Computer Networks 205 (2022) 108710

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A Novel Lyapunov based Dynamic Resource Allocation for UAVs-assisted Edge Computing

Jie Lin [a], Lin Huang [a], Hanlin Zhang [b], Xinyu Yang [a,*], Peng Zhao [a]

[a] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
[b] Qingdao University, Qingdao, China

ARTICLE INFO

Keywords:
UAVs-assisted edge computing
Dynamic resource allocation
Lyapunov optimization
System cost and utility

ABSTRACT

Mobile edge computing (MEC), as a key component in the development of IoT and 5G technologies, can provide extra computation resources in edge servers for mobile devices to complete their computation tasks with low latency and high reliability. Considerable efforts on computation offloading and resource allocation have been developed to reduce the energy consumption and computation latency in edge computing. Nonetheless, the system utility of heterogeneous edge computing system (e.g., UAVs-assisted edge computing), in which multiple unmanned aerial vehicles (UAVs) are involved in an edge computing system to serve as edge servers still needs to be further investigated. To this end, in this paper, we propose a novel Lyapunov based Dynamic Resource Allocation (LDRA) for UAVs-assisted Mobile Edge Computing, which can effectively choose suitable edge servers for mobile devices to offload and complete their computation tasks with low system cost and great system utility of UAVs-assisted edge computing system, as well as acceptable computation latency and great reliability for computation tasks of mobile devices. Particularly, a random queue model for edge servers is conducted in our LDRA scheme to support the dynamic of offloaded computation tasks of mobile devices. Additionally, a system cost model of UAVs-assisted edge computing is developed considering the combination of multiple constraints, such as both the mobility of UAVs and mobile devices, energy consumption, communication cost, etc. With the objective of minimizing the system cost and maximizing the system utility in providing edge resources to complete the offloaded computation tasks of mobile devices, by introducing Lyapunov optimization, a dynamic resource allocation scheme is proposed to effectively determine edge servers to offload tasks of mobile devices with considering both the real-time execution state of offloaded tasks in edge servers and states of the communication link. Through analysis and performance evaluations, our results show that our proposed LDRA scheme can achieve a great balance between system cost and system stability. Additionally, our results also demonstrate that our LDRA scheme also can achieve better system utility in comparison with existing schemes.

1. Introduction

Mobile edge computing (MEC) can deploy edge servers close to the mobile end-devices to provide extra computation, communication and storage resource to assist mobile end-devices in completing their computation tasks with low latency and great reliability [1–3]. Most Internet-of-things (IoT) applications merged with MEC (e.g., smart transportation, emergency television transmission, etc.) are deployed with fixed edge servers (e.g., road-side units) and cannot effective deal with the sharply increased communication and computation offloaded by mobile end-devices in emergency hours. For example, during the traffic peak hours in smart transportation system, the communication and computation tasks will be increased sharply around the congested roads [4], and it is costly and will be achieving low-cost performance to deploy more fixed edge servers to assist in completing these increased communication and computation tasks due to the traffic peak hours are very short periods of time during the day and these added fixed edge servers will be idle most of the time. Hence, in these scenarios, taking the flexible, plug and play unmanned aerial vehicles (UAVs) as the mobile edge servers in the MEC system can achieve high cost performance and flexibility for MEC system [5,6].

Unmanned aerial vehicles (UAVs), taking advantage of flexible deployment, quick response and wide coverage, have been widely involved in MEC to serve as mobile edge servers to provide extra computation and communication resources for mobile devices in task peak hours [5,6]. Due to the UAV are powered by portable batteries, the

* Corresponding author.
E-mail addresses: jielin@mail.xjtu.edu.cn (J. Lin), hl2014@stu.xjtu.edu.cn (L. Huang), hanlin@qdu.edu.cn (H. Zhang), yxyphd@mail.xjtu.edu.cn (X. Yang), p.zhao@mail.xjtu.edu.cn (P. Zhao).

https://doi.org/10.1016/j.comnet.2021.108710
Received 14 July 2021; Received in revised form 25 October 2021; Accepted 13 December 2021
Available online 4 January 2022

energy consumption of UAV in computation offloading and resource allocation must be carefully investigated [7]. Recently, many efforts on computation offloading and resource allocation have been developed in UAVs-assisted MEC. Some existing efforts focused on developing the framework of UAVs-assisted MEC. For example, Zhan et al. [8] proposed a novel UAVs-assisted MEC framework, which can maximize the number of serviced mobile devices through joint optimization of UAV trajectories and service metrics, as well as resource allocation and computational offloading. Liao et al. [9] proposed a UAVs-assisted MEC framework, namely HOTSPOT, which can determine the position of UAVs through the distribution of customers and then determine the task offloading schedules to balance the tasks offloaded to UAVs.

A number of existing efforts also have been developed to minimize the energy consumption and task latency for UAVs-assisted MEC. For instance, Ji et al. [10] proposed a model to jointly optimize the resource allocation and trajectory of UAVs with the objective of minimizing the energy consumption. Sun et al. [11] proposed a tasks offloading scheme to minimize the weighted total energy consumption of UAVs and end-devices through jointly optimization of bit allocation, transmitting power, CPU frequency, bandwidth allocation, and UAV trajectory. Zhao et al. [12] considered wireless powered UAV-assisted MEC with only one UAV existed and proposed a scheme to minimize the energy consumption of UAV via adjusting communication power. Cheng et al. [13] and Cao et al. [14] focused on the tasks completed time in UAVs. In addition, other existing efforts also investigated the deployment of UAVs in MEC [9], deep-learning based tasks allocation in UAVs [15,16], secure communications for UAV-assisted MEC [17], etc.

Although a number of efforts on tasks offloading and resource allocations have been developed with the objective of minimizing the energy consumption and tasks latency, the sustainability of UAVs-assisted MEC with a group of UAVs collaboratively served as mobile edge servers still need to be further investigated. Therefore, designing a dynamic computation offloading and resource allocation scheme for UAVs-assisted MEC with considering the mobility, energy consumption, as well as computation, communication, and storage capabilities of a group of UAVs to reduce the total system cost and improve total system utility (i.e., extend the sustainability of the UAVs-assisted MEC) is still an open issue.

To address this issue, in this paper, a UAVs-assisted edge computing system is considered in which a group of UAVs are collaboratively served as mobile edge servers to provide extra computation, communication, and storage capabilities for mobile devices, and a Lyapunov-based dynamic resource allocation scheme (namely LDRA) is proposed to dynamically assign the resource of UAVs (i.e., mobile edge servers) for offloaded computation tasks of mobile devices to complete these tasks with the objective of minimizing system cost and maximizing system utility of UAVs-assisted edge computing systems, as well as guaranteeing the computation and communication latency of computation tasks of mobile devices. Note that, in our paper, the system utility represents the sustainability of the UAVs-assisted MEC, which is defined as the total number of offloaded computation tasks that are processed by mobile edge servers over a long period of time.

The contributions in this paper are summarized as follows.

First, to formalize the system sustainability, a system cost model with considering multiple factors, including the mobility of both UAVs and mobile devises, the communication cost among UAVs and mobile devises, the execution of offloaded computation tasks, as well as the energy consumption of UAVs, is conducted for UAVs-assisted edge computing system. In addition, a random queue model with considering the dynamics of computation task offloading of mobile devices is also proposed to achieve the reliable and stable execution for offloaded computation tasks.

Second, matching resource of UAVs to offloaded tasks of mobile devices is formalized as an optimization problem with the objective of minimizing the system cost of UAVs-assisted MEC system, and a Lyapunov-based dynamic resource allocation scheme (namely LDRA) is proposed to transform the time-dependent long-term optimization problem to a time-independent optimization problem, and then a greedy-based matching parameter determination algorithm is proposed to solve the optimization problem and find the optimal matching parameter between UAVs and mobile devices to complete the offloaded tasks with minimum system cost and maximum utility of UAVs-assisted MEC system, as well as the acceptable latency of offloaded tasks. The proposed LDRA can synergistically consider the energy consumption of a group of UAVs, guarantee the latency and reliability of offloaded computation tasks, as well as achieve effective utilization of resources in UAVs without any predictions of the arrival of future offloaded computation tasks.

Lastly, a number of evaluations have been conducted to demonstrate the effectiveness of our LDRA scheme with respect to system stability, system cost, and system utility in comparison with two existing schemes (i.e., JSQ and RSA). Our evaluation results show that, in the proposed LDRA scheme, the backlogs of offloaded tasks in the task queue of UAVs can converge, which means the proposed LDRA scheme can achieve great system stability. The evaluation results also show that the proposed LDRA scheme can achieve lower system cost and greater system utility while achieving the similar execution efficiency of offloaded computation tasks in comparison with the existing schemes (i.e., JSQ and RSA). That means our LDRA scheme can achieve better sustainability for the UAVs-assisted MEC system, as well as better stability and reliability for offloaded tasks execution.

The remainder of the paper is organized as follows: We introduce system models in Section 2. We formalize the problem in Section 3. We present our proposed scheme in Section 4. In Section 5, we show experimental results to validate our findings. We review related works in Section 6 and conclude this paper in Section 7, respectively.

2. System model

In this paper, the UAVs-assisted edge computing system with resource-constrained edge servers is considered, in which a group of UAVs are involved in the MEC to serve as mobile edge servers to provide extra computation, communication, and storage capabilities to mobile devices. The UAVs involved in MEC have limited resources and are powered by batteries, i.e., energy-constrained. Hence, collaborative management of energy consumption of these UAVs is important to extend the stability, sustainability, and longevity of the UAVs-assisted MEC.

