Computation Offloading and Resource Allocation in UAV-Assisted Satellite Network Systems

Bintao Hu*, Yuan Gao†, Miguel López-Benítez¶, Jianbo Du‡, Jie Zhang§, Xiaoli Chu§
*School of Internet of Things, Xi'an Jiaotong-Liverpool University, Suzhou, China
†School of Communication and Information Engineering, Shanghai University, Shanghai, China
¶Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK
¶ARIES Research Centre, Antonio de Nebrija University, Madrid, Spain
‡School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China
§Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, UK
Email: *Bintao.Hu@xjtlu.edu.cn, †gaoyuansie@shu.edu.cn, ¶m.lopez-benitez@liverpool.ac.uk,
‡dujianboo@163.com,§jie.zhang@sheffield.ac.uk, x.chu@sheffield.ac.uk

***Abstract*—In order to support the Internet of Things (IoT) systems, it is necessary to allow IoT user equipment (UE) to offload its delay-sensitive computational tasks to a server both UAV-assisted networks and satellite networks have been considered for computation offloading but mostly studied separately. In this paper, we investigate computation offloading and resource allocation in a UAV-assisted satellite network, where an IoT UE can offload its task to a UAV-carried edge server or a satellite edge server. To minimise the maximum total service delay of all IoT UEs, which includes the transmission delay and processing delay, we propose a Q-learning-based task offloading decision optimisation algorithm in conjunction with the computation resource allocation strategies for the UAV-assisted satellite network. Simulation results show that our proposed optimisation algorithm achieves a much lower total service delay of all IoT UEs than the benchmarks.**

***Index Terms*—Internet of Things, resource allocation, Q-learning, edge computing, U2X communications, satellite networks**

I. Introduction

With the continuous expansion of wireless communications activity space and the intelligent service requirements, the new Internet of Things (IoT) network technology assisted by satellite and 6G wireless communications technologies will gradually become the main body of future academia and industrial needs [1]. Industry 4.0 is one of the main applications of IoT that may spawn tremendous and ubiquitous high-data-rate computational tasks, which will lead to complex interference management and high-latency task delivery and processing [2]. Moreover, most industry 4.0 user equipment (UEs) may repeatedly create requests and delay-sensitive tasks, which results in a mass of redundant content transmission over the traditional IoT network [3].

To overcome the aforementioned challenges, mobile edge computing (MEC) and Low Earth Orbit (LEO) satellite network technologies have appeared as potential supplement paradigms to the existing cloud computing technology-based IoT networks in industry and academia [4]. One of the advantages of the traditional Cloud-MEC hierarchy is that a response to tasks from resources close to the end UE is fast and can satisfy UE low-latency requirements, while still having access to computing resources at the Cloud servers. Thus, based on industry 4.0 IoT applications' characteristics, like the frequently requested industry functional contents, UEs are able to send requests to the central controller, and fetch the requested content from the closest linked MEC server, rather than from the remote cloud server through backhaul links [5]. In addition, Artificial intelligence (AI)-empowered satellite communication technologies allow for the automated processing and analysis of large amounts of generated raw data before being transmitted to the ground, to improve the overall efficiency of the network and reduce data transmission consumption and additional storage usage [6].

In practical satellite communications scenarios, such as unmanned aerial vehicle (UAV)-assisted satellite communication scenarios, how to design an appropriate optimisation mechanism for the offloading decisions in conjunction with resource allocation is a challenge [7]. Kim *et al.* [8] proposed a satellite edge computing-based offloading strategy for an IoT sliced network to improve satellite constellation efficiency in terms of optimising proper satellite altitudes and satellite offloading rates. Song *et al.* [9] designed a novel terrestrial-satellite IoT-based MEC framework and proposed to minimise the latency and energy consumption among all IoT mobile devices by jointly optimising the offloading strategy and resource allocation. In [10], authors proposed a K-means-based clustering optimisation algorithm to optimise the resource allocation for all virtual machines (VMs) under the transmission and connection delays between the VM and the satellite in a software-defined networking (SDN) model-based satellite network. Leyva-Mayorga *et al.* [11] proposed a global optimisation algorithm to minimise satellite tasks processing energy consumption by jointly optimising the image task segment division and relevant bandwidth resource allocation. All the above-existing works on satellite-based edge computing frameworks and optimisation algorithms avoided the vast of raw data to be transmitted back to ground stations, which is able to reduce the data transmission and storage costs while improving the overall satellite network efficiency. However,

This work is supported in part by XJTLU Research Development Fund under Grant No. RDF-22-02-047, in part by the Natural Science Foundation of China under Grant 62271391, in part by the Serving Local Special Scientific Research Project of Education Department of Shaanxi Province under Grant 21JC032, and in part by the Innovation Program of Shanghai Municipa Science and Technology Commission under Grant 22511103202.

they have not sufficiently considered the long distance between satellites and all mobile devices that may increase the delay computation offloading.

