

RESEARCH

Open Access



# Collaborative task offloading and resource allocation with hybrid energy supply for UAV-assisted multi-clouds

Yu Zhou, Hui Ge, Bowen Ma, Shuhang Zhang and Jiwei Huang\*

## Abstract

Cloud computing has emerged as a promising paradigm for meeting the growing resource demands of Internet of Things (IoT) devices. Meanwhile, with the popularity of mobile aerial base stations, Unmanned Aerial Vehicle (UAV) assisted cloud computing is essential for providing diversified service at areas without available infrastructure. However, it is difficult to meet the requirements of a number of IoT devices which distribute a large area through one single UAV cloud server, and thus multi-clouds have been applied in large-scale IoT environments. Due to the limited battery capacity of UAV, hybrid energy supply has been considered as an effective approach. How to allocate the computing resources and offload the tasks to the UAV-assisted clouds is a challenging task. In this paper, we study the trade-off between the energy consumption and system performance in a UAV-assisted multi-clouds system. Considering the transmission and execution cost, a dynamic optimization problem with the objective of minimizing the power consumption of UAVs with the constraint of queue stability is formulated, which is further decomposed into three sub-problems using stochastic optimization techniques. A collaborative task offloading and resources allocation algorithm (CTORAA) based on artificial intelligent (AI) technique is proposed to jointly determine task offloading and energy harvesting. We provide corresponding mathematical analysis showing that CTORAA can reach the arbitrary profit-stability trade-off. Finally, we conduct simulation experiments to validate the efficacy of our algorithm.

**Keywords:** Multi-clouds, Unmanned Aerial Vehicle (UAV), Internet of Things (IoT), Lyapunov optimization, Hybrid energy supply, Task offloading, Resource allocation

## Introduction

Internet of Things (IoT) is an emerging technology that connects everything in the Internet. It has been widely applied in various domains such as healthcare, transportation, industry, etc. [1]. Many applications which require heavy computation and high energy consumption have been emerging, such as virtual reality, face recognition, and service recommendation [2]. However, due to the limited resources of IoT devices [3], such as limited battery and weak computing capability, computation-intensive and resource-consuming tasks cannot be fully

performed on IoT devices [4, 5]. Combining the advantage of mobile computing and cloud computing, Mobile Cloud Computing (MCC) has been proposed to overcome the limited processing power and poor storage capacity of mobile devices [6]. In MCC, tasks generated by IoT devices can be transferred to multiple cloud servers, and thus the Quality of Service (QoS) can be guaranteed [7]. For edge-based IoT services, Zhang et al. [8] proposed a distributed edge QoS prediction model with privacy-preserving

In some extreme conditions (such as earthquake, flood, and typhoon), MCC deployed on the base station or centralized data centers may not work due to the failure of communication and computing infrastructures. To attack this challenge, Unmanned Aerial Vehicle (UAV) has been

\*Correspondence: huangjw@cup.edu.cn

Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum, Beijing, China

introduced to carry computing devices acting as a small cloud server as well as providing communication support [9]. Such promising technology, namely UAV-assisted cloud computing, has attracted extensive attention in both academia and industry.

Due to the limited battery capacity of the UAV, it is difficult to fly a UAV and let it provide services for IoT devices for a long time. Therefore, in order to prolong the service lifecycle, UAV with energy harvesting has been widely studied. However, energy is still the most limited resource, since both UAV itself and its payload (cloud server) consume energy while providing services. How to allocate the computing resources and schedule the tasks among the devices and multi-clouds with maximized energy utilization adapting to the dynamic environment and task arrivals is a challenging task.

In this paper, we study the problem of collaborative task offloading and resource allocation with hybrid energy supply for UAV-assisted multi-clouds. Our goal is to minimize the average energy consumption of the system meanwhile ensuring the performance. In order to deal with the dynamics of task arrival, we formulate a stochastic optimization problem, and further decompose the original problem into three independent sub-problems. Mathematical optimization methodology as well as AI-empowered approaches are presented to solve the sub-problems. We design a Collaborative Task Offloading and Resource Allocation Algorithm (CTORAA), and provide quantitative analysis to prove the effectiveness of our approach. Finally, we conduct simulation experiments to validate the performance of our CTORAA algorithm by comparing with other benchmark algorithms.

The rest of this paper is organized as follows. In Section 2, we survey the related works. In Section 3, we introduce the basic scenario, and present the system model and problem formulation. In Section 4, a collaborative task offloading and resource allocation algorithm is designed, and its mathematical analysis is provided. In Section 5, we conduct simulation experiments to validate our approach. Finally, in Section 6, we conclude this paper.

## Related Work

It is practical to support IoT applications by transferring heavy computation tasks to powerful servers in the Edge-Cloud system. In [10], Almutairi and Aldossary introduced an Edge-Cloud system architecture that includes the required components to support scheduling offloading tasks of IoT applications. They provided a set of algorithms for offloading tasks scheduling in order to enhance service time and resource utilization in Edge-Cloud environments. Chen et al. [11] integrated the resources of Mobile Edge Computing (MEC) and cloud to study the

collaborative and dynamic task offloading for IoT devices. They set one of the task offloading goals as maximizing the task finishing ratio with tolerable delay, and the other offloading goal is minimizing the power consumption. The two offloading goals are conflicting. The authors formulated the detailed task offloading problem to balance the two conflicting goals subject to resources constraints. In [12], Rong et al. presented the computing model and the architecture of Sophon Edge, an edge-cloud collaborative Artificial Intelligence of Things (AIoT) platform. While the cloud exploits its sufficient computing and storage resources to train sophisticated machine learning models, frequent but less complicated computation tasks are offloaded to the edge of the IoT network to provide low latency services, allowing the end users of the AIoT application to get near real-time responses, which really improves the user experience.

There is a growing interest in the study of UAV-assisted cloud computing system. Gao et al. [13] proposed a potential game combined multi-agent deep deterministic policy gradient (MADDPG) approach to optimize multiple UAVs' trajectory with the consideration of ground users' offloading delay, energy efficiency as well as obstacle avoidance system. In [14], with multi-UAV and ground mobile UEs in the system, heterogeneous performance requirement as well as fast-changing communication condition make the system more complicated. Wang et al. proposed a heuristic joint power and quality of experience (HJPQ) algorithm where the user equipments' (GEs') offloading delay, MIMO channel, transmission power, as well as UAVs' placement are jointly optimized. Various aerial offloading data with different QoE requirements is stored and relayed in the multi-queueing architecture. Hence offloading rate differentiation is utilized to ensure the high-priority data a better QoE. In [15], Gao et al. proposed an energy-efficient resource allocation scheme with the ability of QoE enhancement. In [16], on the premise of ensuring that all user tasks are completed, the time-division multiple access protocols were adopted to minimize the energy consumption of all users. And the energy-performance tradeoff problem has always been a hot research topic in various aspects of wireless communication networks and IoT technology, as well as other network technologies.

The energy consumption of UAVs is also an important aspect to consider in the design of UAV systems. In order to extend the life of UAVs, energy must be used efficiently [17]. In recent years, there were many people have made efforts to improve the energy efficiency of UAVs. Du et al. [18] investigated a UAV-enabled mobile computing system in which the UAV used wireless power transfer (WPT) technology to power IoT devices. Then the data collected by each IoT device is transmitted to the

UAV for processing. To improve the energy efficiency of the UAV, a new Time Division Multiple Access (TDMA) based workflow model was proposed that allows parallel transmission and execution in the UAV-enabled system. The total energy consumption of UAV is minimized by jointly optimizing the flight zone association, computation resource allocation, UAV hovering time, wireless power supply time, and flight zone order of service. In [19], they studied the energy-performance tradeoff for data transmission in UAV-enabled MEC, and they proposed a method that provides an effective scheme for realizing the energy-performance tradeoff in system design. In [20], the author investigated a multi-UAV collaborative data acquisition system, multiple UAVs perform data collection on the two-dimensional distributed devices in a flying or hovering mode to save energy.

