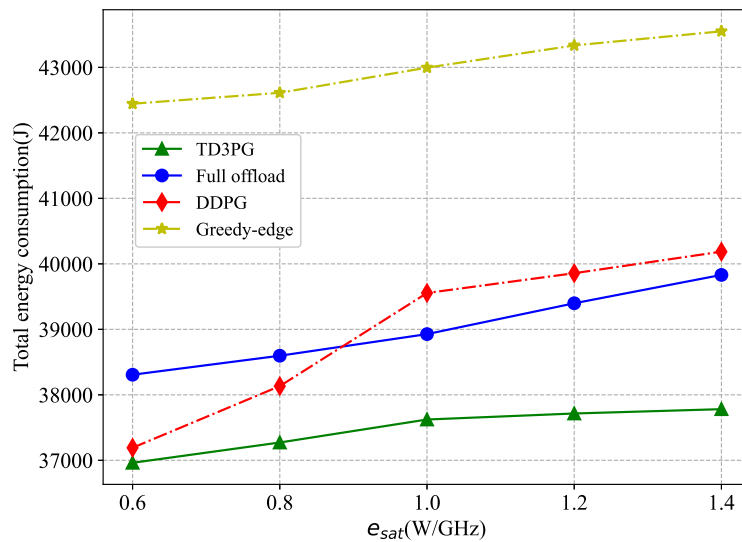


Figure 7. Comparison of the total energy consumption with the energy consumption per unit computation resource for LEO satellite.

In order to illustrate the performance of the proposed algorithm in optimizing the UAV trajectory to reduce latency, three cases with different benchmark algorithms are analyzed. These three cases are trajectory optimization, trajectory fixed in one direction, and UAV flying in place, respectively. As shown in Figure 8, the lowest overall latency of the three cases is trajectory optimization, which reduces the overall latency by 2.93% compared to UAV flying in place and 5.78% compared to the overall time reduction for trajectory fixed in one direction. The total task latency under both the proposed algorithm and the DDPG algorithm is lower in each flight trajectory mode, significantly improving the latency/QoS. The greedy-edge and full offload algorithm show poor performance. The greedy-edge algorithm offloads a higher percentage of tasks to UAV, resulting in a higher task latency. Meanwhile, the local computation resources are not the maximum computation capability, which leads to longer computation time, which explains the lower total task latency of the full offload compared to the greedy-edge algorithm.

Since the UAV have limited battery, the remaining power of the UAV needs to be maintained as much as possible to ensure the sustainable operation of the network. Figure 9 shows the average residual power of the UAV at each time slot for different benchmark algorithms. Among these benchmark algorithms, both the proposed algorithm and DDPG algorithm employ WPT, which ensure that the UAV always has remaining power to maintain operation. However, the other two do not

have WPT, so the power is quickly depleted and unable to maintain the sustainable operation of the network. It is worth noting that the energy consumed by UAV computation is not large compared to the energy consumed by UAV flight. Hence, the difference in the average remaining power of UAV under the remaining three benchmark algorithms is small. The greedy-edge algorithm and the full offload policy offload a large of tasks to the computation server, so the average remaining power of the UAV is lower. The proposed algorithm, on the other hand, integrates dynamic task offloading with UAV trajectory optimization, which not only maximize the relative residual power of UAV, but also enables UAV to replenish power through WPT when the power of UAV falls below a threshold value, maintaining sustainable network operation.