Taking into account the long distance between satellites and mobile devices, UAVs could be considered as a flexible edge server to overcome this restriction. Lu *et al*. [12] designed a multi-UAV-assisted MEC framework and proposed a spiral placement algorithm and a reinforcement learning-based optimisation algorithm to maximise the system utility, which consists of energy consumption and task processing delays, in conjunction with optimising all UAVs' deployment and offloading decisions. Zhao *et al*. [13] proposed a cooperative multi-agent deep reinforcement learning optimisation algorithm to minimise the energy and execution delay consumptions of all UAVs by jointly optimising their trajectories and offloading decisions in a multi-UAV-assisted MEC system network. In [14], authors proposed two deep reinforcement learning-based optimisation algorithms for a UAV-aided edge computing network to maximise the average aggregate quality of experience (QoE) among all devices and optimise the trajectories of all UAVs, respectively. In [15], authors proposed a deep Q-learning algorithm for optimising offloading decisions for all users, while optimising the UAV hotspot selection under task priority considerations in a UAV mobile edge computing network. We note that, the above existing works have not sufficiently considered the coverage capacity and battery limitations of UAV-empowered edge servers, then some delay-sensitive tasks may not be available to be processed at the UAV-empowered edge server due to the limited computation resources capacity and the coverage capacity.

Inspired by the aforementioned works, we envision a UAV-assisted satellite edge intelligent computing model to minimise the maximum service delay consumption among all IoT UEs via optimising the task offloading decision in conjunction with optimising computation resources at the UAV and the satellite. The major contributions of this paper can be summarised as follows.

- A UAV-assisted satellite two-layer edge intelligent network model is proposed for the sake of minimising the maximum service delay consumption (i.e., transmission delay and processing delay) among all IoT UEs, which is to incorporate LEO and UAV-to-everything (U2X) communications for edge computing, it is beneficial to be conveniently and quickly deployed to provide low-latency and computation-intensive services for IoT UEs.
- A Q-learning-based UAV-assisted satellite network offloading decision optimisation (QUSNO) framework is proposed to optimise the offloading decisions among all IoT UEs while optimising the computation resources at the UAV-assisted and satellite edge servers for the edge computing IoT UEs, respectively.
- Massive simulation results show that the proposed QUSNO algorithm has superior performance in terms of service delay consumption reduction in comparison with the benchmarks (i.e., local-processing, UAV edge-processing, and satellite edge-processing), which illustrates the efficacy of the proposed QUSNO algorithm.

II. System Model

In this section, we first introduce the UAV-assisted satellite two-layer computation offloading framework. Then, the communication and computation models will be introduced respectively. Finally, we introduce the delay consumption model of each UE.

A. UAV-Assisted Satellite Network System Model

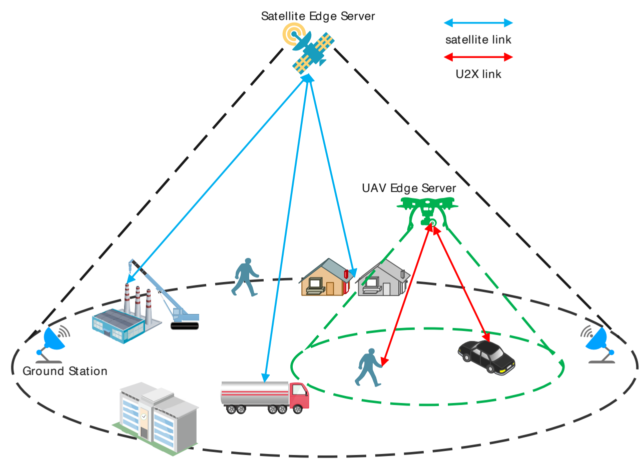

Fig. 1: An edge intelligent two-layer UAV-assisted satellite network system

As shown in Fig.1, we consider a UAV-assisted satellite network system consisting of two different layers, which include the IoT UE layer and the edge server layer. The IoT-based UE layer includes N UEs, which can be denoted by $\mathcal{N} = \{1, 2, 3, ..., N\}$. Different from the existing works [8]–[15], we consider the edge server layer has a lower edge layer and an upper edge layer. To reduce the system complexity, we consider that the lower edge layer is composed of one edge computing-empowered UAV and the upper edge layer consists of one LEO satellite with an edge computing server. For further research according to this architecture, more UAVs and satellites could be involved. Apart from IoT UEs and edge servers, we also consider a gateway, which is used to collect the requests from all IoT UEs and make offloading decisions. After that, the corresponding resources will be allocated accordingly. In addition, in our system, we consider a time slot mechanism, where the index and the length of a time slot can be denoted as t and Δt, and the set of time slot is denoted as $\mathcal{T} = \{1, 2, ..., T\}$, respectively.