Zhan et al. [21] concentrated on the completion time and energy optimization in UAV-enabled mobile computing system. To maximize the remaining energy, in [22], they combined the energy production model and energy consumption model to optimize the level flight trajectory. In [23], the authors investigated multi-UAV

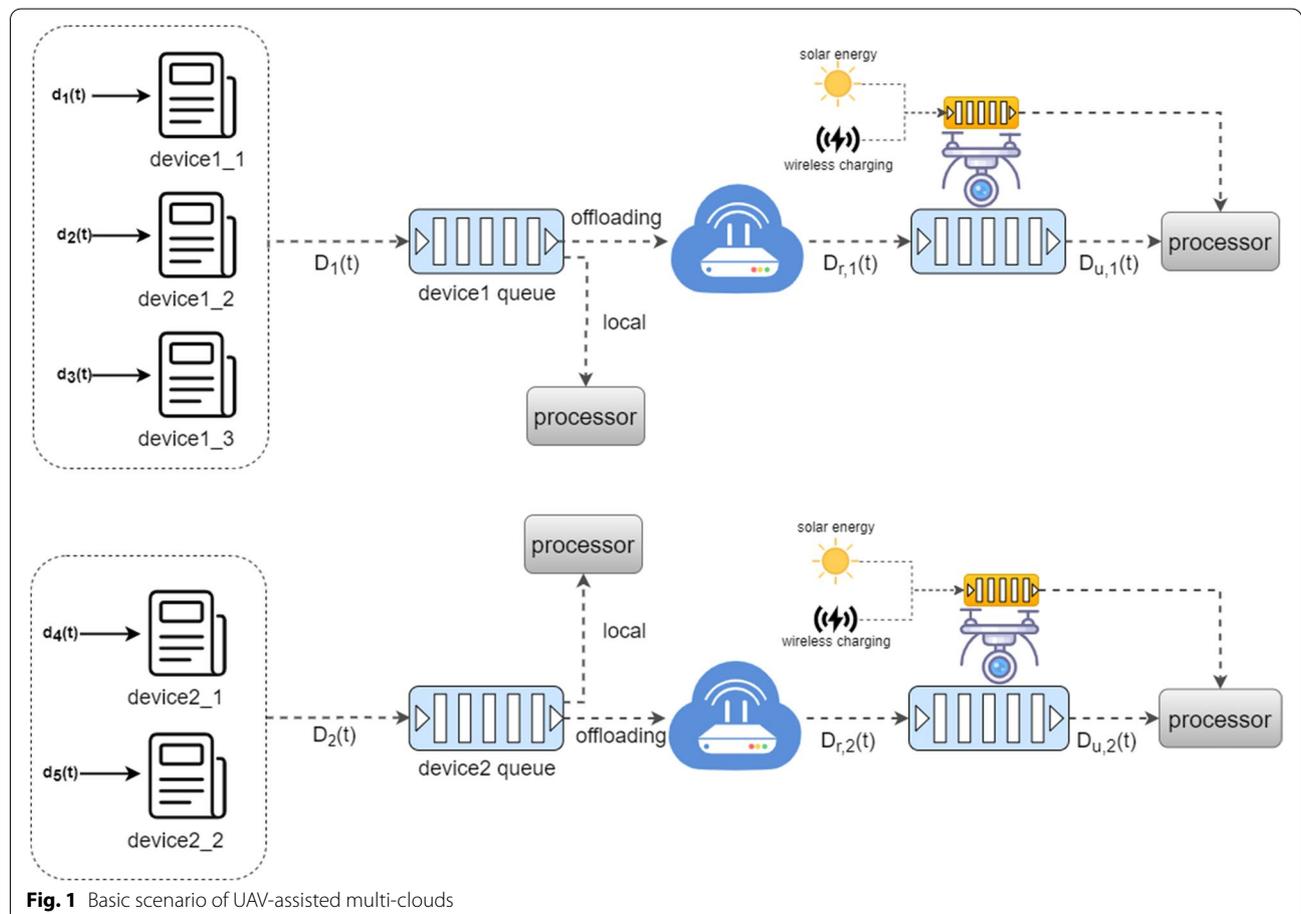
providing edge computing services by collaboration. They developed a multi-UAV cooperative edge computing framework. The tasks offloaded from IoT devices to UAVs are processed cooperatively by UAVs. To solve this problem, they jointly optimized the number of deployed UAVs and the offloading decision for each IoT device. Then, they determined the optimization objective of minimizing system energy consumption while ensuring that all offloading tasks were completed. And an efficient collaborative greedy task scheduling algorithm was designed to achieve the optimization objective.

However, little attention has been paid to the relationship between energy consumption and system performance in a UAV-assisted multi-clouds computing system. So we concentrate on the tradeoff between the energy consumption generated by transmission and computing and the system stability.

### System Model and Problem Formulation

#### System Framework for UAV-assisted multi-clouds

Figure 1 demonstrates a basic scenario of a UAV-assisted multi-clouds system. We consider the system consisting of



$X$  energy-limited IoT devices denoted as  $A = \{1, 2, \dots, X\}$ , and  $Y$  UAVs denoted as  $B = \{1, 2, \dots, Y\}$ . We assume that all UAVs fly at a fixed height and all IoT devices randomly distributed in a flat ground. In this model, the ground equipment is divided into  $Y$  disjoint partitions, and each partition is associated with a UAV. The task requests from each IoT device can be transmitted to the UAV belonging to its region by the gateway.

In order to express the computing task offload strategy more clearly, we introduce  $A_j$  to denote the set of IoT devices covered by UAV  $j$ . It yields the following constraints:

$$A_1 \cup A_2 \cup \dots \cup A_j \cup \dots \cup A_Y = A$$

$$A_j \cap A_k = \emptyset, \forall j, k \in B \& j \neq k \tag{1}$$

In the above-mentioned UAV-assisted multi-clouds computing system, the computing task of each IoT device  $i (i \in A_j, j \in B)$  can have two offloading strategies: local computing and cloud server computing. First, local computing means that the computing task is executed at IoT device  $i$ . Second, cloud server computing means that IoT device  $i$  offloading the computing tasks to UAV  $j$ . We consider a discrete time-slotted system  $T = \{1, 2, 3, \dots\}$ . And the slot length is denoted by  $\tau$ . In Table 1, the main notations in the system model are provided.

**Task Model for IoT devices**

In every time slot  $t \in T$ , the number of tasks that arrive at the  $i$ th IoT device can be expressed as  $d_i(t), \forall i \in A, 0 \leq d_i(t) \leq d_i^{max}$ . We only focus on computationally intensive tasks. The processing complexity of computationally intensive tasks can be considered to be proportional to the size of the input task. Then, the total number of tasks that arrive at the IoT devices set  $A_j$  is calculated as

$$D_j(t) = \sum_{i \in A_j} d_i(t), j \in B. \tag{2}$$

Within the  $t$ th time slot, IoT devices set  $A_j$  will locally execute  $D_{l,j}(t)$  tasks and will offload  $D_{r,j}(t)$ (the requests that the UAV allowed) tasks to the UAV  $j$ . At the beginning of the  $t$ th time slot, the local queue length of the  $j$ th IoT device set  $A_j$  is  $S_j(t)$ . Initially,  $S_j(0) = 0$ . At  $t$ th time slot, the amount of tasks which has arrived but not been executed locally or offloaded will be put in  $S_j(t)$ . Then, we can capture the following queue dynamics over time:

$$S_j(t + 1) = \max\{S_j(t) - D_{l,j}(t) - D_{r,j}(t), 0\} + D_j(t), j \in B. \tag{3}$$

Then we set up a cache queue  $M_j(t)$  to indicate the number of task requests of the UAV  $j$  that have not been