In our model, multiple mobile devices, multiple mobile edge servers are considered in the system, as shown in Fig. 1, in which mobile devices move on the ground and mobile edge servers are deployed in the air. Here, the mobile devices can be the devices with limited computation and communication capabilities, such as smartphones, the embedded devices in smart vehicles, etc. Mobile devices may communicate with multiple mobile edge servers that are close to them and offload their computation tasks to mobile edge servers whenever necessary. To simplify, in our model UAVs are considered to fly stably at a fixed altitude within a certain period of time. Note that, due to fixed edge servers can be reliably powered and the utility of UAVs-assisted MEC is mainly influenced by the utility of mobile edge servers (i.e., UAVs), in our model only UAVs serving as mobile edge servers are considered and fixed edge servers are not considered. Additionally, due to the motionless, actually fixed edge servers can also be easy to be merged in our model.

In addition, we assume that the communication between mobile devices and mobile edge servers (i.e., UAVs) is reliable. That is the offloaded computation tasks can be reliably transmitted from mobile devices to mobile edge servers. Even if unstable communication exists between mobile devices and mobile edge servers, some existing efforts on users' quality of service in tasks offloading can be effectively applied in our model.

Fig. 1. System Model.

Table 1
Notation.

t:	Time slot
m_j:	Mobile device
$J_{m_j}(t)$:	The computation task of mobile device m_j at time slot t.
n_i:	Mobile edge server, i.e., the UAV.
$(x(t), y(t))$:	The projection coordinate of mobile devices or mobile edge servers at time slot t.
H:	The altitude of mobile edge servers.
$h_{n_i}^{m_j}(t)$:	The channel power gain from mobile device m_j to mobile edge server n_i at time slot t.
$q_{n_i}^{m_j}(t)$:	The data transmission rate between mobile device m_j to mobile edge server n_i at time slot t.
$T_{comp}(t)$:	The total computation latency of all mobile edge servers (i.e., UAVs) at time slot t.
$T comm(t)$:	The total computation transmission latency of all mobile edge servers (i.e., UAVs) at time slot t.
$p_{m_j}(t)$:	The transmitting power of mobile device m_j at time slot t.
β_o:	The unit channel power gain.
$Q_{n_i}(t)$:	The length of task queue in mobile edge server n_i at time slot t.
$A_{n_i}(t)$:	The total number of tasks received by mobile edge server n_i at time slot t.
$L_{n_i}(t)$:	The number of tasks processed by mobile edge server n_i at time slot t.
$\overline{Q}$:	The average length of task queue (i.e., task backlog) of all mobile edge servers.
$E_{n_i}^{comp}(t)$:	The energy consumption for task computation of mobile edge server n_i at time slot t.
F_{n_i}:	The total computation capability of mobile edge server n_i.
$E_{n_i}^{comm}(t)$:	The energy consumption for task communication of mobile edge server n_i at time slot t.
$E_{n_i}^{fly}(t)$:	The energy consumption for flying of mobile edge server n_i at time slot t.
$E_{n_i}^{hold}(t)$:	The energy consumption for holding in air of mobile edge server n_i at time slot t.
$E_{total}(t)$:	The total energy consumption all mobile edge servers (i.e., UAVs) at time slot t.
$E_{n_i}^{Max}$:	The maximum energy stated in the battery of UAV n_i (i.e., the battery capability)
$P(t)$:	The system cost of UAVs-assisted MEC at time slot t.

3. Problem formalization

To effectively offload tasks of mobile devices to mobile edge servers with minimum system cost, thereby extending the stability, sustainability, and the longevity of the UAVs-assisted MEC, the computation offloading model, communication model, offloaded task queue model, and system cost model should be investigated first. Hence, in the following, we first conduct the computation offloading model of mobile devices, communication model among mobile devices and mobile edge servers, the offloaded tasks queue model in mobile edge servers, and the system cost model of UAVs-assisted MEC, respectively. Then, the resource allocation in UAVs-assisted MEC is formalized as an optimization problem. All notations are shown in Table 1. Note that, in our paper, the mobile edge servers mean the servers embedded in UAVs.

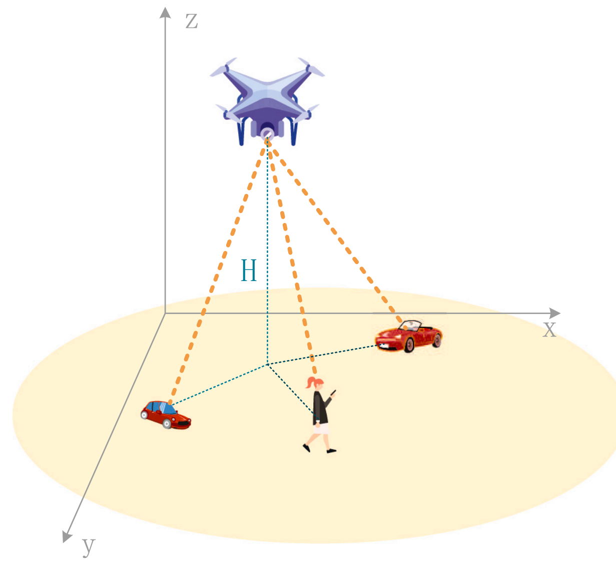


Fig. 2. Wireless communication of UAVs-assisted MEC in three dimensional Cartesian coordinates.

3.1. Computation offloading model

In this paper, the discrete-time model is considered and t represents the time slot. In each time slot t, a number of computation tasks of each mobile device are needed to be offloaded to mobile edge servers to be processed in order to guarantee the latency constraints. Note that, in our model, full computation offloading is considered. That is the task needed to be offloaded cannot be partitioned and can only be processed by one mobile edge server, which can be represented as

$$\sum_{i=1}^{N} \delta_{n_i}^{m_j}(t) = 1 \qquad (\forall j = 1, 2, \ldots, M), \tag{1}$$

where N and M are the number of mobile edge servers and mobile devices in the UAVs-assisted MEC, respectively. n_i and m_j represents the mobile edge server and mobile device, respectively. $\delta_{n_i}^{m_j}(t)$ is a boolean function (i.e., 0 or 1). If $\delta_{n_i}^{m_j}(t)$ gets 1, that means in time slot t the computation task of mobile device m_j is offloaded to mobile edge server n_i to be processed. To the contrary, If $\delta_{n_i}^{m_j}(t)$ gets 0, that means in time slot t the computation task of mobile device m_j cannot be offloaded to mobile edge server n_i.

In the meantime, the total received tasks in mobile edge server n_i should not be exceeded its capability, which can be represented as

$$A_{n_i}(t) = \sum_{j=1}^{M} \left(J_{m_j}(t) \cdot \delta_{n_i}^{m_j}(t) \right), \tag{2}$$

where $J_{m_j}(t)$ is the computation task of mobile device m_j in time slot t, $A_{n_i}(t)$ represents the received tasks of mobile edge server n_i in time slot t.

3.2. Communication model

In our model, by introducing the three-dimensional Cartesian coordinates [18], an air-to-ground wireless communication system is considered, in which the mobile devices move on the ground and mobile edge servers (i.e., UAVs) move in the air with fixed flight altitude, as shown in Fig. 2. Assuming that mobile devices and mobile edge servers are deployed with automatic positioning equipment (i.e., GPS etc.), and then the coordinate position of mobile devices in time slot t can be obtained, which can be denoted as $(x_{m_j}(t), y_{m_j}(t))$, where $x_{m_j}(t)$ and $y_{m_j}(t)$ represents the coordinate of x-axis and y-axis

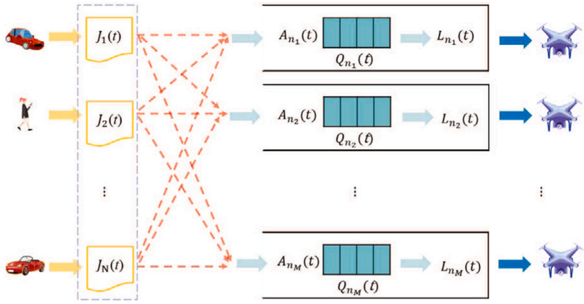


Fig. 3. An example of task queue model in mobile edge servers.

in three-dimensional Cartesian coordinates, respectively. Similarly, the projection coordinates of the mobile edge server in time slot t, denoted as $(x_{n_i}(t), y_{n_i}(t))$, can be obtained as well. In this scenario, the Euclidean distance between mobile edge server n_i and mobile device m_j in time slot t can be obtained by

$$d_{n_i}^{m_j}(t) = \sqrt{H^2 + \|(x_{n_i}(t), y_{n_i}(t)) - (x_{m_j}(t), y_{m_j}(t))\|^2}, \tag{3}$$

where H represents the flight altitude of mobile edge server n_i.

Due to the mobile edge servers fly in the air at a proper altitude, a Line-of-Sight (LoS) communication link can be established and dominate the communication between mobile edge servers and mobile devices, thereby reducing signal jamming and shadows. Additionally, in our model, a quasi static scenario is considered, in which the positions of mobile devices and mobile edge servers are considered as being static at a time slot. Actually, this is a valid assumption if the time slot is small and the mobile devices (and mobile edge servers) do not move fast. For simplicity, the small-signal fading caused by the shadows is ignored in our model. Hence, in this scenario, the channel power gain depends mainly on the air-to-ground distance and follows the free-space path loss model [19,20]. To this end, in our model, the channel power gain from mobile device m_j to mobile edge server n_i in time slot t can be represented by

$$\begin{aligned} h_{n_i}^{m_j}(t) &= \beta_o \cdot (d_{n_i}^{m_j}(t))^{-2} \\ &= \frac{\beta_o}{H^2 + \|(x_{n_i}(t), y_{n_i}(t)) - (x_{m_j}(t), y_{m_j}(t))\|^2}, \end{aligned} \tag{4}$$

where β_o is the unit channel power gain, i.e., the channel power gain at the communication distance of one meter.