The offloading decision for IoT UE n in the tth time slot is denoted by $x_n(t), y_n(t) \in \{0, 1\}$, where $x_n(t) = 1$ or $y_n(t) = 1$ indicate that the task is processed by IoT UE n itself or an edge server, respectively; otherwise $x_n(t) = 0$ or $y_n(t) = 0$; and we have

$$x_n(t) + y_n(t) = 1, \forall n \in N. \tag{1}$$

If the task of IoT UE n in time slot t is offloaded to an edge server for processing, which has two options and can be denoted by $e_n(t) \in \{0,1\}$, where $e_{n,u}(t) = 1$ indicates that task n is processed by the UAV-assisted edge server; otherwise $e_{n,s}(t) = 1$ indicates that task n will be processed by the satellite-based edge server, and we have

$$y_n(t)(e_{n,u}(t) + e_{n,s}(t)) = 1, \forall n \in N. \tag{2}$$

If the task of IoT UE n in time slot t is locally processed, and the processing delay of local processing can be expressed as

$$T_{n,l}(t) = \frac{D_n(t)C_n(t)}{f_n^l(t)}, \tag{3}$$

where $D_n(t)$ (in bits) is the input data size of the nth IoT UE task for local processing; $C_n(t)$ (in cycles/bit) denotes the required number of CPU cycles to process one-bit task; $f_n^l(t)$ (in CPU cycles/s) denotes the local processing capability of IoT UE n to process the task by itself.

B. Edge Processing Model

1) UAV-assisted edge processing: In edge computing mode, if the task of IoT UE n in time slot t is offloaded to the UAV-assisted edge server, i.e., $e_{n,u}(t) = 1$, then the input data of the nth task should be transmitted from the nth IoT UE to the UAV. Since the UAV flies to IoT UE n, there are Line-of-Sight (LoS) link and Non-Line-of-Sight (NLoS) link between the UAV and IoT UE n. The corresponding LoS probability is given by

$$P_{u,n}^{LoS}(t) = \frac{1}{1 + a\exp(-b(\theta - a))}, \tag{4}$$

where a and b denote positive constants that depend on the environment and the values are given in [16]; θ denotes the elevation angle of the UAV and it can be calculated by $\theta = \arctan(H_u^{pos}(t)/\sqrt{(x_u^{pos}(t) - x_n^{pos}(t))^2 + (y_u^{pos}(t) - y_n^{pos}(t))^2}$, $\forall n \in N$, where $H_u^{pos}(t), x_u^{pos}(t), x_n^{pos}(t), y_u^{pos}(t), y_n^{pos}(t)$ are the location coordinates of the UAV and the nth IoT UE, respectively.

According to equation (4), the path loss between the UAV and IoT UE n in time slot t can be expressed in dB as

$$\begin{aligned} PL_n^u(t) = 20log\left(\frac{4\pi f_c}{c} d_{n,u}\right) + P_{u,n}^{LoS}(t)\eta_{LoS} \\ +(1 - P_{u,n}^{LoS}(t))\eta_{NLoS}, \end{aligned} \tag{5}$$

where $d_{n,u}$ is the distance between the UAV and IoT UE n; η_{LoS} and η_{NLoS} (in dB) are the losses corresponding to the LoS link and NLoS link, respectively; f_c is the carrier frequency and c is the speed of light.

The achievable uplink data transmission rate in time slot t between IoT UE n and the UAV is given by

$$r_{n,u}(t) = W_n log_2\left(1 + \frac{p_n(t)10^{(-PL_n^u(t)/10)}}{N_0}\right), \tag{6}$$

where W_n is the channel bandwidth allocated by UAV to the nth IoT UE; $p_n(t)$ denotes the transmission power of IoT UE n; N_0 is the additive white Gaussian noise power.

The transmission delay of the task from IoT UE n to the UAV in time slot t is given by

$$T_{n,u}^{up}(t) = \frac{D_n(t)}{r_{n,u}(t)}. \tag{7}$$

After all the input data of the task has been fully received by the UAV, the processing delay for the task of IoT UE n at the UAV can be expressed by

$$T_{n,u}^{off}(t) = \frac{D_n(t)C_n(t)}{f_{n,u}^e(t)}, \tag{8}$$

where $f_{n,u}^e(t)$ (in cycles/s) is the allocated computation resources in time slot t at the UAV to IoT UE n.