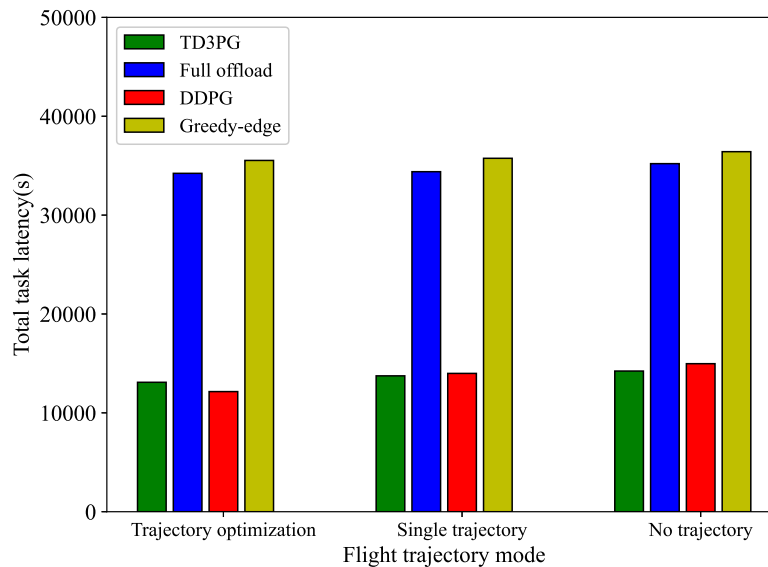


Figure 8. Comparison of total task latency with three trajectory settings.

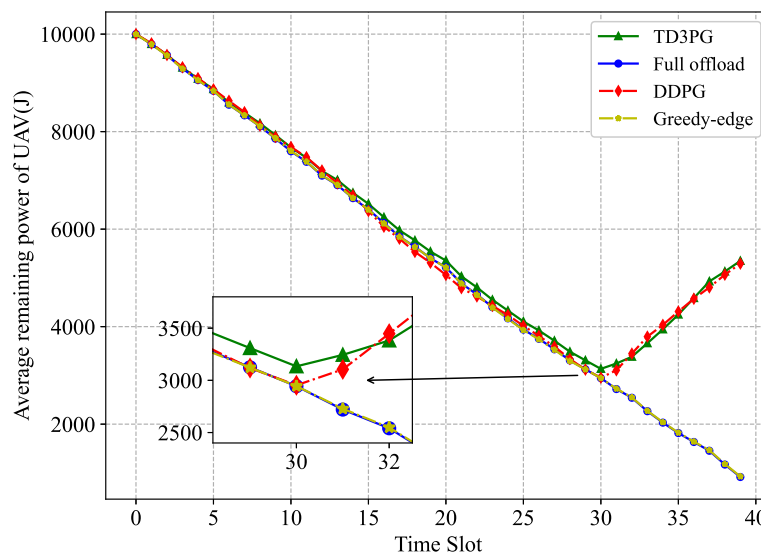


Figure 9. Comparison of average remaining power of UAV in each time slot.

In order to evaluate the performance of the proposed algorithm in reducing the carbon emissions of the network, Figure 10 shows the average loss ratio of RE for different loss factors of energy transmission. The smaller the average loss rate of RE is, the more RE is utilized. And the more carbon emissions are reduced. As the two algorithms can intelligently optimize the sharing proportion $v_n(t)$ of RE, the proposed algorithm shows better performance compared to the DDPG algorithm. It is worth

noting that the average loss ratio of RE under the DDPG algorithm at $\eta = 0.05$ is higher than that $\eta = 0.1$. It may be that too much energy is carried over when too little energy is shared. When RE arrives in the next time slot, the total remaining energy exceeds the backup battery capacity, which leads to an increase in the amount of RE $e_n^{exc}(t)$ that is discarded.