3.3. Task queue model

The random task model is considered in our paper, in which the computation tasks are randomly generated by the mobile devices and offloaded to the mobile edge servers. Thus, due to the limited resources, the mobile edge servers usually cannot complete all offloaded computation tasks in a time slot. Hence, to complete offloaded computation tasks with low latency and great reliability, a task queue model should be considered in the mobile edge servers. In our model, multiple mobile devices and multiple mobile edge servers are considered, as shown in Fig. 3, in which each mobile edge server maintains a tasks queue that can cache the computation tasks offloaded by mobile devices.

In time slot $t+1$, the length of task queue (also representing the task backlog) in mobile edge server n_i, denoted as $Q_{n_i}(t+1)$, can be updated by

$$Q_{n_i}(t+1) = \max\{Q_{n_i}(t) - L_{n_i}(t), 0\} + A_{n_i}(t), \tag{5}$$

where $L_{n_i}(t)$ and $A_{n_i}(t)$ represents the completed tasks and received tasks of mobile edge server n_i at time slot t. Eq. (5) indicates that the offloaded tasks at time slot t will be cached in the task queue and be processed in the next time slot. Hence, it is obvious that the length of the task queue will be growth if the completed task loads are smaller than the received task loads at a time slot (i.e., appearing the task backlog aggravation). Conversely, the length of the task queue will be shorten (i.e, the queue backlog decreases). Additionally, due to the limited resource in mobile edge servers, a maximum task loads that a mobile edge server can be processed and completed at a time slot should be existed, denoted as $L_{n_i}^{max}$, and at each time slot, the processed and completed task loads should be less the maximum task loads, i.e., $L_{n_i}(t) \le L_{n_i}^{max}$.

In addition, to guarantee the reliability of UAVs-assisted MEC, the stability of the task queue should be considered. That is, the task queue should not grow indefinitely, resulting in long wait times for offloaded tasks. Hence, the length of the task queue should be met the following constraints, i.e.,

$$\overline{Q} = \lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=0}^{N} E\{Q_{n_i}(t)\}, \tag{6}$$

where $\overline{Q}$ is the average length of task queue (i.e., task backlog) of all mobile edge servers, and $E\{Q_{n_i}\}$ represents the expectation length of task queue in mobile edge server n_i. Eq. (6) ensures that the task queue model is bounded and stable.

Obviously, the longer the task queue (i.e., the larger the queue backlog), the longer the computation task waits. Hence, to ensure the reliability of task execution, our paper considers the upper limit of the task queue, denoted as $Q_{n_i}^{max}$. If and only if the length of task queue in a mobile edge server is smaller than the upper limit of the task queue (i.e., $Q_{n_i}(t) \le Q_{n_i}^{max}$), the computation tasks of mobile devices can be offloaded to the mobile edge servers with considering the latency.

3.4. System cost model

Due to the UAVs are considered as mobile edge servers and usually powered by batteries (i.e., energy constraints), the energy consumption of computation offloading and data transmission should be carefully investigated. In addition, computation and communication latency will be inevitably appeared in the computation task offloading. To this end, in this section, we conduct a system cost model that considered both energy consumption and task latency. Note that our model mainly focuses on energy consumption and latency of the offloaded tasks in mobile edge servers, and the tasks completed locally in mobile devices are not considered in our model because these locally completed tasks will not affect the performance of mobile edge servers.

3.4.1. Energy consumption model

The energy consumption of a mobile edge server usually consists of the energy computation from task computation, data transmission, and UAV kinetic.

Energy computation from task computation: At each time slot, the occupied computation resource depends on the task loads needed to be completed and should not be larger than the maximum computation resource owned as well [21], which can be represented as

$$f_{n_i}(t) = \frac{\mu \cdot L_{n_i}(t)}{\tau} \le F_{n_i}, \tag{7}$$

where $f_{n_i}(t)$ is the occupied computation resource of mobile edge server n_i at time slot t, F_{n_i} is the maximum computation resource owned by mobile edge server n_i, μ is the transforming parameter, indicating the relationship between task loads and the needed computation resource, and τ is the length of time slot.

During the task computation, the energy consumption mainly comes from CPU. Additionally, in most cases, the CPU cycle frequency is approximately linear to the CPU voltage. Hence, the CPU cycle frequency and energy consumption for the task computation can be adjusted by the CPU voltage [21], and the energy consumption from task computation can be represented as

$$E_{n_i}^{comp}(t) = \kappa \cdot \tau \cdot (f_{n_i}(t))^3, \tag{8}$$

where κ is the effective switching capacitance of CPU and depends on the CPU hardware architecture and usually determined as $\kappa = 10^{-28}$.

Therefore, the total energy consumption from task computation in UAVs-assisted MEC can be obtained by

$$E_{comp}(t) = \sum_{i=1}^{N} E_{n_i}^{comp}(t) = \sum_{i=1}^{N} \left(\kappa \cdot \tau \cdot (f_{n_i}(t))^3\right). \tag{9}$$

Energy computation from data transmission: As mentioned above, the computation tasks will be offloaded from mobile devices to mobile edge servers through LoS communication links, and thus the energy consumption from data transmission will appear for the mobile edge servers.

Usually the energy consumption from data transmission depends on the transmission power and can be obtained by

$$E_{n_i}^{comm}(t) = p_{n_i} \cdot A_{n_i}(t), \tag{10}$$

where p_{n_i} represents the energy consumption generated by receiving 1 *bit* data for mobile edge servers. Note that, here, the energy consumption from data transmission in mobile edge servers only considered the energy consumption generated by receiving offloaded tasks and does not consider the energy consumption generated by sending the results back to mobile devices. The reason is that usually, the computation results of offloaded tasks are small, and thus the energy consumed by sending the results back to mobile devices is ignored in our model. Eq. (10) shows that the energy consumption from data transmission in mobile edge servers is linear to the received task loads, i.e., as the increase of received task loads, the energy consumption from data transmission increases.

Therefore, the total energy consumption from data transmission in UAVs-assisted MEC can be obtained by

$$E_{comm}(t) = \sum_{i=1}^{N} E_{n_i}^{comm}(t) = \sum_{i=1}^{N} \left(p_{n_i} \cdot A_{n_i}(t)\right). \tag{11}$$

Energy computation from UAV kinetic: In UAVs-assisted MEC, the energy consumption for UAV flying consists of gravitational potential energy consumption, kinetic energy (i.e., the energy consumption for UAV flying or holding in the air). Due to in our model, UAVs are considered to fly with a fixed flight altitude, the gravitational potential energy consumption will be kept constant. Hence, only the energy consumption for UAV flying and holding in the air is considered in our model, in which the energy consumption for UAV flying is related to the moving speed of UAVs [7,21] and can be represented as

$$E_{n_i}^{fly}(t) = \frac{c \cdot \|v_{n_i}(t)\|^2}{\tau}, \tag{12}$$

$$v_{n_i}(t) = \|(x_{n_i}(t), y_{n_i}(t)) - (x_{n_i}(t-1), y_{n_i}(t-1))\| \le v_{n_i}^{max}, \tag{13}$$

where c is the flight energy consumption coefficient and usually determined as $0.5 \cdot W_{n_i} \cdot \tau$ and W_{n_i} is the weight of the mobile edge server n_i (i.e., UAVs), $v_{n_i}(t)$ and $v_{n_i}^{max}$ is the real-time speed at time slot t and maximum speed of mobile edge server n_i.

When the UAVs hold in the air, the energy consumption for UAV holding can be considered as content, denoted as $E_{n_i}^{hold}$, which is only related to the weight of the mobile edge server n_i. Hence, the total energy consumption from UAV kinetic is determined by the energy consumption for both UAV flying and holding and can be represented as

$$E_{cki}(t) = \sum_{i=1}^{N} \left(E_{n_i}^{fly}(t) + E_{n_i}^{hold}\right). \tag{14}$$

As mentioned above, the total energy consumption of mobile edge servers (i.e., UAVs) includes the energy computation from task computation, data transmission, and UAV kinetic and thus can be represented as

$$E_{total}(t) = E_{comp}(t) + E_{comm}(t) + E_{cki}(t). \tag{15}$$

Note that, Due to the execution of offloaded tasks only is related to the energy consumption for task computation and data transmission (i.e., task communication and computation), the tasks communication and computation energy consumption of UAVs for task offloading should be investigated. While, UAVs are powered by batteries and the UAVs kinetic energy consumption will directly affect the active time of UAV, and thus the UAVs kinetic energy consumption should be considered in the total energy consumption of UAVs. However, the energy consumption from task computation and data transmission may much smaller than the energy consumption from UAVs kinetic, and the magnitude difference will affect the investigation of the tasks communication and computation energy consumption of UAVs for task offloading. Hence, as mentioned in other existing literatures [21–23], an incentive factor, denoted as ω, is introduced in our total energy consumption model to mitigate the magnitude differences, thereby assisting in investigating the impact of computation offloading and resource allocation schemes on energy consumption from task computation and data transmission. Therefore, the updated energy consumption model in our paper can be represented as

$$E'_{total}(t) = \omega \cdot (E_{comp}(t) + E_{comm}(t)) + E_{cki}(t). \tag{16}$$

3.4.2. Latency model

As mentioned in Section 2, the reliable LoS communication links are considered in our model, in which the links between mobile devices and mobile edge servers will not be disconnected during the data transmission. The main latency of task offloading consists of computation latency and communication latency. According to the Aromatic theorem [24], at time slot t, the communication latency between mobile devices m_j and mobile edge server n_i is related to the loads of offloaded tasks $J_{m_j}(t)$, communication bandwidth B, transmitting power $p_{m_j}(t)$ and the channel power gain $h_{n_i}^{m_j}(t)$. Hence, the total communication latency can be represents

$$T_{comm}(t) = \sum_{i=1}^{N} \sum_{j=i}^{M} \frac{\delta_{n_i}^{m_j}(t) \cdot J_{m_j}(t)}{q_{n_i}^{m_j}(t)}, \tag{17}$$

$$q_{n_i}^{m_j}(t) = B \cdot \log_2\left(1 + \frac{p_{m_j}(t) \cdot h_{n_i}^{m_j}(t)}{\sigma^2}\right), \tag{18}$$

where σ is the Gaussian noise power.