Therefore, the total delay consumption of processing task n at the UAV-assisted edge server is given by

$$T_{n,u}(t) = T_{n,u}^{up}(t) + T_{n,u}^{off}(t). \tag{9}$$

2) Satellite-based edge processing: Similar to the UAV-assisted edge processing model, if the task of IoT UE n is offloaded to the satellite-based edge server in time slot t, i.e., $e_{n,s}(t) = 1$, then the input data of the nth task should be transmitted from the nth IoT UE to the satellite. The satellite channel mainly consists of free-space path loss in time slot t which can be expressed in dB as [9]

$$PL_n^s(t) = 92.44 + 20log_{10}(H_s(t)) + 20log_{10}(f_s(t)), \tag{10}$$

where $H_s(t)$ (in km) is the height of the satellite edge server, and $f_s(t)$ (in GHz) is the satellite system operating frequency. The achievable uplink data rate from IoT UE n to the satellite in time slot t can be presented as

$$r_{n,s}(t) = W_n log_2\left(1 + \frac{p_n(t)g_{n,s}(t)}{N_0}\right) \tag{11}$$

where W_n is the channel bandwidth allocated by satellite to the nth IoT UE; $g_{n,s}(t)$ denotes the channel gain between the satellite and IoT UE n, where the fast fading follows complex Gaussian distribution $\mathcal{CN}(0,1)$ and the shadowing model follows log-normal distribution $\mathcal{C}(0,8)$ [9].

The transmission delay of the task from the nth IoT UE to the satellite in time slot t is given by

$$T_{n,s}^{up}(t) = \frac{D_n(t)}{r_{n,s}(t)}. \tag{12}$$

and the processing delay for the task n at the satellite can be expressed by

$$T_{n,s}^{off}(t) = \frac{D_n(t)C_n(t)}{f_{n,s}^e(t)}, \tag{13}$$

where $f_{n,s}^e(t)$ (in cycles/s) denotes the allocated computation resources at the satellite to the nth IoT UE.

Therefore, the total delay consumption of processing task n at the satellite-based edge server is given by

$$T_{n,s}(t) = T_{n,s}^{up}(t) + T_{n,s}^{off}(t). \tag{14}$$

III. Problem Formulation and Proposed Algorithm

In this section, we first formulate the minimised total delay consumption problem, then propose a Q-learning-based UAV-assisted satellite network offloading decision optimisation (QUSNO) algorithm to solve the optimisation problem.

A. Problem Formulation

We propose to minimise the maximum total delay consumption among all IoT UEs while optimising the offloading decisions $\mathbf{I} = \{I_n(t) = (x_n(t), y_n(t)e_{n,u}(t), y_n(t)e_{n,s}(t)), \forall n \in \mathcal{N}, \forall t \in \mathcal{T}\}$, computation resource allocation $\mathbf{f^e} = \{f_{n,u}^e(t), f_{n,s}^e(t), \forall t \in \mathcal{T}\}$ at edge servers. Then, the joint optimisation problem can be formulated as follows

$$\mathcal{P}: \min_{\mathbf{I},\mathbf{f^e}} \max_{n\in\mathcal{N}} T_n(t) \tag{15}$$

$$\begin{aligned}
s.t.\quad & x_n(t), y_n(t), e_{n,u}(t), e_{n,s}(t), \in \{0,1\},\ \forall n \in \mathcal{N}, && \text{(15a)}\\
& x_n(t) + y_n(t)(e_{n,u}(t) + e_{n,s}(t)) = 1,\ \forall n \in \mathcal{N}, && \text{(15b)}\\
& \sum_{n\in\mathcal{N}_1} y_n(t)e_{n,u}(t)f_{n,u}^e(t) \le F_u^e, && \text{(15c)}\\
& \sum_{n\in\mathcal{N}_1} y_n(t)e_{n,s}(t)f_{n,s}^e(t) \le F_s^e, && \text{(15d)}\\
& 0 \le f_n^l(t) \le f_{n,u}^e(t) \le f_{n,s}^e(t),\ \forall n \in \mathcal{N}, && \text{(15e)}\\
& p_n(t) \le p_{max},\ \forall n \in \mathcal{N} && \text{(15f)}\\
& T_n(t) \le \tau_n(t),\ \forall n \in \mathcal{N} && \text{(15g)}
\end{aligned}$$