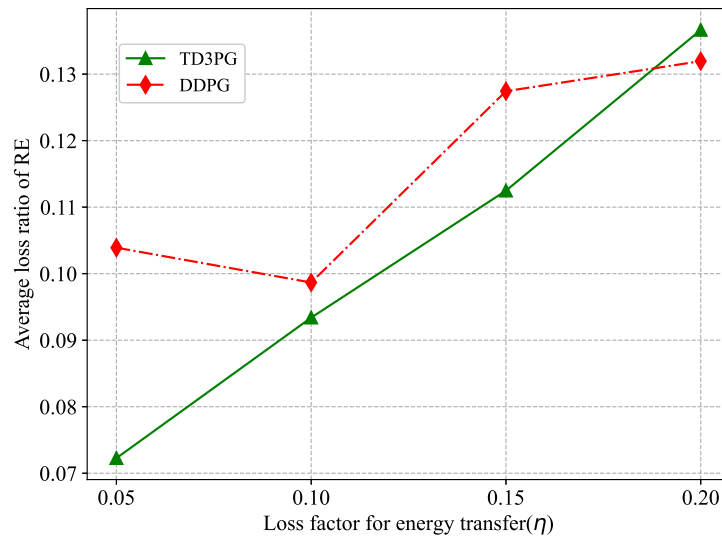


Figure 10. Comparison of average loss ratio of RE with the loss factor η .

6. Conclusions

In this article, a SAGIN scenario with new designed power technology is proposed firstly. A cost function is formulated with joint latency/QoS, green communication and peak carbon dioxide. Specifically, the cost function consists of task latency, energy consumption, and the carbon emissions of the BSs. To maintain the flight and computation energy consumption of the UAV, BSs explore WPT to power UAV to ensure the sustainability of the network. Then a TD3PG method is proposed to optimizing the cost function. Finally, the proposed algorithm is compared with three benchmark algorithms. Simulation results show that the approach improves the sustainability of the network, while offering advantages in terms of task latency and energy saving. By optimizing the UAV trajectory under this algorithm, the total task latency is reduced by 2.93% and 5.78% compared to single trajectory and no trajectory, respectively. The proposed algorithm also shows better performance when it comes to reducing the carbon emissions of the network. In the future, software defined network will be utilized to better optimize the energy resources and wireless resources in SAGIN.

Author Contributions: Y.W.: methodology, software, writing. B.L.: supervision, writing—reviewing and editing. J.H.: validation. J.D.: supervision. Y.L.: editing, Y.L.: original draft preparation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ningxia Autonomous Region key R&D plan project (general project) (2020BDE03006), Ningxia Natural Science Foundation (2019AAC03072), and the National Natural Science Foundation of China (61301145).

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

References

1. Cheng, N.; Quan, W.; Shi, W.; Wu, H.; Ye, Q.; Zhou, H.; Zhuang, W.; Shen, X.; Bai, B. A Comprehensive Simulation Platform for Space-Air-Ground Integrated Network. *IEEE Wireless Communications* **2020**, *27*, 178–185.
2. Wang, Y.; Su, Z.; Ni, J.; Zhang, N.; Shen, X. Blockchain-Empowered Space-Air-Ground Integrated Networks: Opportunities, Challenges, and Solutions. *IEEE Communications Surveys & Tutorials* **2022**, *24*, 160–209.