The computation latency on mobile edge servers is related to the computation capability of mobile edge servers and the loads of received tasks, and thus the total computation latency can be represented as

$$T_{comp}(t) = \sum_{i=1}^{N} \frac{\mu \cdot A_{n_i}(t)}{f_{n_i}(t)} = \sum_{i=1}^{N} \frac{\tau \cdot A_{n_i}(t)}{L_{n_i}(t)}, \tag{19}$$

where $L_{n_i}(t)$ and $A_{n_i}(t)$ represents the completed tasks and received tasks of mobile edge server n_i at time slot t, respectively, $f_{n_i}(t)$ is the occupied computation resource of mobile edge server n_i at time slot t, μ is the transforming parameter, indicating the relationship between task loads and the needed computation resource, and τ is the length of time slot. Hence, the total latency of UAVs-assisted MEC can be obtained by

$$T_{total}(t) = T_{comm}(t) + T_{comp}(t). \tag{20}$$

3.4.3. System cost model

In this paper, we conduct a system cost model with considering both the energy consumption and task latency, which can be represented as

$$P(t) = \theta \cdot \frac{E'_{total}(t)}{E_e} + (1-\theta) \cdot \frac{T_{total}(t)}{T_e}, \tag{21}$$

where θ is a weight factor to adjust the proportion of energy consumption and latency considered in system cost and belongs to $[0, 1]$, and E_e and T_e is the ideal energy consumption and latency in UAVs-assisted MEC. The ideal energy consumption only considers the energy consumption for UAV kinetic and the ideal latency considers the computation latency to complete computation tasks with maximum computation capability and corresponding communication latency in mobile edge servers. Hence, the ideal energy consumption and latency can be obtained by

$$E_e = E\{E_{cki}(t)\}, \tag{22}$$

$$T_e = E\{\tau \cdot \frac{\sum_{i=1}^{N} A_{n_i}}{\sum_{i=1}^{N} L_{n_i}(t)} + \frac{\sum_{i=1}^{N} A_{n_i}}{\sum_{i=1}^{N}\sum_{j=1}^{M} q_{n_i}^{m_j}(t)}\}. \tag{23}$$

Note that, the system cost model in our paper also can be considered as an system cost rate model, indicating the cost rate of the actual cost to the ideal cost.

3.4.4. Problem formalization

As the location of mobile edge servers is not fixed and the computing capacity of the servers carried by the UAVs is different, the system cost of the same computing task is different on different servers. Hence, computation offloading and resource allocation schemes should be considered to take full advantage of resources in UAVs-assisted MEC. Actually, it is not feasible to simply offloading tasks and allocate resources according to the cost. For instance, if a mobile edge server can process offloaded tasks at a lower cost, more tasks will be offloaded to this mobile edge server to be processed, cause the mobile edge server to receive a large number of tasks and appearing large queue backlog, thereby affecting the quality of service. Therefore, in our paper, a dynamic resource allocation scheme is considered, in which the objective is to minimize the whole system cost of UAVs-assisted MEC over a long period of time on the basis of guaranteeing the stability of the task queue in mobile edge servers. With considering the random arrival of offloaded tasks, our dynamic resource allocation scheme can be formalized as

$$\min \left\{ \lim_{T\to+\infty} \frac{1}{T} \cdot \sum_{t=0}^{T-1} E\{P(t)\} \right\} \tag{24}$$

$S.T.$

$$\overline{Q} = \lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T-1}\sum_{i=0}^{N} E\{Q_{n_i}(t)\} < +\infty, \tag{25}$$

$$\forall n_i \in [1, N], \qquad f_{n_i} \le F_{n_i}, \tag{26}$$

$$\forall n_i \in [1, N], \qquad Q_{n_i} \le Q_{n_i}^{max}, \tag{27}$$

$$\forall n_i \in [1, N], \qquad \sum_{j=1}^{M} \delta_{n_i}^{m_j} = 1, \tag{28}$$

$$\forall n_i \in [1, N], \qquad \sum_{t=0}^{T-1} \left(E_{n_i}^{comm}(t) + E_{n_i}^{comp}(t) + E_{n_i}^{fly}(t) + E_{n_i}^{hold} \right) \le E_{n_i}^{Max} \tag{29}$$

$$\sum_{i=1}^{N} \delta_{n_i}^{m_j} \cdot \left(\frac{\tau \cdot \left(A_{n_i}(t) + Q_{n_i}(t)\right)}{f_{n_i}(t)} + \frac{J_{m_j}}{q_{n_i}^{m_j}} \right) \le TD_{m_j}. \tag{30}$$

where TD_{m_j} is the maximum allowed latency of offloaded tasks of mobile device m_j, and other notations are defined in Table 1. Eq. (25) represents the stability constraints of the task queue in mobile edge servers. Eq. (26) ensures the allocated computation resource is less than the total computation resource a mobile edge server has. Eq. (27) represents the backlog constraints of the task queue in mobile edge servers. When the maximum backlog of the task queue is achieved, the large latency will be generated and a large number of tasks will be lost. Eq. (28) represents the full offloading constraints of the task, ensuring that each task only can be offloaded to one mobile edge server. Eq. (29) ensures that at each time slot, the total energy consumption of a UAV (including the communication and computation energy consumed by the tasks offloaded, as well as the UAVs kinetic) must be not larger than its current residual power energy in the battery. Eq. (30) ensures all offloaded tasks can be completed in their allowed latency.

4. The Lyapunov based dynamic resource allocation (LDRA)

To solve the above formalized optimization problem, the Lyapunov function is introduced in our paper to control the dynamic system and transform the time-dependent long-term optimization problem to a time-independent optimization problem that only requires the information of the current time slot. Then, a greedy-based matching parameter determination algorithm is proposed to solve the optimization problem and find the optimal matching parameter between UAVs and mobile devices. Finally, the stability of our LDRA scheme is analyzed.

Note that, our LDRA model mainly focuses on several emergency scenarios. For example, in the traffic peak hours, the communication and computation tasks will be increased sharply around the congested roads, and UAVs are used as temporary mobile edge servers to be added in MEC to process tasks from the congested roads. In most of these emergency scenarios, computation-complexity applications are usually not included. When our model is used to process computation-complexity applications, the MEC-enabled base stations must be involved in our model. Additionally, due to fixed edge servers which can be considered as MEC-enabled base stations can be reliably powered and the objective of our model is to investigate the utility of mobile edge servers (i.e., UAVs) for offloaded computation tasks, in our model only UAVs serving as mobile edge servers are considered and fixed edge servers are not considered in our model. Note that, due to the motionless, actually fixed edge servers can also be easy to be merged in our model to process computation-complexity applications although adding several powerful MEC-enabled base stations in our LDRA model (Eqs. (24)∼(30)) to participate in the process of offloaded computation tasks.

4.1. Lyapunov optimization framework

To measure the sum of queue backlogs (congestion) in the UAVs-assisted MEC, a quadratic Lyapunov function [25], denoted as $Lya(t)$, is defined and can be represented as

$$Lya(t) = \frac{1}{2} \cdot \sum_{i=1}^{N} \left(Q_{n_i}(t)\right)^2. \tag{31}$$

Eq. (31) is known as the quadratic Lyapunov function of queue states. The smaller the quadratic Lyapunov function $Lya(t)$, the smaller the queue backlogs. If and only if all task queues are empty (i.e., $Q_{n_i}(t) = 0, (\forall n_i \in [1, N])$), the quadratic Lyapunov function $Lya(t)$ is equal to zero. Otherwise, the quadratic Lyapunov function $Lya(t)$ is always larger than zero.

Here, the Lyapunov drift function [25] at time slot t also is considered in our scheme and can be represented as

$$\Delta Lya(t) = E\{Lya(t+1) - Lya(t)\}. \tag{32}$$

Eq. (32) represents the variations of task backlogs in two adjacent time slots. Based on Eq. (32), our optimization problem formalized in Eq. (24) can be decomposed as ensuring constraints on queue stability in each time slot. To minimize the system cost under the premise of queue stability, the Lyapunov drift and penalty function [25] is defined as

$$LDP(t) = \Delta Lya(t) + V \cdot P(t), \tag{33}$$

where V is a non-negative control parameter and $P(t)$ is the system cost defined in Eq. (21).

Eq. (33) represents that without violating the long-term constraints, the system cost minimization is considered as the optimal objective, which not only can ensure the queue stability but also minimize the system cost. Through adjusting the control parameter V, the balance between system cost and task backlog can be achieved.