where $T_n(t) = x_n(t)T_{n,l}(t) + y_n(t)(e_{n,u}(t)T_{n,u}(t) + e_{n,s}(t)T_{n,s}(t))$ ($\forall n \in \mathcal{N}$); $\mathcal{N}_1$ is the set of edge-processing IoT UEs; F_u^e, F_s^e denote the maximum computation capability in the UAV-assisted edge server and the satellite edge server, respectively; (15a) indicates the offloading decision binary indicator for each IoT UE; (15b) denotes that each IoT UE task is processed either locally by the IoT UE itself or by an edge server; (15c) and (15d) denote that the total computation resources allocated at the UAV-assisted edge server or the satellite edge server must not surpass its maximum computation capability; (15e) denotes that for each IoT UE, the amount of computation resource available at the satellite edge server is the largest, followed by that at the UAV-assisted edge server, while that available for local processing is the smallest but should be non-negative; (15f) indicates the constraint on the transmit power of each IoT UE; and (15g) indicates that each task processing delay consumption should be kept below the maximum tolerable delay threshold.

B. Q-Learning-Based UAV-Assisted Satellite Network Offloading Decision Optimisation Algorithm

To solve the formulated optimisation problem (15), which can be modelled as a Markov decision process (MDP) model [4], and the MDP model is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ and the details of each component are introduced as follows.

- ***State Space*** $\mathcal{S}$: the state space set can be defined as $\mathcal{S} = \{s_t, t \in \mathcal{T}\}$, where s_t is the state containing five parts in each time slot and it can be expressed as

$$s_t \triangleq \{\mathbf{D}(t), \mathbf{C}(t), \tau(t), \mathbf{r}(t), \mathbf{F}^e(t)\}, \tag{16}$$

  where $\mathbf{D}(t) = [D_1(t), ..., D_n(t)(t)]$ and $\mathbf{C}(t) = [C_1(t), ..., C_n(t)(t)]$ are denoted in time slot t the IoT UE input data size and the required total CPU cycles, respectively; $\tau(t) = [\tau_1(t), ..., \tau_N(t)]$ is the delay consumption constraints for each IoT UE task in time slot t; $\mathbf{r}(t) = \{[r_{1,u}(t), ..., r_{N,u}(t)], ..., [r_{1,s}(t), ..., r_{N,s}(t)]\}$ is the set of the achievable uplink transmission rate in the tth time slot between each IoT UE and an edge server; and $\mathbf{F}^e(t) = [F_u(t), F_s(t)]$ denotes that the available computation capacity at a remote edge server.
- ***Action Space*** $\mathcal{A}$: similar to the state space, the action space set can be defined as $\mathcal{A} = \{a_t, t \in \mathcal{T}\}$, which contains two main parts: offloading decisions (i.e., $\mathbf{x}(t), \mathbf{y}(t), \mathbf{e_u}(t), \mathbf{e_s}(t)$) and computation resource allocation (i.e., $\mathbf{f_u^e}(t), \mathbf{f_s^e}(t)$). Therefore, in time slot t, the action $a_t \in \mathcal{A}$ is given by

$$a_t \triangleq \{\mathbf{x}(t), \mathbf{y}(t), \mathbf{e_u}(t), \mathbf{e_s}(t), \mathbf{f_u^e}(t), \mathbf{f_s^e}(t)\}, \tag{17}$$

  where $\mathbf{x}(t) = [x_1(t), ..., x_N(t)]$ indicates that IoT UE n processes the task by itself; $\mathbf{y}(t) = [y_1(t), ..., y_N(t)]$ indicates that IoT UE n processes the task by a remote edge server; $\mathbf{e_u}(t) = [e_{1,u}(t), ..., e_{N,u}(t)]$ indicates that IoT UE n processes the task at the UAV-assisted edge server; $\mathbf{e_s}(t) = [e_{1,s}(t), ..., e_{N,s}(t)]$ indicates that IoT UE n task is processed at the satellite edge server; $\mathbf{f_u^e}(t) = [f_{1,u}^e(t), ..., f_{N,u}^e(t)]$ and $\mathbf{f_s^e}(t) = [f_{1,s}^e(t), ..., f_{N,s}^e(t)]$ denote the allocated computation resources in time slot t at the UAV-assisted or satellite edge server, respectively.
- ***State-Transition Probability*** $\mathcal{P}$: the state-transition probability policy is a mapping operator from state space to action space (i.e., $\pi(a_t|s_t): \mathcal{S} \to \mathcal{A}$).
- ***Reward State-Value Function*** $\mathcal{R}$: in this paper the reward state-value function can be defined as the total amount of reward that an agent in the training network is expected to receive in the long term starting from state s_0, which according to equation (15) can be expressed as