**Table 1** Notations and definitions

Notion	Definition
$d_i(t)$	The number of arrived requests of IoT device $i$ in time slot $t$
$D_j(t)$	The total number of arrived requests of IoT devices $A_j$ in time slot $t$
$D_{l,j}(t)$	The number of tasks locally executed by IoT devices set $A_j$ in time slot $t$
$D_{r,j}(t)$	The number of tasks offloaded by IoT devices set $A_j$ in time slot $t$
$D_{u,j}(t)$	The number of tasks that executed the UAV $j$ in time slot $t$
$S_j(t)$	Queue backlog of IoT devices set $A_j$ in time slot $t$
$M_j(t)$	Queue backlog of UAV $j$ in time slot $t$
$H_j(t)$	The remaining battery capacity of UAV $j$ in time slot $t$
$W_j(t)$	The size of obtained energy of UAV $j$ in time slot $t$
$E_H$	The capacity of battery of UAV
$E_{safe}$	the minimum battery energy level
$E_{u,j}$	The energy consumption of UAV $j$ in time slot $t$
$f_{i,j}(t)$	The CPU-cycle frequency of the IoT device $i$ in time slot $t$
$f_{max,i}$	The maximum CPU-cycle frequency of the IoT device $i$
$L_i$	The CPU cycles of IoT device $i$ when executing 1 bit task in time slot $t$
$p_{l,i}(t)$	The CPU power consumption of IoT device $i$ in time slot $t$
$k_{mod,i}$	The effective switched capacitance of the CPU of IoT device $i$
$r_{i,j}(t)$	The transmission rate between the IoT device $i$ and the UAV $j$ in time slot $t$
$p_{r,j}(t)$	The transmission power of the IoT device $i$ in time slot $t$
$p_{max,i}$	The maximum transmission power of IoT device $i$
$B_i$	The channel bandwidth of IoT device $i$
$G_i$	The channel power gain at a reference distance 1m of IoT device $i$
$d_{i,j}(t)$	The distance between IoT device $i$ and the UAV $j$ in time slot $t$

processed by the cloud server. Initially, we assume that  $M_j(0) = 0$ . The total number of tasks that have been executed by the cloud server can be expressed as  $D_{u,j}(t)$ . Thus, the queue length of each UAV evolves as the following equality

$$M_j(t + 1) = \max\{M_j(t) - D_{u,j}(t), 0\} + D_{r,j}(t), j \in B. \tag{4}$$

### Hybrid Energy Model for UAVs

For each UAV, it is equipped with a battery with limited capacity  $E_H$ . Denote the remaining battery capacity of UAV  $j$  at the beginning of the  $t$ th time slot as  $H_j(t)$ . And we assume that  $H_j(0) = \frac{1}{2}E_H$ . To ensure UAVs can continue to serve IoT devices, we provide UAVs with a hybrid energy supply [24]. We assume that energy source of UAVs includes two parts, which are the harvested solar energy and grid energy [25]. When the solar energy would be unable to afford the operations of the UAVs, they will obtain energy by wireless charging. At the beginning of the  $t$ th time slot, we assume that successive energy with size  $W_j(t)$  arrives in UAV  $j$ . Thus, battery charging has the following constraint:

$$H_j(t) + W_j(t) \leq E_H, j \in B. \quad (5)$$

We consider that energy consumption of UAVs consists of computing energy consumption. Then, let  $E_{u,j}(t)$  represents the energy consumption incurred by UAV  $j$  within the  $t$ th time slot. The battery life is affected by excessive discharging. Therefore, we introduce safe threshold as  $E_{safe}$ , i.e., the minimum battery energy level. Thus, we have

$$M_j(t) + W_j(t) - E_{u,j}(t) \geq E_{safe}, j \in B. \quad (6)$$

The evolution equation of energy for the  $j$ th UAV is

$$H_j(t + 1) = \max\{H_j(t) - E_{u,j}(t), 0\} + W_j(t), j \in B. \quad (7)$$

### Task Offloading and Resource Allocation Models for IoT devices

Within the  $t$ th time slot, the total number of tasks computed locally by the IoT devices set  $A_j$  can be expressed as

$$D_{l,j}(t) = \sum_{i \in A_j} \tau f_{l,i}(t) L_i^{-1}, j \in B, \quad (8)$$

$$0 \leq f_{l,i}(t) \leq f_{max,i}, i \in A, \quad (9)$$

where  $f_{l,i}(t)$  is the CPU-cycle frequency of the  $i$ th IoT device, and  $f_{max,i}$  is the maximum value.  $L_i$  (cycles/bit) denotes the CPU cycles of the  $i$ th IoT device when executing 1 bit task.

According to circuit theories, the CPU power consumption of the  $i$ th IoT device in the  $t$ th time slot is calculate as

$$p_{l,i}(t) = k_{mod,i} f_{l,i}^3(t), i \in A, \quad (10)$$

where  $k_{mod,i}$  is the effective switched capacitance of the CPU of the  $i$ th IoT device.

In the UAV-assisted multi-clouds computing system, the  $i$ th ( $i \in A_j$ ) IoT device can send computing task to the  $j$ th UAV by a gateway. We denote the transmission power of the  $i$ th UAV in the  $t$ th time slot is  $p_{r,i}(t)$ . Thus, the transmission rate between the  $i$ th IoT device and the  $j$ th UAV is expressed as

$$r_{i,j}(t) = B_i \log_2 \left( 1 + \frac{p_{r,i}(t) G_i Z_0}{B_i N_0 d_{i,j}^2} \right), j \in B, i \in A_j, \quad (11)$$

$$0 \leq p_{r,i}(t) \leq p_{max,i}, i \in A, \quad (12)$$

where  $B_i$  is the channel bandwidth of IoT device  $i$ , the  $G_i$  is the channel power gain at a reference distance 1m of IoT device  $i$ .  $d_{i,j}$  is denoted as the distance between the  $i$ th IoT device and the  $j$ th UAV.  $N_0$  is the noise power spectrum density.  $Z_0$  is a positive constant ( $\approx 2.2846$ ).  $p_{max,i}$  denotes the maximum transmission power of the  $i$ th IoT device.

Within the  $t$ th time slot, the total number of tasks transmitted through IoT device set  $A_j$  to UAV  $j$  can be expressed as

$$D_{r,j}(t) = \sum_{i \in A_j} r_{i,j}(t) \tau, j \in B. \quad (13)$$

The transmission energy consumption of the  $i$ th IoT device in time slot  $t$  is calculate as

$$E_{r,j}(t) = \sum_{i \in A_j} p_{r,i}(t) \tau, j \in B \quad (14)$$

In the  $t$ th time slot, the amount of tasks of the  $j$ th UAV is denoted as  $D_{u,j}(t)$ . Then,  $D_{u,j}(t)$  is calculated as

$$D_{l,j}(t) = \tau f_{u,j}(t) L_j^{-1}, j \in B, \quad (15)$$

$$0 \leq f_{u,j}(t) \leq f_{max,j}, j \in B, \quad (16)$$

where  $L_j$ (cycles/bit) is the CPU cycles which the  $j$ th UAV uses to execute 1 bit task.  $f_{u,j}(t)$  is denoted as the CPU-cycles frequency of the  $j$ th UAV, and  $f_{max,j}$  is the maximum value.

The energy consumption of the computing of the  $j$ th UAV in time slot can be express as

$$E_{u,j}(t) = k_{mod,j} f_{u,j}^3 \tau, j \in B, \quad (17)$$

where  $k_{mod,j}$  denotes the effective switched capacitance of the CPU of the  $j$ th UAV.

### Task Offloading Problem Formulation

In UAV-assisted multi-clouds computing system, system stability is also an important factor to evaluate system performance. According to Little’s law [26], the average queue delay is proportional to the queue length. Furthermore, the queue length can be used as the metric to evaluate the performance of the system [27, 28]. We want all queues not to be congested. In other words, we require all the actual queues to be stable in the time average sense. Inspired by [29], we can obtain:

$$\bar{S} = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{ \sum_{j \in B} |S_j(t)| \right\} < \infty, \quad (18)$$

$$\bar{M} = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{ \sum_{j \in B} |M_j(t)| \right\} < \infty, \quad (19)$$

$$\bar{H} = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{ \sum_{j \in B} |H_j(t)| \right\} < \infty, \quad (20)$$

where  $\bar{S}$ ,  $\bar{M}$  and  $\bar{H}$  denote the time-averaged queue backlog. We hope that the value of the queue backlog is as small as possible.

For computationally intensive tasks, the energy computation of sending the results back is much smaller than the transmission and execution. Besides, we assume all UAVs consume the same energy for flight in each time slot. In the above-mentioned UAV-assisted multi-clouds computing system, we only consider the energy consumption for transmission and computing. First, we denote the system operation at the  $t$ th time slot with  $X(t) \triangleq \{f_{l,i}(t), p_{r,i}(t), f_{u,j}(t), W_j(t)\}$ . Then the objective function that minimize the average energy consumption of the system is defined as P1:

$$P1 : \min_{\{X(t)\}} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[ \sum_{j \in B} (E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)) \right], \quad (21)$$

s.t. (5), (6), (9), (12), (16), (18), (19), (20).