Obviously, at each time slot, the number of completed tasks in a mobile edge server is always greater than or equal to zero, and thus the following inequations can be obtained,

$$\max\{Q_{n_i}(t) - L_{n_i}(t), 0\} \le Q_{n_i}(t), \tag{34}$$

$$\left(\max\{Q_{n_i}(t) - L_{n_i}(t), 0\}\right)^2 \le \left(Q_{n_i}(t) - L_{n_i}(t)\right)^2, \tag{35}$$

where $Q_{n_i}(t)$ and $L_{n_i}(t)$ is the number of total tasks and completed tasks at time slot t in the task queue of mobile edge server n_i, respectively.

With considering Eq. (35), Eq. (5) can be updated as

$$\begin{aligned}\left(Q_{n_i}(t+1)\right)^2 &= \left(\max\{Q_{n_i}(t) - L_{n_i}(t), 0\}\right)^2 + \left(A_{n_i}(t)\right)^2 \\ &\quad + 2\cdot\max\{Q_{n_i}(t) - L_{n_i}(t), 0\}\cdot A_{n_i}(t), \\ &\le \left(Q_{n_i}(t) - L_{n_i}(t)\right)^2 + \left(A_{n_i}(t)\right)^2 \\ &\quad + 2\cdot Q_{n_i}(t)\cdot A_{n_i}(t), \\ &= \left(Q_{n_i}(t)\right)^2 + \left(A_{n_i}(t)\right)^2 + \left(L_{n_i}(t)\right)^2 \\ &\quad - 2\cdot Q_{n_i}(t)\cdot\left(L_{n_i}(t) - A_{n_i}(t)\right),\end{aligned} \tag{36}$$

where $A_{n_i}(t)$ is the number of received tasks of mobile edge server n_i at time slot t.

With considering N mobile edge servers (i.e., UAVs) in the UAVs-assisted MEC, based on Eq. (36), the Lyapunov drift function at time slot t (i.e., Eq. (32)) can be updated as

$$\begin{aligned}\Delta Lya(t) &= E\{\sum_{i=1}^{N}\frac{1}{2}\cdot((Q_{n_i}(t+1))^2 - (Q_{n_i}(t))^2)\} \\ &\le E\{\sum_{i=1}^{N}\frac{\left(A_{n_i}(t)\right)^2 + \left(L_{n_i}(t)\right)^2}{2}\} \\ &\quad - E\{\sum_{i=1}^{N}Q_{n_i}(t)\cdot\left(L_{n_i}(t) - A_{n_i}(t)\right)\} \\ &\le \sum_{i=1}^{N}\frac{\left(A_{n_i}^{max}(t)\right)^2 + \left(L_{n_i}^{max}(t)\right)^2}{2} \\ &\quad - E\{\sum_{i=1}^{N}Q_{n_i}(t)\cdot\left(L_{n_i}(t) - A_{n_i}(t)\right)\}.\end{aligned} \tag{37}$$

In the meanwhile, the Lyapunov drift and penalty function at time slot t (i.e., Eq. (33)) can be updated as

$$\begin{aligned}LDP(t) &\le \sum_{i=1}^{N}\frac{\left(A_{n_i}^{max}\right)^2 + \left(L_{n_i}^{max}\right)^2}{2} \\ &\quad + E\{\sum_{i=1}^{N}\left(V\cdot P(t) + Q_{n_i}(t)\cdot A_{n_i}(t)\right)\} \\ &\quad - E\{\sum_{i=1}^{N}Q_{n_i}(t)\cdot L_{n_i}(t)\}.\end{aligned} \tag{38}$$

Note that the objective of our scheme is to minimize the system cost over a long period of time on the basis of guaranteeing the stability of task queues in mobile edge servers. Hence, based on the Lyapunov optimization framework mentioned above, the optimization problem in our scheme can be transformed to find the optimal matching parameter $\delta_{n_i}^{m_j}(t)$ at each time slot to minimize the whole system cost, i.e., minimizing the $E\{\sum_{i=1}^{N}\left(V\cdot P(t) + Q_{n_i}(t)\cdot A_{n_i}(t)\right)\}$ in Eq. (38).

Therefore, based on the Lyapunov optimization framework mentioned above, the optimization problem of our dynamic resource allocation scheme (i.e., Eqs. (24)~(28)) can be updated as

$$\min\left\{E\left\{\left(V\cdot P(t) + \sum_{i=1}^{N}Q_{n_i}(t)\cdot A_{n_i}(t)\right)\right\}\right\} \tag{39}$$

$S.T.$ (26), (27), (28).

Based on Eqs. (2), (9), (11), (12), (16), (17), (19) and (21), we can have Eq. (40).

$$\begin{aligned}E\{V\cdot P(t)\} &= E\{V\cdot\theta\cdot\frac{E'_{total}(t)}{E_e} + V\cdot(1-\theta)\cdot\frac{T_{total}(t)}{T_e}\} \\ &= \frac{V\cdot\theta}{E_e}\cdot E\left\{\sum_{i=1}^{N}\left(\omega\cdot\kappa\cdot\tau\cdot(f_{n_i}(t))^3 + \omega\cdot p_{n_i}\right.\right. \\ &\quad \cdot\left(\sum_{j=i}^{M}J_{m_j}(t)\cdot\delta_{n_i}^{m_j}(t)\right) \\ &\quad \left.\left. + \frac{c\cdot\|v_{n_i}(t)\|^2}{\tau} + E_{n_i}^{hold}\right)\right\} \\ &\quad + \frac{V\cdot(1-\theta)}{E_e}\cdot E\left\{\sum_{i=1}^{N}\sum_{j=i}^{M}\frac{\delta_{n_i}^{m_j}(t)\cdot J_{m_j}(t)}{q_{n_i}^{m_j}(t)}\right. \\ &\quad \left. + \sum_{i=1}^{N}\sum_{j=i}^{M}\frac{\tau\cdot J_{m_j}(t)\cdot\delta_{n_i}^{m_j}(t)}{L_{n_i}(t)}\right\}\end{aligned} \tag{40}$$

$$\begin{aligned}\boldsymbol{\delta}(t) &= \arg\min_{\{\delta_{n_i}^{m_j}\}} E\left\{\sum_{i=1}^{N}\sum_{j=i}^{M}J_{m_j}(t)\cdot\delta_{n_i}^{m_j}(t)\cdot\left(Q_{n_i}(t) + \frac{V\cdot\theta}{E_e}\cdot\omega\cdot p_{n_i}\right.\right. \\ &\quad \left.\left. + \frac{V\cdot(1-\theta)}{E_e\cdot q_{n_i}^{m_j}(t)} + \frac{\tau\cdot V\cdot(1-\theta)}{E_e\cdot L_{n_i}(t)}\right)\right\} \\ &= \arg\min_{\{\delta_{n_i}^{m_j}\}} E\left\{\sum_{i=1}^{N}\sum_{j=i}^{M}\delta_{n_i}^{m_j}(t)\cdot C_{n_i}^{m_j}(t)\right\}\end{aligned} \tag{41}$$

$$C_{n_i}^{m_j}(t) = J_{m_j}(t)\cdot\left(Q_{n_i}(t) + \frac{V\cdot\theta}{E_e}\cdot\omega\cdot p_{n_i} + \frac{V\cdot(1-\theta)}{E_e\cdot q_{n_i}^{m_j}(t)} + \frac{\tau\cdot V\cdot(1-\theta)}{E_e\cdot L_{n_i}(t)}\right) \tag{42}$$

Due to being considered as a constant, the energy consumption for UAVs holding in air, denoted as $E_{n_i}^{hold}(t)$, can be ignored when we investigate the effect of matching parameter $\delta_{n_i}(t)$ on the system cost. Additionally, the number of tasks that needs to be completed at time slot t (i.e., the computation resource of mobile edge servers need to be occupied at time slot t) is dependent of matching parameter $\delta_{n_i}^{m_j}(t-1)$ and is independent of $\delta_{n_i}^{m_j}(t)$, because the offloaded tasks at time slot t will be executed as earlier as next time slot. Hence, the occupied computation resource of a mobile edge server at time slot t, denoted as $f_{n_i}(t)$, can be considered as a known parameter and thus $\omega\cdot\kappa\cdot\tau\cdot(f_{n_i}(t))^3$ in Eq. (40) will also not have effect on choosing the matching parameter $\delta_{n_i}^{m_j}(t)$ and can be ignored as well.

Therefore, according to Eq. (40), the optimization problem in Eq. (39) can be considered as choosing optimal matching parameter $\delta_{n_i}^{m_j}(t)$ with the objective of minimizing the total system cost, which can be represented as Eq. (41).

Hence, by introducing Lyapunov optimization in our LDRA scheme, the optimization problem of dynamic resource allocation with the objective of minimizing the total system cost over a long period of time can be transformed as an optimization problem of choosing optimal matching parameter $\delta_{n_i}^{m_j}(t)$ at each time slot, thereby reducing the complexity of solving the optimization problem of our LDRA scheme.

4.2. Greedy-based matching parameter determination

To solve the optimization problem in Eq. (41) and find the optimal matching parameter $\delta_{n_i}^{m_j}(t)$, a greedy-based matching parameter determination algorithm is proposed in this section, as shown in Algorithm 1. Note that, as mentioned in Section 2, we assume stable LoS links are existed between mobile devices and mobile edge servers, and once the

Algorithm 1 Greedy-based Matching Parameter Determination Algorithm

Input: Control parameter V, weight factor θ, channel power gain β_0, position of mobile devices and mobile edge servers $(x(t), y(t))$, the computation capability of mobile edge servers F_{n_i}, the maximum length of task queue $Q^{max}_{n_i}$.