$$\mathcal{R} = \max \mathbb{E}\left[\sum_{t=0}^{T-1} \alpha_t \mathcal{R}'(s_t, a_t)\right], \tag{18}$$

  where $\alpha \in [0, 1]$ is the discount factor; and $\mathcal{R}'(s_t, a_t)$ is the immediate reward function in each time slot, which is given by

$$\begin{aligned}\mathcal{R}'(s_t,a_t) &= -\mathbb{E}\left[T_n(t)\right]\\ &= -\mathbb{E}[x_n(t)T_{n,l}(t)+y_n(t)(e_{n,u}(t)T_{n,u}(t)\\ &\quad + e_{n,s}(t)T_{n,s}(t))].\end{aligned} \tag{19}$$

To solve (19), we propose an actor-critic framework-based optimisation algorithm, which is called the QUSNO algorithm, which is summarised in Algorithm 1. Firstly, the deep neural network environment information and state space are initialised and formulated by the agent accordingly. Then the actor deep neural network selects an action a_t based on the current state s_t, the policy $\pi(s_t|\theta_\pi)$ (which denotes that the neutral network explored offloading decisions and computation resource allocation schemes, and θ_π is the actor-network parameter), and adds Ornstein-Uhlenbeck noise Ω, i.e., $a_t = \pi(s_t|\theta_\pi) + \Omega$. To update the actor neutral network parameter θ_π, we adopt the stochastic gradient descent method as

$$\nabla_{\theta_\pi} J \approx \mathbb{E}\left[\nabla_a Q(s,a|\theta_Q)|_{s=s_t,a=\pi(s_t)} \nabla_{\theta_\pi}\pi(s|\theta_\pi)|_{s=s_t}\right], \tag{20}$$

where θ_Q is the critic network parameter, and $Q(s,a|\theta_Q)$ is the action-value function, which can be obtained based on Bellman Optimality Equation as

$$Q(s,a|\theta_Q) = \mathbb{E}\left[\mathcal{R}'(s_t,a_t) + \gamma Q(s_{t+1},\pi(s_{t+1})|\theta_Q)\right], \tag{21}$$

where the critic network parameter θ_Q can be updated by minimising the loss function, which is expressed as

$$L = \mathbb{E}\left[y_t - Q(s_t,a_t|\theta_Q)\right]^2, \tag{22}$$

where $y_t = \mathcal{R}'(s_t,a_t) + \gamma Q'(s_{t+1},\pi'(s_{t+1}|\theta'_\pi)|\theta'_Q)$. Lastly, the parameters θ_π and θ_Q of target networks can be updated in each iteration as

$$\theta'_\pi = \omega\theta_\pi + (1-\omega)\theta_\pi, \tag{23}$$

$$\theta'_Q = \omega\theta_Q + (1-\omega)\theta_Q, \tag{24}$$

where $\omega \in [0,1]$ is the update coefficient.

Algorithm 1 Q-Learning-Based UAV-Assisted Satellite Network Offloading Decision Optimisation (QUSNO) Algorithm

1: **Initialise** actor-critic network parameters θ_π, θ_Q, learning rate, reward discount factor, system model parameters (i.e., number of episodes, number of time steps, etc.).
2: **Initialise** θ'_Q, $\pi(s_t|\theta_\pi)$, $Q(s_t,a_t|\theta_Q)$, $\pi'(s_t|\theta'_\pi)$, and $Q'(s_t,a_t|\theta_Q)'$ according to θ_π, θ_Q.
3: **for** each episode **do**
4: Initialise the state s_0.
5: **for** each time slot t **do**
6: Select action a_t according to $a_t = \pi(s_t|\theta_\pi) + \Omega$.
7: Calculate the immediate reward value according to (19) and update state s_{t+1}.
8: Randomly sample mini-batch of transition tuples $(s_t, a_t, R'_t, s_{t+1})$, and replace the oldest ones.
9: Update Q value according to $y_t = \mathcal{R}'(s_t,a_t) + \gamma Q'(s_{t+1},\pi'(s_{t+1}|\theta'_\pi)|\theta'_Q)$.
10: Minimise the loss function (22), and update the critic network parameter θ_Q.
11: Update the actor network parameter θ_π according to (20).
12: Update the target networks' parameters θ'_π and θ'_Q according to (23) and (24), respectively.
13: **end for**
14: **end for**
15: **return** $\mathbf{I}^*, \mathbf{f}^{e*}$.