Since the optimization goal and constraints depend on not only the current system states but also the future ones, the formulated problem P1 is a stochastic optimization problem. Based on Lyapunov optimization, the stochastic optimization problem is converted into a deterministic optimization problem. In next section, we will introduce an online algorithm for solving the above problem.

### Collaborative Task Offloading and Resource Allocation Algorithm Design

In this section, we take advantage of the Lyapunov optimization techniques combining with an intelligent approach to design an online control algorithm. By leveraging the stochastic optimization techniques, the original problem (21) is decomposed into three independent sub-problems, corresponding to the three decisions that need to be made: 1) local computing control; 2) task offloading control; 3) cloud computing and charging control. Then mathematical analysis shows that the algorithm can reach the arbitrary optimal solution on the premise of maintaining the system stability.

#### Drift-plus-penalty Design

According to [30], first we define the vector of queue backlogs at time  $t$  by:

$$\begin{aligned} S(t) &= (S_1(t), S_2(t), \dots, S_Y(t)), \\ M(t) &= (M_1(t), M_2(t), \dots, M_Y(t)), \\ H(t) &= (H_1(t), H_2(t), \dots, H_Y(t)). \end{aligned} \quad (22)$$

Let  $\Theta(t) = [S(t), M(t), H(t)]$  denotes the combined matrix of all the queues.

To model the problem as a Lyapunov optimization problem, we define the Lyapunov function  $L(\Theta(t))$  as the sum of the squares of the current queue backlogs for each slot.

$$L(\Theta(t)) = \frac{1}{2} \sum_{j \in B} [S_j^2(t) + M_j^2(t) + H_j^2(t)]. \quad (23)$$

This function is a scalar measure of the total queue backlog in the network. It is called *quadratic Lyapunov function* on the queue state. Next we introduce the Lyapunov drift  $\Delta(\Theta(t))$  as the change from one slot to the next:

$$\Delta(\Theta(t)) = \mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t)) | \Theta(t)\}. \quad (24)$$

The smaller  $\Delta(\Theta(t))$ , the more stable the queues. In order to make the network stable while minimizing the time average of the penalty  $E_{j,t} + E_{j,v}$ , the algorithm can be designed to minimize the upper bound of the drift-plus-penalty function in each time slot  $t$ . Then, we denote the Lyapunov drift-plus-penalty function as

$$\Delta(\Theta(t)) + V \mathbb{E}\left[ \sum_{j \in B} (E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)) | \Theta(t) \right], \quad (25)$$

where  $V$  is a non-negative parameter that is used to as adjusting the tradeoff between queue length and energy consumption [31, 32].

**Deriving Upper Bound**

Then a key derivation step is to obtain an upper of the drift-plus-penalty expression (25). The following lemma defines such an upper bound.

**Lemma 1** *In each time slot  $t$ , for any value of  $\Theta(t)$  and any parameter value of  $V$ , we have:*

$$\begin{aligned} \Delta(\Theta(t)) + V\mathbb{E}[\sum_{j \in B} (E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)) | \Theta(t)] &\leq B \\ + \mathbb{E}[\sum_{j \in B} S_j(t)(D_j(t) - D_{l,j}(t) - D_{r,j}(t)) | \Theta(t)] \\ + \mathbb{E}[\sum_{j \in B} M_j(t)(D_{r,j}(t) - D_{u,j}(t)) | \Theta(t)] \\ + \mathbb{E}[\sum_{j \in B} H_j(t)(W_j(t) - E_{u,j}(t))] \\ + V\mathbb{E}[\sum_{j \in B} (E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)) | \Theta(t)], \end{aligned} \tag{26}$$

where  $B$  is a constant.

*Proof* First, we define

$$\Delta Q_j(t) = S_j^2(t+1) + M_j^2(t+1) + H_j^2(t+1) - S_j^2(t) - M_j^2(t) - H_j^2(t). \tag{27}$$

From (3), (4) and (7), we have

$$\begin{aligned} \Delta Q_j(t) &= \{ \max \{ S_j(t) - D_{l,j}(t) - D_{r,j}(t), 0 \} + D_j(t) \}^2 \\ &+ \{ \max \{ M_j(t) - D_{u,j}(t), 0 \} + D_{r,j}(t) \}^2 \\ &+ \{ \max \{ H_j(t) - E_{u,j}(t), 0 \} + W_j(t) \}^2 \\ &- S_j^2(t) - M_j^2(t) - H_j^2(t). \end{aligned} \tag{28}$$

By applying the inequality that  $(\max\{d, 0\})^2 \leq d^2$ , it can be obtained that

$$\begin{aligned} \Delta Q_j(t) &\leq [S_j(t) - D_{l,j}(t) - D_{r,j}(t)]^2 + D_j^2(t) + 2S_j(t)D_j(t) \\ &+ [M_j(t) - D_{u,j}(t)]^2 + D_{r,j}^2(t) + 2M_j(t)D_{r,j}(t) \\ &+ [H_j(t) - E_{u,j}(t)]^2 + W_j^2(t) + 2H_j(t)W_j(t) \\ &- S_j^2(t) - M_j^2(t) - H_j^2(t). \end{aligned} \tag{29}$$

After calculating, we can rewrite (29) as follow

$$\begin{aligned} \Delta Q_j(t) &\leq D_{l,j}^2(t) + D_{r,j}^2(t) + D_j^2(t) + D_{r,j}^2(t) + D_{u,j}^2(t) \\ &+ E_{u,j}^2(t) + W_j^2(t) + 2S_j(t)(D_j(t) - D_{l,j}(t) - D_{r,j}(t)) \\ &+ 2M_j(t)(D_{r,j}(t) - D_{u,j}(t)) + 2H_j(t)(W_j(t) - E_{u,j}(t)). \end{aligned} \tag{30}$$

According to the previous definition, we can learn that  $D_{l,j}(t)$ ,  $D_{r,j}(t)$ ,  $D_j(t)$ ,  $D_{u,j}(t)$ ,  $E_{u,j}(t)$  and  $W_j(t)$  all have the upper bound, i.e.,  $D_{l,j}(t) \leq D_{l,j}^{max}(t)$ ,  $D_{r,j}(t) \leq D_{r,j}^{max}(t)$ ,  $D_j(t) \leq D_j^{max}(t)$ ,  $D_{u,j}(t) \leq D_{u,j}^{max}(t)$ ,  $E_{u,j}(t) \leq E_{u,j}^{max}(t)$  and  $W_j(t) \leq W_j^{max}(t)$ . Let  $B$  take the value of

$$\begin{aligned} B &= \frac{1}{2} \sum_{j \in B} \left[ \left( D_{l,j}^{max}(t) \right)^2 + 2 \left( D_{r,j}^{max}(t) \right)^2 + \left( D_j^{max}(t) \right)^2 \right. \\ &\left. + \left( D_{u,j}^{max}(t) \right)^2 + \left( W_j^{max}(t) \right)^2 + \left( E_{u,j}^{max}(t) \right)^2 \right]. \end{aligned} \tag{31}$$

Then, we can obtain that

$$\begin{aligned} \Delta(\Theta(t)) &= \frac{1}{2} E \left( \sum_{j \in B} \Delta Q_j(t) \mid \Theta(t) \right) \\ &\leq B + E \left[ \sum_{j \in B} S_j(t)(D_j(t) - D_{l,j}(t) - D_{r,j}(t)) \mid \Theta(t) \right] \\ &+ E \left[ \sum_{j \in B} M_j(t)(D_{r,j}(t) - D_{u,j}(t)) \mid \Theta(t) \right] \\ &+ E \left[ \sum_{j \in B} H_j(t)(W_j(t) - E_{u,j}(t)) \mid \Theta(t) \right]. \end{aligned} \tag{32}$$

The Lyapunov drift-plus-penalty function is defined as:

$$\begin{aligned} \Delta_v(\Theta(t)) &= \Delta(\Theta(t)) + VE[E_{f,j}(t) + E_{t,j}(t) \mid \Theta(t)] \\ &\leq B + E \left[ \sum_{j \in B} S_j(t)(D_j(t) - D_{l,j}(t) - D_{r,j}(t)) \mid \Theta(t) \right] \\ &+ E \left[ \sum_{j \in B} M_j(t)(D_{r,j}(t) - D_{u,j}(t)) \mid \Theta(t) \right] \\ &+ E \left[ \sum_{j \in B} H_j(t)(W_j(t) - E_{u,j}(t)) \mid \Theta(t) \right] \\ &+ VE[E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t) \mid \Theta(t)]. \end{aligned} \tag{33}$$