3. Wang, C.; Yu, X.; Xu, L.; Jiang, F.; Wang, W.; Cheng, X. QoS-aware offloading based on communication-computation resource coordination for 6G edge intelligence. *China Communications* **2023**, *20*, 236–251.
4. Feng, X.; Ling, X.; Zheng, H.; Chen, Z.; Xu, Y. Adaptive Multi-Kernel SVM With Spatial-Temporal Correlation for Short-Term Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems* **2019**, *20*, 2001–2013.
5. Zhao, B.; Cui, Q.; Liang, S.; Zhai, J.; Hou, Y.; Huang, X.; Pan, M.; Tao, X. Green concerns in federated learning over 6G. *China Communications* **2022**, *19*, 50–69.
6. Cheshmehzangi, A.; Chen, H., Key Suggestions and Steps Ahead for China's Carbon Neutrality Plan. In *China's Sustainability Transitions: Low Carbon and Climate-Resilient Plan for Carbon Neutral 2060*; Springer Singapore: Singapore, 2021; pp. 189–206.
7. Peng, H.; Shen, X. Deep Reinforcement Learning Based Resource Management for Multi-Access Edge Computing in Vehicular Networks. *IEEE Transactions on Network Science and Engineering* **2020**, *7*, 2416–2428.
8. Zhao, J.; Li, Q.; Gong, Y.; Zhang, K. Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks. *IEEE Transactions on Vehicular Technology* **2019**, *68*, 7944–7956.
9. Abderrahim, W.; Amin, O.; Alouini, M.S.; Shihada, B. Latency-Aware Offloading in Integrated Satellite Terrestrial Networks. *IEEE Open Journal of the Communications Society* **2020**, *1*, 490–500.
10. Dai, Y.; Zhang, K.; Maharjan, S.; Zhang, Y. Edge Intelligence for Energy-Efficient Computation Offloading and Resource Allocation in 5G Beyond. *IEEE Transactions on Vehicular Technology* **2020**, *69*, 12175–12186.
11. Wang, F.; Xing, H.; Xu, J. Real-Time Resource Allocation for Wireless Powered Multiuser Mobile Edge Computing With Energy and Task Causality. *IEEE Transactions on Communications* **2020**, *68*, 7140–7155.
12. Chen, M.; Hao, Y. Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network. *IEEE Journal on Selected Areas in Communications* **2018**, *36*, 587–597.
13. Guo, Y.; Li, Q.; Li, Y.; Zhang, N.; Wang, S. Service Coordination in the Space-Air-Ground Integrated Network. *IEEE Network* **2021**, *35*, 168–173.
14. Zhang, L.; Abderrahim, W.; Shihada, B. Heterogeneous Traffic Offloading in Space-Air-Ground Integrated Networks. *IEEE Access* **2021**, *9*, 165462–165475.
15. Chen, M.; Hao, Y. Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network. *IEEE Journal on Selected Areas in Communications* **2018**, *36*, 587–597.
16. Buzzi, S.; I, C.L.; Klein, T.E.; Poor, H.V.; Yang, C.; Zappone, A. A Survey of Energy-Efficient Techniques for 5G Networks and Challenges Ahead. *IEEE Journal on Selected Areas in Communications* **2016**, *34*, 697–709.
17. Alqasir, A.M.; Kamal, A.E. Cooperative Small Cell HetNets With Dynamic Sleeping and Energy Harvesting. *IEEE Transactions on Green Communications and Networking* **2020**, *4*, 774–782.
18. Malta, S.; Pinto, P.; Fernández-Veiga, M. Using Reinforcement Learning to Reduce Energy Consumption of Ultra-Dense Networks With 5G Use Cases Requirements. *IEEE Access* **2023**, *11*, 5417–5428.
19. Yuan, H.; Tang, G.; Guo, D.; Wu, K.; Shao, X.; Yu, K.; Wei, W. BESS Aided Renewable Energy Supply Using Deep Reinforcement Learning for 5G and Beyond. *IEEE Transactions on Green Communications and Networking* **2022**, *6*, 669–684.
20. Nabavi, S.A.; Motlagh, N.H.; Zaidan, M.A.; Aslani, A.; Zakeri, B. Deep Learning in Energy Modeling: Application in Smart Buildings With Distributed Energy Generation. *IEEE Access* **2021**, *9*, 125439–125461.
21. Rubina Aktar, M.; Shamim Anower, M.; Zahurul Islam Sarkar, M.; Sayem, A.S.M.; Rashedul Islam, M.; Akash, A.I.; Rumana Akter Rume, M.; Moloudian, G.; Lalbakhsh, A. Energy-Efficient Hybrid Powered Cloud Radio Access Network (C-RAN) for 5G. *IEEE Access* **2023**, *11*, 3208–3220.
22. Saad, W.; Bennis, M.; Chen, M. A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Network* **2020**, *34*, 134–142.
23. Wang, F.; Xu, J.; Wang, X.; Cui, S. Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems. *IEEE Transactions on Wireless Communications* **2018**, *17*, 1784–1797.
24. Zheng, K.; Jiang, G.; Liu, X.; Chi, K.; Yao, X.; Liu, J. DRL-Based Offloading for Computation Delay Minimization in Wireless-Powered Multi-Access Edge Computing. *IEEE Transactions on Communications* **2023**, *71*, 1755–1770.
25. Zheng, K.; Jia, X.; Chi, K.; Liu, X. DDPG-Based Joint Time and Energy Management in Ambient Backscatter-Assisted Hybrid Underlay CRNs. *IEEE Transactions on Communications* **2023**, *71*, 441–456.