TABLE I: Simulation Parameters [6], [9], [17]

Parameters	**Value**
Number of IoT UEs, N	5
Number of UAV-assisted edge servers	1
Number of satellite edge servers	1
The height of the UAV-assisted edge server, H_{u}	100 m
The height of the satellite edge server, H_{s}	100 km
The satellite system operating frequency, f_{s}	2 GHz
Transmit bandwidth, W	10 MHz
Transmit power of IoT UE n, p_{n}	200 mW
The AWGN power density at the MEC edge node, N_0	-174 dBm/Hz
Data size of a task of IoT-based UE n, $D_n(t)$	$[300, 500]$ kbits
Computation capability of IoT UE n, f_n^l	0.5 G cycles/s
UAV-assisted edge server computation capability, F_{u}^e	1 G cycles/s
Satellite edge server computation capability, F_{s}^e	5 G cycles/s

IV. Simulation Results

In this section, the simulation results with an edge intelligent two-layer UAV-assisted satellite network system as shown in Fig.1 is provided and analysed. The scenario considered is that a satellite edge server offers edge computing service in a $1000m \times 1000m$ area, and a UAV-assisted edge server offers edge computing service in a $200m \times 200m$ area within the satellite edge server coverage [9]. The loss of the LoS link is $\eta_{LoS} = 1dB$ and the loss of the NLoS link is $\eta_{NLoS} = 20dB$ [16]. We assume that each IoT UE only generates one task in each time slot to be processed, which cannot be divided into multiple sub-tasks. Unless specified otherwise, the values for all parameters used in the simulation work are listed in TABLE I [6], [9], [17].

Fig. 2 illustrates the total service delay consumption among all IoT UE devices versus the number of episodes with different discount factors in Algorithm 1. We can see that Algorithm 1 converges after the 330 and 272 episodes for the given discount factors $\{0.7, 0.8\}$, respectively. Moreover, it indicates that a larger discount factor results in better performance. When the discount factor is set to a value close to 1, it indicates that future reward values have a greater impact on cumulative reward relative to the current immediate reward value.

Fig. 3 shows the total service delay consumption versus the number of IoT UEs, where 'QUSNO' is our proposed Algorithm 1, 'Local-Processing', 'UAV Edge-Processing' and

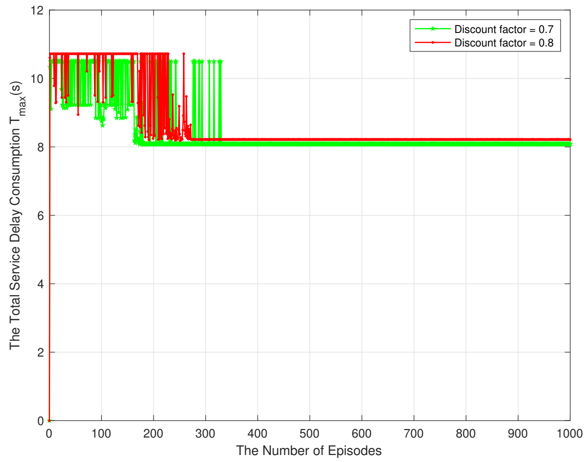

Fig. 2: Convergence of Algorithm 1 ($N = 3$).

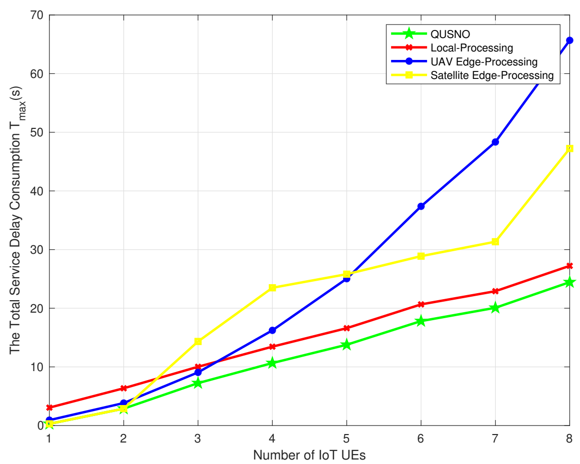

Fig. 3: Total service delay consumption versus the number of IoT UEs.

'Satellite Edge-Processing' are the cases where all tasks will be processed locally, by the UAV-assisted edge server, or by the satellite edge server, respectively. The total service delay consumption increases with the number of IoT UEs among all the considered scenarios, we can see that the QUSNO algorithm illustrates the best for any given number of IoT UEs based on the offloading optimisation in conjunction with the optimised computation resource allocation schemes at the UAV-assisted edge server and the satellite edge server accordingly. When the IoT UE number is larger than 3, satellite edge-processing leads to the highest total service delay due to the transmission delays caused by the long distance between the satellite and IoT UEs. However, When the IoT UE number is larger than 6, UAV-assisted edge-processing leads to the highest total service delay consumption due to processing delays caused by many tasks sharing the limited computation capacity of the UAV-assisted edge server.