Notably, in time slot  $t$ ,  $B$  and  $S_j(t)D_j(t)$  are constants. To be more specific, parameter  $B$  is the constant defined in (31). Besides,  $S_j(t)D_j(t)$  can be obtained and known at the beginning of each time slot  $t$ . Thus, the supremum bound minimization problem can be equivalently reformulated by eliminating the constant term.

$$\begin{aligned}
 P2 : \min_{\{X(t)\}} & \sum_{j \in B} [-S_j(t)D_{l,j}(t) + VE_{l,j}(t)] \\
 & + \sum_{j \in B} [D_{r,j}(t)(M_j(t) - S_j(t)) + VE_{r,j}(t)] \\
 & + \sum_{j \in B} [H_j(t)(W_j(t) - E_{u,j}(t)) - M_j(t)D_{u,j}(t) + VE_{u,j}(t)] \\
 \text{s.t.} & \quad (5), (6), (9), (12), (16).
 \end{aligned} \tag{34}$$

**Collaborative Task Offloading and Resource Allocation Algorithm**

Instead of solving the  $P2$  optimally, we decompose it into three independent sub-problems, corresponding to the three decisions that need to be made. Next, we derive the optimal solutions to these three sub-problems separately.

Specifically, our algorithm is described in Algorithm 1.

**Algorithm 1** : Collaborative Task Offloading and Resource Allocation Algorithm (CTORAA)

- 1: Initialization: At the beginning of  $t$ th, obtain random event  $\{d_i(t)\}$ , monitor the queue backlog  $S_j(t), M_j(t)$  and  $H_j(t)$  of each UAV  $j$ .
- 2:  $\forall i \in A$ , get optimal  $f_{i,j}^*(t)$  and  $p_{r,i}^*(t)$  by solving  $SP1$  and  $SP2$ , respectively.
- 3:  $\forall j \in B$ , solve  $SP3$  and obtain the optimal  $f_{u,j}^*(t)$  and  $W_j^*(t)$ .
- 4: According to the iteration formula in (3), (4) and (7), update queues  $S_j(t), M_j(t)$  and  $H_j(t)$  when the current time slot  $t$  ends.
- 5: Go to the next time slot, with  $t = t + 1$ .

**Local Computing Control**

We determine  $f_{l,i}(t)$  by solving the problem  $SP1$  as follows

$$\begin{aligned}
 SP1 : \min_{\{f_{l,i}(t)\}} & \sum_{j \in B} \left[ -S_j(t) \sum_{i \in A_j} f_{l,i}(t)L_i^{-1} + V \sum_{i \in A_j} k_{mod,i} f_{l,i}^3(t) \right] \\
 \text{s.t.} & \quad (9).
 \end{aligned} \tag{35}$$

It can be seen that  $SP1$  is a simple convex optimization problem. Then, we can obtain the optimal solution:

$$f_{l,i}^*(t) = \min \left\{ \sqrt{\frac{S_j(t)}{3V k_{mod,i} L_i}}, f_{l,i}^{\max}(t) \right\}. \tag{36}$$

**Task Offloading Control**

The optimal decision vector  $p_{r,i}(t)$  can be obtained by solving the sub-problem  $SP2$ , as follow:

$$\begin{aligned}
 SP2 : \min_{\{p_{r,i}(t)\}} & \sum_{j \in B} \left[ (M_j(t) - S_j(t)) \sum_{i \in A_j} B_i \log_2 \left( 1 + \frac{p_{r,i}(t)G_i Z_0}{B_i(t)N_0 d_{i,j}^2} \right) + V \sum_{i \in A_j} p_{r,i}(t) \right] \\
 \text{s.t.} & \quad (12).
 \end{aligned} \tag{37}$$

$SP2$  is a convex optimization problem, which can be proved via the Second-order derivation. We can solve it by taking the derivative. Let us denote

$$\gamma_i(t) = \frac{(S_j(t) - M_j(t))B_i}{V \ln 2} - \frac{N_0 B_i d_{i,j}^2}{G \cdot Z_0}. \tag{38}$$

Then, the  $p_i^*(t)$  can be written as:

$$p_{r,i}^*(t) = \min\{\gamma_i(t), p_{max,j}\}. \tag{39}$$

**Cloud Computing and Charging Control**

$f_{u,j}^*(t)$  and  $W_j^*(t)$  are defined as the optimal  $f_{u,j}(t)$  and the optimal  $W_j(t)$ . We can obtain them by solving  $SP3$ .

$$\begin{aligned}
 SP3 : \min_{\{W_j(t), W_j(t)\}} & \sum_{j \in B} \left[ (V - H_j(t))k_{mod,j} \tau f_{u,j}^3(t) - \frac{M_j(t)\tau}{L_j} f_{u,j}(t) + H_j(t)W_j(t) \right] \\
 \text{s.t.} & \quad (5), (6), (16).
 \end{aligned} \tag{40}$$

$SP3$  is a combinatorial optimization problem. This objective function is determined by two variables, i.e.,  $f_{u,j}(t)$  and  $W_j(t)$ . To obtain the optimal solution, we propose an AI-empowered algorithm based on Simulated Annealing (SA), as described in algorithm 2. The algorithm produces near-optimal solution with less computation cost and avoids falling into local optimal solution.

**Algorithm 2** : Simulated Annealing-based Cloud Computing and Charging Decision

- 1: **Input:**  $I_{max}, T_0, T_f, \alpha, H_j(t), M_j(t)$
- 2: **Output:**  $f_{u,j}^*(t), W_j^*(t)$
- 3: **for**  $j \in B$  **do**
- 4:     Initialize the  $T = T_0$
- 5:     **while**  $T_0 \leq T_f$  **do**
- 6:         **for**  $l = 0 \rightarrow I_{max}$  **do**
- 7:             Generate random feasible solution  $f_0$  and  $W_0$
- 8:             Calculate function value  $F_0$  as (40)
- 9:             Generate new feasible solutions  $f_{new}$  and  $W_{new}$
- 10:             Calculate new function value  $F_{new}$  as (40)
- 11:             **if**  $F_{new} \leq F_0$  **then**
- 12:                 Accept new solutions  $f_{new}$  and  $W_{new}$
- 13:             **else**
- 14:                  $p = e^{\frac{F_{new} - F_{old}}{T}}$
- 15:                 Accept  $f_{new}$  and  $W_{new}$  with probability  $p$
- 16:             **end if**
- 17:             **end for**
- 18:             Update annealing temperature  $T = \alpha T$
- 19:         **end while**
- 20:         Find the optimal solutions  $f_{u,j}^*(t)$  and  $W_j^*(t)$
- 21:         **return**  $f_{u,j}^*(t), W_j^*(t)$
- 22: **end for**

**Algorithm Analysis**

In this subsection, we use mathematical analysis to prove that the algorithm we proposed can achieve the arbitrary optimal solution on the premise of maintaining the stability of the system.

**Theorem 1** For all  $t > 0$  the time average profit and time average queue lengths satisfy:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{j \in B} \mathbb{E}\{E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)\} \leq e^* + \frac{B}{V}, \tag{41}$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{j \in B} \mathbb{E}\{S_j(t) + M_j(t) + H_j(t)\} \leq \frac{B + V(\hat{e} - \check{e})}{\varepsilon}, \tag{42}$$

where  $e^*$  is the optimal solution of the problem (21).