Output: matching parameter $\delta^{m_j}_{n_i}(t)$.

1: $Q_{n_i}(t) = 0, A_{n_i}(t) = 0, L_{n_i}(t) = 0$;
2: **for** t=0, t++, t < T **do**
3: $\delta(t) \leftarrow [-1]_{N\times M}$;
4: **for** $j \in [1, M]$ **do**
5: Getting $(x_{m_j}(t), y_{m_j}(t))$ and $J_{m_j}(t)$;
6: **end for**
7: **for** $i \in [1, N]$ **do**
8: Getting $(x_{n_i}(t), y_{n_i}(t))$;
9: **end for**
10: Obtaining $q^{m_j}_{n_i}$ based on Eqs. (4) and (18);
11: **while** $\{C^{m_j}_{n_i}(t)\}_{N\times M} != NULL$ **do**
12: $C^{m^*_j}_{n^*_i} \leftarrow \min\{C^{m_j}_{n_i}(t)\}_{N\times M}$;
13: **if** $\left(Q_{n^*_i}(t) + \delta^{m^*_j}_{n^*_i}(t) \cdot J_{m^*_j}(t) \le Q^{max}_{n^*_i}\right) \&\& (\delta^{m^*_j}_{n^*_i}(t) != -1)$ **then**
14: $A_{n^*_i}(t) \leftarrow A_{n^*_i}(t) + J_{m^*_j}(t)$;
15: $\{C^{m_j}_{n_i}(t)\}_{N\times M} \leftarrow \{C^{m_j}_{n_i}(t)\}_{N\times M} - \{C^{m^*_j}_{n^*_i}(t)\}$;
16: $\delta^{m^*_j}_{n^*_i}(t) \leftarrow 1$;
17: **for** $n_i \in \{n_1, n_2, \cdots, n_N\} - \{n^*_i\}$ **do**
18: $\delta^{m^*_j}_{n_i}(t) \leftarrow 0$;
19: $\{C^{m_j}_{n_i}(t)\}_{N\times M} \leftarrow \{C^{m_j}_{n_i}(t)\}_{N\times M} - \{C^{m^*_j}_{n_i}(t)\}$;
20: **end for**
21: **else**
22: $\delta^{m^*_j}_{n^*_i}(t) \leftarrow 0$;
23: **end if**
24: **end while**
25: $P(t) \leftarrow \theta \cdot \frac{E'_{total}(t)}{E_e} + (1-\theta) \cdot \frac{T_{total}(t)}{T_e}$;
26: $Q_{n_i}(t+1) \leftarrow \max\{Q_{n_i}(t) - L_{n_i}(t), 0\} + A_{n_i}(t)$;
27: **end for**

mobile devices are matched to a mobile edge server, the mobile edge server cannot refuse to perform the offloaded task of this mobile device.

In Algorithm 1, at the beginning of each time slot t, the backlog of task queue and the available computation resource in mobile edge servers is determined. Additionally, based on the GPS devices, the positions of both mobile edge servers and mobile devices are determined and the channel power gain and transmission rate can be determined by Eqs. (4) and (18) (Line 1∼10 in Algorithm 1). Then, the greedy algorithm is introduced to determine the matching parameter δ_{n_i}. That is, the device-edge pairs, denoted as n^*_i and m^*_j, is selected so that the $C^{m^*_j}_{n^*_i}$ is smallest one in set $\{C^{m_j}_{n_i}(t)\}_{N\times M}$. If the length constraints of the task queue of mobile edge server n_i can be guaranteed, the matching parameter $\delta^{m^*_j}_{n^*_i}$ is set as 1, i.e., the tasks of mobile device m^*_j will be offloaded to mobile edge server n^*_i to be completed, and then all $C^{m^*_j}_{n_i}$ are removed out of the set $\{C^{m_j}_{n_i}(t)\}_{N\times M}$, due to the tasks of mobile device only can be offloaded to one mobile edge server. Through the iteration, all matching parameter $\boldsymbol{\delta(t)}$ can be determined (Line 11∼23 in Algorithm 1). Due to the smallest $C^{m^*_j}_{n^*_i}$ is selected in set $\{C^{m_j}_{n_i}(t)\}_{N\times M}$ at each iteration, according to Eq. (41), the determined matching parameter $\boldsymbol{\delta(t)}$ can be the optimal one leading to the minimum value of $E\left\{\sum_{i=1}^{N}\sum_{j=i}^{M}\delta^{m_j}_{n_i}(t)\cdot C^{m_j}_{n_i}(t)\right\}$. Finally, Based on the obtained optimal matching parameter $\boldsymbol{\delta(t)}$, the task offloading scheme can be obtained and the system cost can be determined by Eq. (21) and the task queue state can be updated by Eq. (5) (Line 25∼26 in Algorithm 1).

4.3. Stability analysis

In this section, the stability of our LDRA scheme is analyzed. According to Lyapunov stability theorem [25], when the system is stable, the Lyapunov drift and penalty function of the system is falling in a threshold range.

At each time slot, according to Eq. (38), with the task queue state being $Q_{n_i}(t)$, the Lyapunov drift and penalty function of our LDRA scheme can be falling a range, which can be represented as

$$E\{Lya(t) + V \cdot P(t)\} \le B + V \cdot p^* - \epsilon \cdot E\left\{\sum_{i=1}^{N} Q_{n_i}(t)\right\} \tag{43}$$

$$B = \sum_{i=1}^{N} \frac{\left(A^{max}_{n_i}\right)^2 + \left(L^{max}_{n_i}\right)^2}{2} \ge 0, \tag{44}$$

where V is the control parameter, p^* is average system cost, ϵ and B are determined as constants.

Hence, based on Lyapunov stability theorem, both the time average system cost and time average length of task queue in our LDRA scheme are stable, i.e., existing the upper limits, which can be represented as

$$\lim_{T\to+\infty} \frac{1}{T} \cdot \sum_{t=0}^{T-1} E\{P(t)\} \le p^* + \frac{B}{V} \tag{45}$$

$$\lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T-1}\sum_{i=0}^{N} E\{Q_{n_i}(t)\} \le \frac{B + V \cdot p^*}{\epsilon}. \tag{46}$$

Eqs. (43), (45) and (46) can demonstrate that, the Lyapunov drift and penalty function of our LDRA scheme can be falling a range of $[0, B + V \cdot p^* - \epsilon \cdot E\left\{\sum_{i=1}^{N} Q_{n_i}(t)\right\}]$, and the time average system cost is $O(\frac{1}{V})$ greater than the expected system cost and the complexity of time average length of task queue is $O(V)$. That means, the control parameter V can be effectively used to adjust the time average system cost and time average length of task queue, thereby achieving the balance of system cost and length of task queue in our LDRA scheme for UAVs-assisted MEC.

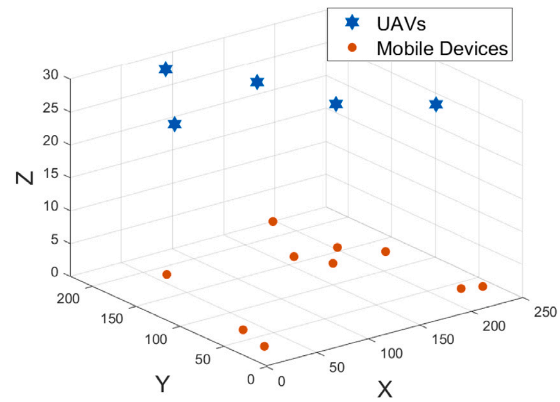


Fig. 4. The positions of UAVs and mobile devices.

5. Evaluations

5.1. Evaluation setup

In this section, we evaluate the performance of our LDRA in terms of system stability, system cost, and system utility in comparison with Randomized Scheduling Algorithm (RSA) and (JSQ) [26].

We carry out the performance evaluation based on a simple UAVs-assisted MEC model. Ten mobile devices and five UAVs (considered as mobile edge servers) are included in a range of 250 m × 250 m, and the flight altitude of all UAVs is fixed as 30 m. Fig. 4 shows the positions of mobile devices and UAVs, in which the blue stars are the positions of mobile edge servers and red dots are the positions of mobile devices. ’

In our evaluations, the weight of the UAV is 9.65 kg, and the maximum flying speed is 20 m/s and the energy consumption of the

Table 2
Parameters setting.

Parameter	Value	Parameter	Value
H	20 m	ω	18 000
β_0	−20 dB	κ	10^{-28}
$p_{n_i}(t)$	[80 mW, 120 mW]	σ^2	10^{-9} W
F_{n_i}	[6 $\cdot 10^6$, 9 $\cdot 10^6$]	E^{hold}	220 W/s

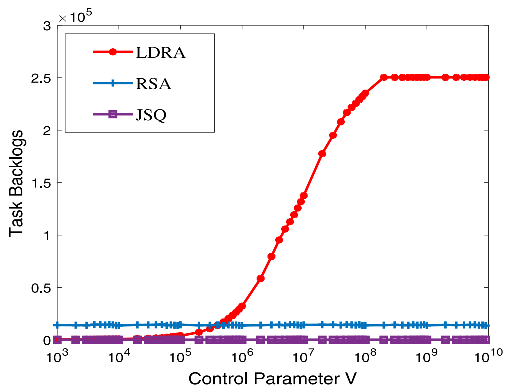


Fig. 5. Control parameter V vs. Task backlogs (without backlog constraints).