26. Liu, J.; Zhao, X.; Qin, P.; Geng, S.; Meng, S. Joint Dynamic Task Offloading and Resource Scheduling for WPT Enabled Space-Air-Ground Power Internet of Things. *IEEE Transactions on Network Science and Engineering* **2022**, *9*, 660–677.
27. Sun, L.; Pang, H.; Gao, L. Joint Sponsor Scheduling in Cellular and Edge Caching Networks for Mobile Video Delivery. *IEEE Transactions on Multimedia* **2018**, *20*, 3414–3427.
28. Yong, P.; Zhang, N.; Hou, Q.; Liu, Y.; Teng, F.; Ci, S.; Kang, C. Evaluating the Dispatchable Capacity of Base Station Backup Batteries in Distribution Networks. *IEEE Transactions on Smart Grid* **2021**, *12*, 3966–3979.
29. Hu, Q.; Cai, Y.; Yu, G.; Qin, Z.; Zhao, M.; Li, G.Y. Joint Offloading and Trajectory Design for UAV-Enabled Mobile Edge Computing Systems. *IEEE Internet of Things Journal* **2019**, *6*, 1879–1892.
30. Hong, T.; Zhao, W.; Liu, R.; Kadoch, M. Space-Air-Ground IoT Network and Related Key Technologies. *IEEE Wireless Communications* **2020**, *27*, 96–104.
31. Mao, S.; He, S.; Wu, J. Joint UAV Position Optimization and Resource Scheduling in Space-Air-Ground Integrated Networks With Mixed Cloud-Edge Computing. *IEEE Systems Journal* **2021**, *15*, 3992–4002.
32. Lakew, D.S.; Tran, A.T.; Dao, N.N.; Cho, S. Intelligent Offloading and Resource Allocation in Heterogeneous Aerial Access IoT Networks. *IEEE Internet of Things Journal* **2023**, *10*, 5704–5718.
33. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials* **2017**, *19*, 2322–2358.
34. Zhan, C.; Zeng, Y. Aerial-Ground Cost Tradeoff for Multi-UAV-Enabled Data Collection in Wireless Sensor Networks. *IEEE Transactions on Communications* **2020**, *68*, 1937–1950.
35. Huang, H.; Ye, Q.; Zhou, Y. 6G-Empowered Offloading for Realtime Applications in Multi-Access Edge Computing. *IEEE Transactions on Network Science and Engineering* **2023**, *10*, 1311–1325.
36. Wang, Y.; Gao, Z.; Zhang, J.; Cao, X.; Zheng, D.; Gao, Y.; Ng, D.W.K.; Renzo, M.D. Trajectory Design for UAV-Based Internet of Things Data Collection: A Deep Reinforcement Learning Approach. *IEEE Internet of Things Journal* **2022**, *9*, 3899–3912.
37. Gao, G.; Li, J.; Wen, Y. DeepComfort: Energy-Efficient Thermal Comfort Control in Buildings Via Reinforcement Learning. *IEEE Internet of Things Journal* **2020**, *7*, 8472–8484.
38. Zhao, X.; Du, F.; Geng, S.; Fu, Z.; Wang, Z.; Zhang, Y.; Zhou, Z.; Zhang, L.; Yang, L. Playback of 5G and Beyond Measured MIMO Channels by an ANN-Based Modeling and Simulation Framework. *IEEE Journal on Selected Areas in Communications* **2020**, *38*, 1945–1954.
39. Cheng, N.; Lyu, F.; Quan, W.; Zhou, C.; He, H.; Shi, W.; Shen, X. Space/Aerial-Assisted Computing Offloading for IoT Applications: A Learning-Based Approach. *IEEE Journal on Selected Areas in Communications* **2019**, *37*, 1117–1129.

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.