*Proof* Since we assume that arrival process is strictly within the network capacity region, there exists at least one stationary randomized control policy  $\pi^*$  that can stabilize the queue, which has following features:

$$\begin{aligned} \mathbb{E}\{D_{l,j}(t) + D_{r,j}(t) | \Theta(t)\} &\geq \mathbb{E}\{D_j(t) | \Theta(t)\} = \lambda \\ \mathbb{E}\{D_{u,j}(t) | \Theta(t)\} &\geq \mathbb{E}\{D_{r,j}(t) | \Theta(t)\} = \mu \\ \mathbb{E}\{E_{u,j}(t) | \Theta(t)\} &\geq \mathbb{E}\{W_j(t) | \Theta(t)\} = \gamma \end{aligned} \tag{43}$$

we can derive that there exists  $\varepsilon \geq 0$  satisfying:

$$\begin{aligned} \mathbb{E}\{D_{l,j}(t) + D_{r,j}(t) | \Theta(t)\} &= \lambda + \varepsilon \\ \mathbb{E}\{D_{u,j}(t) | \Theta(t)\} &= \mu + \varepsilon \\ \mathbb{E}\{E_{u,j}(t) | \Theta(t)\} &= \gamma + \varepsilon \end{aligned} \tag{44}$$

Then, by applying (26), we obtain:

$$\begin{aligned} &\Delta(\Theta(t)) + V \mathbb{E}\left\{ \sum_{j \in B} (E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)) | \Theta(t) \right\} \\ &\leq B + \mathbb{E}\left[ \sum_{j \in B} (S_j(t)\mu) | \Theta(t) \right] - \mathbb{E}\left[ \sum_{j \in B} (S_j(t)(\mu + \varepsilon)) | \Theta(t) \right] \\ &\quad + \mathbb{E}\left[ \sum_{j \in B} (M_j(t)\lambda) | \Theta(t) \right] - \mathbb{E}\left[ \sum_{j \in B} (M_j(t)(\lambda + \varepsilon)) | \Theta(t) \right] \\ &\quad + \mathbb{E}\left[ \sum_{j \in B} (H_j(t)\gamma) | \Theta(t) \right] - \mathbb{E}\left[ \sum_{j \in B} (H_j(t)(\gamma + \varepsilon)) | \Theta(t) \right] \\ &\quad + V \mathbb{E}\left\{ \sum_{j \in B} (E_{l,j}^{\pi^*}(t) + E_{r,j}^{\pi^*}(t) + E_{u,j}^{\pi^*}(t)) | \Theta(t) \right\} \\ &= B - \varepsilon \mathbb{E}\left[ \sum_{j \in B} (S_j(t) + M_j(t) + H_j(t)) | \Theta(t) \right] \\ &\quad + V \mathbb{E}\left\{ \sum_{j \in B} (E_{l,j}^{\pi^*}(t) + E_{r,j}^{\pi^*}(t) + E_{u,j}^{\pi^*}(t)) | \Theta(t) \right\}. \end{aligned} \tag{45}$$

Taking expectations on both sides and using iterated expectations, it can be obtained that:

$$\begin{aligned} &\mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t))\} + V \mathbb{E}\left\{ \sum_{j \in B} (E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)) \right\} \\ &\leq B - \varepsilon \mathbb{E}\left[ \sum_{j \in B} (S_j(t) + M_j(t) + H_j(t)) \right] + V e^*. \end{aligned} \tag{46}$$

Summing both sides of (46) over  $t \in \{0, 1, \dots, T-1\}$ , and then dividing both sides of  $T$ , we can yield:

$$\begin{aligned} &\frac{1}{T} \mathbb{E}\{L(\Theta(T)) - L(\Theta(0))\} + \frac{V}{T} \sum_{t=0}^{T-1} \sum_{j \in B} \mathbb{E}\{E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)\} \\ &\leq B - \frac{\varepsilon}{T} \sum_{t=0}^{T-1} \sum_{j \in B} \mathbb{E}\{S_j(t) + M_j(t) + H_j(t)\} + V e^*. \end{aligned} \tag{47}$$

By taking the limit  $T \rightarrow \infty$ , we can obtain:

$$\begin{aligned} &\lim_{T \rightarrow \infty} \frac{V}{T} \sum_{t=0}^{T-1} \sum_{j \in B} \mathbb{E}\{E_{l,j}(t) + E_{r,j}(t) + E_{u,j}(t)\} \leq B \\ &+ \lim_{T \rightarrow \infty} \frac{\varepsilon}{T} \sum_{t=0}^{T-1} \sum_{j \in B} \mathbb{E}\{S_j(t) + M_j(t) + H_j(t)\} + V e^*. \end{aligned} \tag{48}$$

Dividing both sides of (48) by  $V$  and taking a limit as  $\varepsilon \rightarrow \infty$ , it can be obtain that (41). By dividing both sides of (48) by  $\varepsilon$ , we can yield (42).

The term (41) shows that as the value of  $V$  increases, the time averaged energy consumption decreases. However, The term (42) indicates that as the tradeoff parameter  $V$  increases, the time average queue congestion also increases. Therefore, we need to make a tradeoff between energy consumption and queue stability by adjusting the value of  $V$ . Besides, according to (41), there is an upper bound  $\frac{B}{V}$  on the difference between CTORAA's utility

**Table 2** Experimental parameter setting

Parameter	Value
$\tau$	1s
$P_{max,j}$	10W
$E_h$	50
$E_{safe}$	5
$L_i$	737.5cycles/bit
$L_j$	737.5cycles/bit
$k_{mod,i}$	$1 \times 10^{-17}$
$k_{mod,j}$	$1 \times 10^{-16}$
$f_{l,max}$	$1 \times 10^9$
$f_{u,max}$	$2.5 \times 10^9$
$B_i$	10W
$G_i$	$1.42 \times 10^{-4}$
$N_0$	$1 \times 10^{-20}$ W/Hz
$Z_0$	2.2846