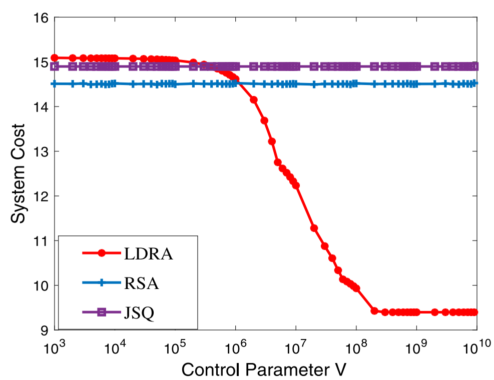


Fig. 6. Control parameter V vs. System costs (without backlog constraints).

UAV for holding in the air is 220 W/s [7]. The bandwidth of LoS link is 40 MHz and the channel power gain is −20 dB [27]. The parameters setting in our evaluations are shown in Table 2.

In the evaluations of system stability, the impact of control parameter V on the backlogs of task queues are evaluated, in which the number of tasks generated by mobile devices follows a Poisson distribution with an average value of 20, and the weight factor θ in Eq. (20) is set as 0.09. In the evaluations of system cost, the time average system cost and the task backlogs are evaluated. In the evaluations of system utility, the number of tasks completed over a long period of time is evaluated. All simulations in this paper are conducted using Matlab.

5.2. Evaluation results

5.2.1. The impact of control parameter V

Figs. 5 and 6 show the impact of control parameter V on the time average task backlogs and the time average system cost without backlog constraints, respectively. Due to the system stability is not considered in RSA and JSQ, the control parameter V cannot affect the performance of RSA and JSQ schemes. Hence, as the increase of control parameter V, the time average task backlogs and the time average system cost of RSA and JSQ schemes, shown in Figs. 5 and 6, can maintain constant. With the objective of maximizing the throughput, JSQ scheme can achieve the lowest task backlogs, but achieve greater system cost. In our LDRA scheme, when the control parameter V is less than 10^5, the time average system cost of our LDRA is less than 15.5 and the time average task backlogs is similar to that of JSQ scheme and is less than that of RSA scheme. As shown in Figs. 5 and 6, as the increase of control parameter V, the task backlogs of our LDRA scheme increases and will be eventually converged with accepted value and the system cost of our LDRA scheme will decrease and will eventually converge with the lowest value as well which is much less than the time average system cost of RSA and JSQ schemes.

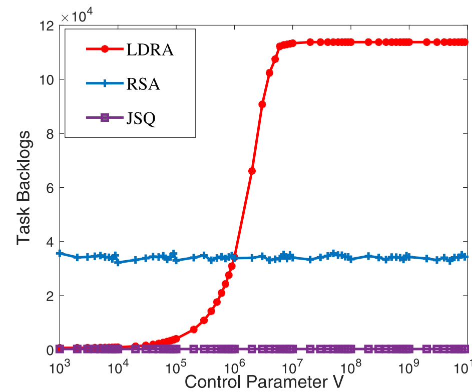


Fig. 7. Control parameter V vs. Task backlogs (with backlog constraints $Q^{max}_{n_i}$ = 30000).

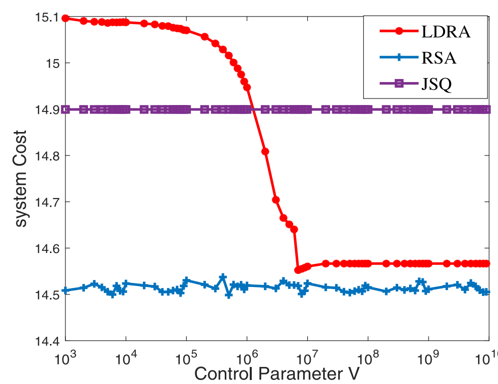


Fig. 8. Control parameter V vs. System costs (with backlog constraints $Q^{max}_{n_i}$ = 30000).

Figs. 7 and 8 show the impact of control parameter V on the time average task backlogs and the time average system cost with backlog constraints being 30 000 (i.e., $Q^{max}_{n_i} = 30000$), respectively. As shown in Figs. 7 and 8, when the control parameter V is larger than 10^6, both the time average task backlogs and the time average system cost of our LDRA can be converged, and the total task backlogs of our LDRA scheme in Fig. 7 is close to maximum backlogs (i.e., $12 \cdot 10^4$), which is much less than the total backlogs of our LDRA without backlog constraints in Fig. 5. Figs. 5–8 show that the control parameter V can effectively adjust and balance the time average task backlogs and the time average system cost of our LDRA scheme. Additionally, the performance boundaries of the time average task backlogs and the time average system cost of our LDRA scheme is also shown to verify the analysis in Section 4.3.

5.2.2. System stability

Fig. 9 shows the system stability of our LDRA scheme with different control parameters V, in which the number of tasks generated by mobile devices obeys the Poisson distribution with an expectation of 20. Fig. 10 shows the system stability of our LDRA scheme with different generation rates of tasks, in which the control parameter V is selected as $5 \cdot 10^5$ and the number of tasks generated by mobile devices

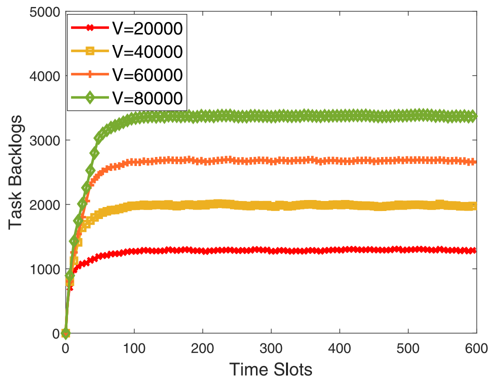


Fig. 9. System stability with different control parameters V.

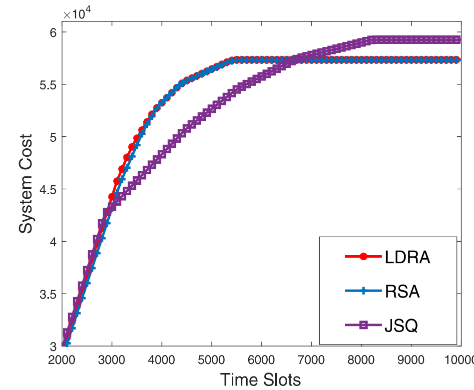


Fig. 11. System Cost over a long period of time.

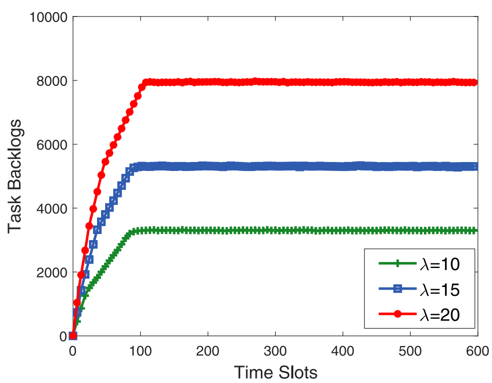


Fig. 10. System stability with different task loads λ.

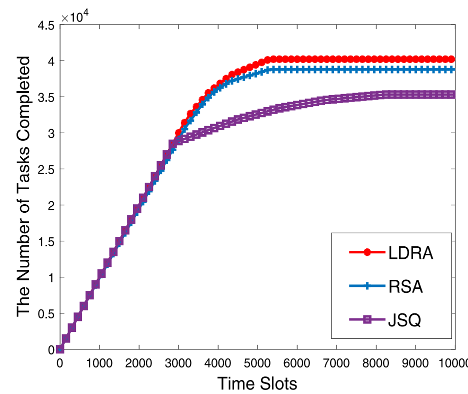


Fig. 12. The number of tasks completed over a long period of time.

obeys the Poisson distribution. As shown in Figs. 9 and 10, with the increase of time slots, the task backlogs increase rapidly at first and then will be converged to a constant. In addition, as the increase of control parameter V, the constant converged increases as well. In the meantime, as the increase of generation rate of tasks (i.e., λ in Poisson distribution), the constant converged increases and the difficulty of convergence increases. Figs. 9 and 10 show that our LDRA can achieve great system stability on task backlogs, i.e., the task backlogs can always be converged over a long period of time.

5.2.3. System utility

Fig. 11 shows the total system cost over a long period of time and Fig. 12 shows the number of tasks that are completed by UAVs-assisted MEC over a period of time in RSA, JSQ and our LDRA models. In this evaluation, the maximum energy stored in each UAV is set as 3000 kJ, and the UAVs will stop providing computation services when they are out of energy. As shown in Figs. 11 and 12, at the beginning, due to enough energy stored in UAVs, both the system cost and the number of tasks completed in all three models increases. That is, all offloaded tasks of mobile devices can be completed in mobile edge servers. As the time goes on, a number of UAVs will be out of energy and some offloaded tasks will be lose, resulting in slowing down the growth rate of system cost and the number of tasks completed and finally being converged. In particular, as shown in Fig. 12, the number of offloaded communication and computation tasks completed in our model is larger than that of existing models (i.e., RSA and JSQ), which means that our LDRA scheme can complete more offloaded tasks with lower energy consumption, thereby achieve greater utility and sustainability of the UAVs-assisted MEC in comparison with RSA and JSQ scheme.