V. Conclusion

In this paper, we have proposed a Q-learning-based UAV-assisted satellite network offloading decision optimisation algorithm in conjunction with computation resource allocation optimisation to minimise the total service delay consumption of all IoT UEs in a UAV-assisted satellite network system, where each IoT UE may process its task locally or offload it to a remote edge computing server. The simulation results demonstrate that the proposed algorithm achieves a much lower total service delay consumption than local-processing, UAV Edge-processing, and satellite Edge-processing.

References

[1] J. Du, J. Lv, and G. Lu, "Economical revenue maximization in mobile edge caching and blockchain enabled space-air-ground integrated networks," *Journal of Cloud Computing*, vol. 12, no. 1, p. 98, 2023.

[2] M. Ghobakhloo, "Industry 4.0, digitization, and opportunities for sustainability," *Journal of cleaner production*, vol. 252, p. 119869, 2020.

[3] R. Huo, S. Zeng, Z. Wang, J. Shang, W. Chen, T. Huang, S. Wang, F. R. Yu, and Y. Liu, "A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges," *IEEE Comm. Sur. & Tutor.*, vol. 24, no. 1, pp. 88–122, 2022.

[4] J. Du, W. Cheng, G. Lu, H. Cao, X. Chu, Z. Zhang, and J. Wang, "Resource pricing and allocation in mec enabled blockchain systems: An a3c deep reinforcement learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 33–44, 2021.

[5] Y. Yang, M. Ma, H. Wu, Q. Yu, P. Zhang, X. You, J. Wu, C. Peng, T.-S. P. Yum, S. Shen *et al.*, "6g network ai architecture for everyone-centric customized services," *arXiv preprint arXiv:2205.09944*, 2022.

[6] J. Du, Z. Kong, A. Sun, J. Kang, D. Niyato, X. Chu, and F. R. Yu, "Maddpg-based joint service placement and task offloading in mec empowered air-ground integrated networks," *IEEE IoT J.*, pp. 1–1, 2023.

[7] X. Zhu and C. Jiang, "Integrated satellite-terrestrial networks toward 6g: Architectures, applications, and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 437–461, 2021.

[8] T. Kim, J. Kwak, and J. P. Choi, "Satellite edge computing architecture and network slice scheduling for iot support," *IEEE Internet of Things journal*, vol. 9, no. 16, pp. 14 938–14 951, 2021.

[9] Z. Song, Y. Hao, Y. Liu, and X. Sun, "Energy-efficient multiaccess edge computing for terrestrial-satellite internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14 202–14 218, 2021.

[10] F. Wang, D. Jiang, S. Qi, C. Qiao, and L. Shi, "A dynamic resource scheduling scheme in edge computing satellite networks," *Mobile Networks and Applications*, vol. 26, pp. 597–608, 2021.

[11] I. Leyva-Mayorga, M. Martinez-Gost, M. Moretti, A. Peŕez-Neira, M. Á. Vázquez, P. Popovski, and B. Soret, "Satellite edge computing for real-time and very-high resolution earth observation," *IEEE Trans. on Comm.*, 2023.

[12] W. Lu, Y. Mo, Y. Feng, Y. Gao, N. Zhao, Y. Wu, and A. Nallanathan, "Secure transmission for multi-uav-assisted mobile edge computing based on reinforcement learning," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 3, pp. 1270–1282, 2022.

[13] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, and D. Niyato, "Multi-agent deep reinforcement learning for task offloading in uav-assisted mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 6949–6960, 2022.

[14] L. Zhang, B. Jabbari, and N. Ansari, "Deep reinforcement learning driven uav-assisted edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 449–25 459, 2022.

[15] W. Zhou, L. Fan, F. Zhou, F. Li, X. Lei, W. Xu, and A. Nallanathan, "Priority-aware resource scheduling for uav-mounted mobile edge computing networks," *IEEE Transactions on Vehicular Technology*, 2023.

[16] H. Ren, C. Pan, K. Wang, Y. Deng, M. Elkashlan, and A. Nallanathan, "Achievable data rate for urllc-enabled uav systems with 3-d channel model," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1587–1590, 2019.

[17] B. Hu, J. Du, and X. Chu, "Enabling low-latency applications in vehicular networks based on mixed fog/cloud computing systems," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 722–727.