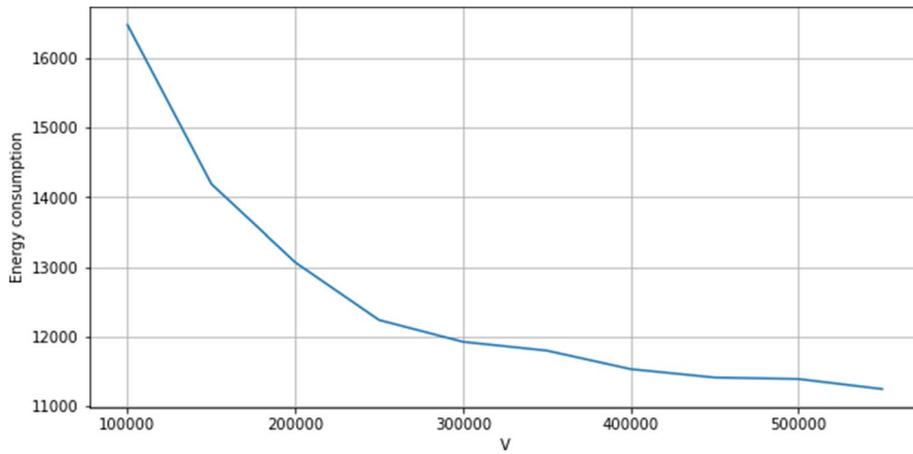


Fig. 2 Time-average energy consumption with different V

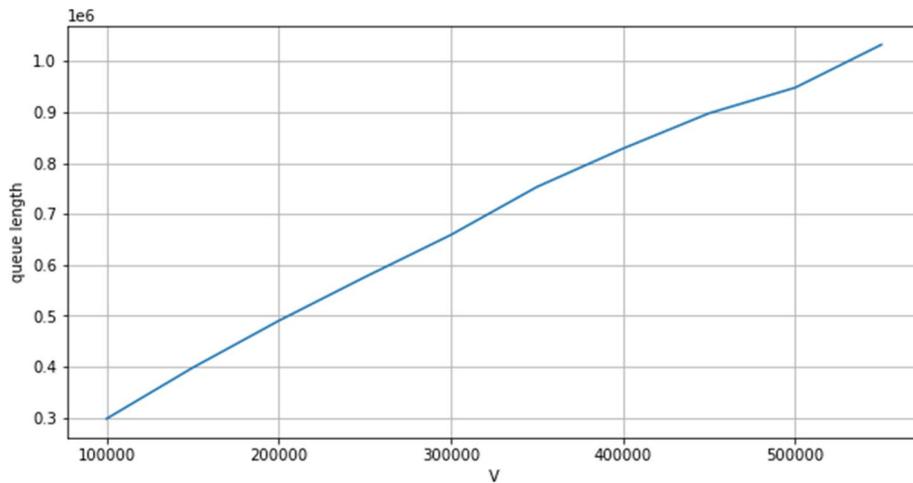


Fig. 3 Time-average queue length with different V

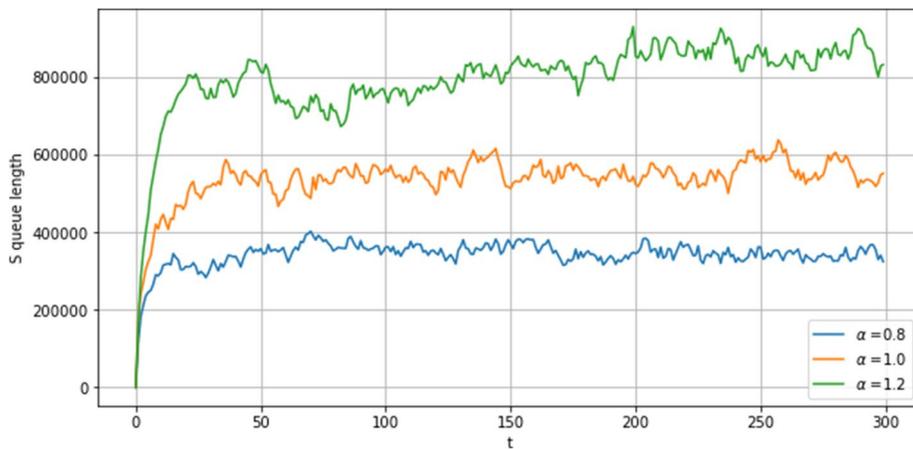
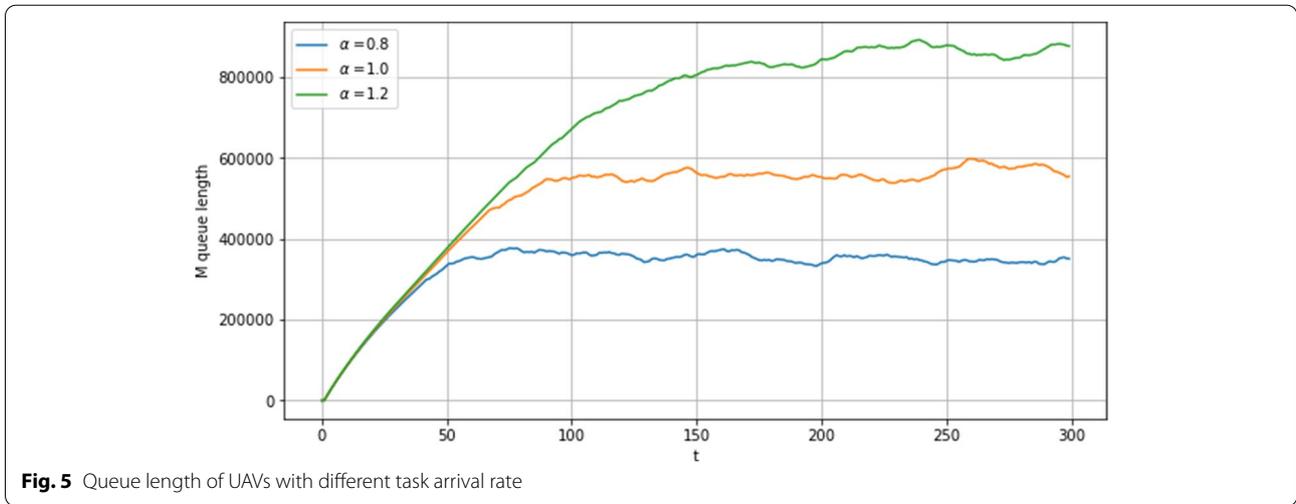


Fig. 4 Queue length of IoT devices with different task arrival rate



and the optimal utility. Furthermore, when  $V$  becomes large enough, the gap is small enough that the optimal solutions can be approximated by the final solutions.

### Experimental Evaluation

#### Experimental Setting

In the experiments, we consider a system consisting of  $X$  IoT devices and  $Y$  UAVs. In general, there is  $X = 30$ ,  $Y = 5$ . The  $d_i(t)$ s are independent and identically distributed (i.i.d) in different time slots with the uniform distribution (i.e.,  $d_i(t) \sim U[0, 10000]bit$ ), and  $d_{i,j} \in [100, 200]m$ . We set the parameters of the experiment in Table 2.

#### Simulation Results

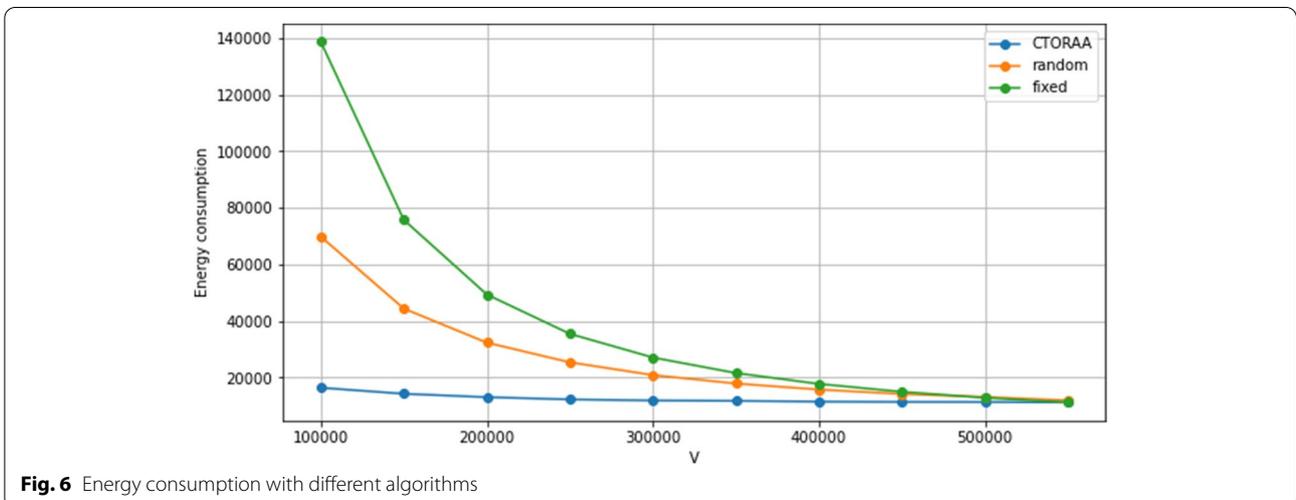
##### Impacts of the Tradeoff Parameter $V$

We set  $V = 1 \times 10^5, 1.5 \times 10^5, 2 \times 10^5, \dots, 5.5 \times 10^5$ . Figure 2 show the time-average energy consumption and

queue length of the UAV-assisted multi-clouds computing system with different  $V$ . In Fig. 3, it can be seen that the energy consumption decreases as the value of the tradeoff parameter  $V$  increases. And it converges to the minimum level for larger values of  $V$ . The agrees with the conclusion obtained by (41) in Theorem 1. In Fig. 3, it shows that the time-average queue congestion increases as the value of  $V$  increases, which is in accordance with (42) in Theorem 1. Together with Figs. 2 and 3, it can be seen that CTORAA can an achieve an arbitrary stability-profit tradeoff by changing the value of the tradeoff parameter  $V$ .

##### Impact of Task Arrival Rate

Figures 4 and 5 show task queue  $S$  and  $M$  with different task arrival rates respectively. In our experiments, the task arrival rate is denoted as  $\alpha \cdot d_i(t)$ . Specifically, we consider three different settings of  $\alpha$ , i.e.,  $\alpha = 0.8, 1$  and  $1.2$ . From two figures, it can be seen that the length



of task queues rises when the task arrival rate rises. We can see that the algorithm can adopt to the different task arrival rates and guarantee the queue stability.

### Comparison Experiment

In this subsection, in order to further present the performance of CTORAA, we compare our proposed algorithm with the two benchmark algorithms, namely random and fixed. Figure 6 shows the energy consumption with the three different algorithms.

It can be seen that the algorithm we proposed is closer to the optimal solution than the other two algorithms. In other words, our proposed algorithm is better than the other two algorithms.

### Conclusions

In this article, we study the problem of cooperative task offloading and resource allocation with hybrid energy supply for UAV-assisted multi-clouds. We consider the tradeoff between the energy consumption of the system and system performance. To capture the high dynamics of the task and schedule each UAV more reasonably for conserving the energy, we take advantage of the Lyapunov optimization techniques to design an algorithm namely CTORAA. The goal is to minimize the average energy consumption of the system with the constraint of queue stability. Then mathematical analysis shows that CTORAA can reach the arbitrary profit-stability tradeoff. Finally, experiments are conducted to validate the performance of our approach.

### Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions on improving this paper.

### Authors' Contributions

Yu Zhou designed the modeling approach and the algorithm, and wrote the paper. Hui Ge designed system model and analyzed the experimental data. Bowen Ma carried out the experiments. Shuhang Zhang contributed to data collection and analysis. Jiwei Huang conceived the initial idea and proofread the manuscript. The authors read and approved the final manuscript.

### Funding

This work is supported by Beijing Nova Program (No. Z201100006820082), National Natural Science Foundation of China (No. 61972414), and Beijing Natural Science Foundation (No. 4202066).

### Availability of data and materials

The datasets used during the current study are available from the corresponding author on reasonable request.

### Declarations

#### Ethics approval and consent to participate

Not applicable.

#### Consent for publication

Not applicable.

### Competing interests

The authors declare that they have no competing interests.

Received: 3 July 2022 Accepted: 17 August 2022

Published online: 09 September 2022

### References

- Ferraz Junior N, Silva AA, Guelfi AE et al (2022) Performance evaluation of publish-subscribe systems in IoT using energy-efficient and context-aware secure messages. *J Cloud Comp* 11:6. <https://doi.org/10.1186/s13677-022-00278-6>
- Huang J, Tong Z, Feng Z Geographical POI recommendation for Internet of Things: A federated learning approach using matrix factorization. *Int J Commun Syst* e5161. <https://doi.org/10.1002/dac.5161>
- Ying C, Hua X, Zhuo M, et al (2022) Cost-Efficient Edge Caching for NOMA-enabled IoT Services. *China Commun*
- Huang J, Wang M, Wu Y, Chen Y, Shen X (2022) Distributed Offloading in Overlapping Areas of Mobile-Edge Computing for Internet of Things. *IEEE Internet Things J* 9(15):13837–13847
- Chen Y, Liu Z, Zhang Y, Wu Y, Chen X, Zhao L (2021) Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things. *IEEE Trans Ind Inform* 17(7):4925–4934
- Dey S, Sampalli S, Ye Q (2016) MDA: message digest-based authentication for mobile cloud computing. *J Cloud Comp* 5:18. <https://doi.org/10.1186/s13677-016-0068-6>
- Qureshi SS, Ahmad T, Rafique K, Shuja-ul-islam (2011) Mobile cloud computing as future for mobile applications - Implementation methods and challenging issues. 2011 IEEE International Conference on Cloud Computing and Intelligence Systems. p. 467–471. <https://doi.org/10.1109/CCIS.2011.6045111>
- Zhang Y, Pan J, Qi L, He Q (2021) Privacy-preserving quality prediction for edge-based IoT services. *Futur Gener Comput Syst* 114:336–348
- Xu J, Li D, Gu W et al (2022) UAV-assisted Task Offloading for IoT in Smart Buildings and Environment via Deep Reinforcement Learning. *Building and Environment*. <https://doi.org/10.1016/j.buildenv.2022.109218>
- Almutairi J, Aldossary M (2021) A novel approach for IoT tasks offloading in edge-cloud environments. *J Cloud Comp* 10:28. <https://doi.org/10.1186/s13677-021-00243-9>
- Chen Y, Gu W, Li K Dynamic task offloading for Internet of Things in mobile edge computing via deep reinforcement learning. *Int J Commun Syst*. e5154. <https://doi.org/10.1002/dac.5154>
- Rong G, Xu Y, Tong X et al (2021) An edge-cloud collaborative computing platform for building AIoT applications efficiently. *J Cloud Comp* 10:36. <https://doi.org/10.1186/s13677-021-00250-w>
- Gao A, Wang Q, Liang W, Ding Z (2021) Game Combined Multi-Agent Reinforcement Learning Approach for UAV Assisted Offloading. *IEEE Trans Veh Technol* 70(12):12888–12901
- Wang Q, Gao A, Hu Y (2021) Joint Power and QoS Optimization Scheme for Multi-UAV Assisted Offloading in Mobile Computing. *IEEE Access* 9:21206–21217
- Gao A, Hu Y, Liang W, Lin Y, Li L, Li X (2019) A QoS-Oriented Scheduling Scheme for Energy-Efficient Computation Offloading in UAV Cloud System. *IEEE Access* 7:68656–68668
- Hu J, Jiang M, Zhang Q, Li Q, Qin J (2019) Joint Optimization of UAV Position, Time Slot Allocation, and Computation Task Partition in Multi-user Aerial Mobile-Edge Computing Systems. *IEEE Trans Veh Technol* 68(7):7231–7235
- Zeng Y, Xu J, Zhang R (2019) Energy Minimization for Wireless Communication With Rotary-Wing UAV. *IEEE Trans Wirel Commun* 18(4):2329–2345
- Du Y, Yang K, Wang K, Zhang G, Zhao Y, Chen D (2019) Joint Resources and Workflow Scheduling in UAV-Enabled Wirelessly-Powered MEC for IoT Systems. *IEEE Trans Veh Technol* 68(10):10187–10200
- Khalil MI (2021) Delay and Energy Balance for Unmanned Aerial Vehicle Networks. 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). p 1453–1458. <https://doi.org/10.1109/CCWC51732.2021.9376092>
- Wang Y et al (2020) Multi-UAV Collaborative Data Collection for IoT Devices Powered by Battery. 2020 IEEE Wireless Communications and

- Networking Conference (WCNC). p 1–6. <https://doi.org/10.1109/WCNC45663.2020.9120646>
21. Zhan C, Hu H, Sui X, Liu Z, Niyato D (2020) Completion Time and Energy Optimization in the UAV-Enabled Mobile-Edge Computing System. *IEEE Internet Things J* 7(8):7808–7822
  22. Zhang T, Zhu X, Zhou Z, Wang R, Chen R (2018) Energy Management of Solar UAV Level Flight. 2018 IEEE International Conference on Prognostics and Health Management (ICPHM). p 1–6. <https://doi.org/10.1109/ICPHM.2018.8448508>
  23. Wang Y, Wang H, Wei X (2020) Energy-Efficient UAV Deployment and Task Scheduling in Multi-UAV Edge Computing. 2020 International Conference on Wireless Communications and Signal Processing (WCSP). p 1147–1152. <https://doi.org/10.1109/WCSP49889.2020.9299765>
  24. Chen Y, Zhao F, Lu Y, Chen X Dynamic task offloading for mobile edge computing with hybrid energy supply. *Tsinghua Sci Technol*. <https://doi.org/10.26599/TST.2021.9010050>
  25. Chen Y, Zhao F, Chen X, Wu Y (2022) Efficient Multi-Vehicle Task Offloading for Mobile Edge Computing in 6G Networks. *IEEE Trans Veh Technol* 71(5):4584–4595
  26. (2014) Preface. In: Ross S (ed) *Introduction to Probability Models*, 11th edn. Boston, Academic Press. <https://doi.org/10.1016/B978-0-12-407948-9.00016-5> <https://www.sciencedirect.com/science/article/pii/B9780124079489000165>
  27. Nouri N, Entezari A, Abouei J, Jaseemuddin M, Anpalagan A (2020) Dynamic Power-Latency Tradeoff for Mobile Edge Computation Offloading in NOMA-Based Networks. *IEEE Internet Things J* 7(4):2763–2776
  28. Jiwei H, Chenxiang Z, Jianbing Z (2020) A Multi-queue Approach of Energy Efficient Task Scheduling for Sensor Hubs. *Chin J Electron* 29(2020–2–242):242
  29. Huang J, Lv B, Wu Y et al (2022) Dynamic Admission Control and Resource Allocation for Mobile Edge Computing Enabled Small Cell Network. *IEEE Trans Veh Technol* 71(2):1964–1973
  30. Neely MJ (2010) *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, San Rafael
  31. Tassiulas L, Ephremides A (1990) Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. 29th IEEE Conference on Decision and Control, vol 4. p 2130–2132. <https://doi.org/10.1109/CDC.1990.204000>
  32. Neely MJ, Modiano E, Li C (2008) Fairness and Optimal Stochastic Control for Heterogeneous Networks. *IEEE/ACM Trans Netw* 16(2):396–409

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---