6. Related works

Mobile edge computing (MEC) aims to provide extra computation, communication, and storage capabilities to assist in completing the tasks of mobile devices with low latency and energy consumption [28]. Some efforts focused on the framework of UAVs-assisted MEC have been developed. For example, Liao et al. [9] proposed a HOTSPOT framework of UAVs-assisted MEC to determine computation offloading schedules with considering the positions of UAVS. Tan et al. [29] proposed a UAV-aided edge/fog computing framework for social augmented reality. Zhang et al. [30] proposed a NOMA-based UAVs-assisted MEC framework to provide extra computation resource for mobile devices, which can reduce the computation offloading energy consumption through NOMA.

A number of efforts on computation offloading and resource allocation are developed with considering various constraints, such as device characteristics and status, network quality or interference, service status, resource status, and user preferences. For instance, Yu et al. [31] proposed a UAVs-assisted MEC system that can minimize the energy consumption of computation offloading with considering the position of UAVs and communication cost. Liu et al. [32] considered UAVs as the relay of the MEC and aimed to minimizing the energy consumption of UAVs through jointly optimize the transmission modes, positions and transmission power of UAVs. Zhang et al. [33] proposed a game-based computation offloading scheme with considering only one UAV and one base station in MEC, which can minimize the cost consisted by latency and energy consumption. Zhang et al. [34] considered the UAVs as the relay of MEC to reduce the task latency of end-devices. Zhang et al. [35] optimized the operation cost for UAVs-assisted MEC with considering computing offloading, spectrum resource allocation, computing resource allocation and UAV placement.

In addition, a number of efforts also focused on the mobility of devices and edge servers. One of the important issues is to ensure the teal-time response of offloaded tasks. For instance, Wang et al. [36] proposed a markov based task migration scheme that can achieve the optimal task migration via considering the positions of both mobile devices and edge servers. Guo et al. [7], Ji et al. [10], Yu et al. [37] considered the joint optimization of tasks offloading and UAV trajectories with the objective of minimizing the task latency. Qin et al. [38] proposed a multi-UAV-assisted and multi-access MEC model, which can minimize the total weighted energy consumption of UAVs and end-devices through jointly bit allocation, transmitting power, CPU frequency, bandwidth allocation, and UAV trajectory. Tun et al. [39] proposed a BSUM scheme to minimize the joint energy consumption of IoT devices and UAVs.

Some efforts also focused on the computation offloading on random task models. Due to the task information is difficult to be obtained, the stochastic optimization schemes are usually introduced to transform the complex problems into continuous optimization problems, thereby reducing the complexity [40–42]. For example, Chen et al. [41] proposed a radio and computation resources joint management scheme for multi-user MEC systems, which can minimize the latency on the basis of satisfying the power constraints. Through introducing the Lyapunov optimization, the optimization problem can be solved in a definite time interval. Qiao et al. [42] proposed a scheme to minimize the energy consumption of mobile devices and introduced Lyapunov optimization to determine the CPU cycle frequency and tasks randomly transferred in each time slot. In addition, some schemes also introduced Bellman function and Markov decision to deal with latency and energy consumption optimization. In addition, due to independent of prior information, Lyapunov optimization also has been considered in computation offloading with random task models to optimize the MEC system performance while ensuring the stability [43–47].

Unlike existing schemes, in this paper, we have developed a Lyapunov based resource allocation scheme for UAVs-assisted mobile edge computing, which considers multiple mobile devices and multiple mobile edge servers (i.e., a group of UAVs) in UAVs-assisted MEC and can effectively allocate the resource of mobile edge servers to the computation tasks of mobile devices with the objective of improving the sustainability of UAVs-assisted MEC in which a group of UAVs are collaboratively served as mobile edge servers. In addition, our LDRA scheme considers the random task generation model for mobile devices and can achieve dynamically resource allocation with great system stability, low system cost, as well as great system utility.

7. Conclusion

In this paper, we proposed a novel Lyapunov based dynamic resource allocation scheme for UAVs-assisted MEC, namely LDRA, which can effectively allocate resources in mobile edge servers to offloaded computation tasks of mobile devices through introducing Lyapunov optimization. Our LDRA scheme formalized a tasks queue model and system cost model with considering both the energy consumption of edge servers and computation latency of offloaded tasks. Via introducing Lyapunov optimization, our LDRA scheme was transformed as a time-independent optimization problem, and then a greedy-based matching parameter determination scheme is proposed to allocate resources in mobile edge servers to offloaded tasks of mobile devices. Through a combination of both theoretical analysis and simulation experiments, our data shows that our LDRA scheme can achieve great system stability, low system cost, as well as great system utility in comparison with existing schemes.

CRediT authorship contribution statement

Jie Lin: Conceptualization, Methodology, Writing – original draft. **Lin Huang:** Software, Data curation, Formal analysis. **Hanlin Zhang:** Writing – review & editing. **Xinyu Yang:** Visualization, Investigation, Supervision. **Peng Zhao:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

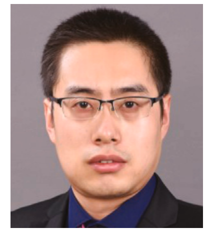
Acknowledgments

This work was partly supported by K. C. Wong Education Foundation, by the Fundamental Research Funds for the Central Universities under Grant xzy012021082, by Natural Science Basic Research Plan in Shaanxi Province of China 2019JQ-124, 2020JQ-070. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the agencies.

References

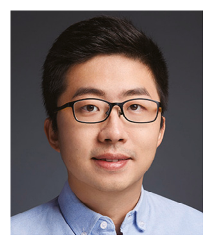
[1] Wei Yu, Fan Liang, Xiaofei He, William Grant Hatcher, Chao Lu, Jie Lin, Xinyu Yang, A survey on the edge computing for the internet of things, IEEE Access 6 (2018) 6900–6919.
[2] Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Mika Ylianttila, A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects, IEEE Commun. Surv. Tutor. 23 (2) (2021) 1160–1192.
[3] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, Wei Zhao, A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications, IEEE Internet Things J. 4 (5) (2017) 1125–1142.
[4] Davide Callegaro, Marco Levorato, Optimal edge computing for infrastructure-assisted UAV systems, IEEE Trans. Veh. Technol. 70 (2) (2021) 1782–1792.
[5] Yong Zeng, Rui Zhang, Teng Joon Lim, Wireless communications with unmanned aerial vehicles: opportunities and challenges, IEEE Commun. Mag. 54 (5) (2016) 36–42.
[6] Lav Gupta, Raj Jain, Gabor Vaszkun, Survey of important issues in UAV communication networks, IEEE Commun. Surv. Tutor. 18 (2) (2016) 1123–1152.
[7] Hongzhi Guo, Jiajia Liu, UAV-enhanced intelligent offloading for internet of things at the edge, IEEE Trans. Ind. Inf. 16 (4) (2020) 2737–2746.
[8] Cheng Zhan, Han Hu, Zhi Liu, Zhi Wang, Shiwen Mao, Multi-UAV-enabled mobile-edge computing for time-constrained IoT applications, IEEE Internet Things J. 8 (20) (2021) 15553–15567.
[9] Zhuofan Liao, Yinbao Ma, Jiawei Huang, Jianxin Wang, Jin Wang, HOTSPOT: A UAV-assisted dynamic mobility-aware offloading for mobile-edge computing in 3-D space, IEEE Internet Things J. 8 (13) (2021) 10940–10952.
[10] Jiequ Ji, Kun Zhu, Changyan Yi, Dusit Niyato, Energy consumption minimization in UAV-assisted mobile-edge computing systems: Joint resource allocation and trajectory design, IEEE Internet Things J. 8 (10) (2021) 8570–8584.
[11] Chao Sun, Wei Ni, Xin Wang, Joint computation offloading and trajectory planning for UAV-assisted edge computing, IEEE Trans. Wireless Commun. 20 (8) (2021) 5343–5358.
[12] Fuhui Zhou, Yongpeng Wu, Haijian Sun, Zheng Chu, Uav-enabled mobile edge computing: Offloading optimization and trajectory design, in: 2018 IEEE International Conference on Communications (ICC), 2018, pp. 1–6.
[13] Cheng Zhan, Han Hu, Xiufeng Sui, Zhi Liu, Dusit Niyato, Completion time and energy optimization in the UAV-enabled mobile-edge computing system, IEEE Internet Things J. 7 (8) (2020) 7808–7822.
[14] Xiaowen Cao, Jie Xu, Rui Zhang, Mobile edge computing for cellular-connected UAV: Computation offloading and trajectory optimization, in: 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2018, pp. 1–5.
[15] Bo Yang, Xuelin Cao, Chau Yuen, Lijun Qian, Offloading optimization in edge computing for deep-learning-enabled target tracking by internet of UAVs, IEEE Internet Things J. 8 (12) (2021) 9878–9893.
[16] Shichao Zhu, Lin Gui, Dongmei Zhao, Nan Cheng, Qi Zhang, Xiupu Lang, Learning-based computation offloading approaches in UAVs-assisted edge computing, IEEE Trans. Veh. Technol. 70 (1) (2021) 928–944.
[17] Yi Zhou, Cunhua Pan, Phee Lep Yeoh, Kezhi Wang, Maged Elkashlan, Branka Vucetic, Yonghui Li, Secure communications for UAV-enabled mobile edge computing systems, IEEE Trans. Commun. 68 (1) (2020) 376–388.
[18] Long Zhang, Zhen Zhao, Qiwu Wu, Hui Zhao, Haitao Xu, Xiaobo Wu, Energy-aware dynamic resource allocation in UAV assisted mobile edge computing over social internet of vehicles, IEEE Access 6 (2018) 56700–56715.
[19] X. Zhang, Y. Zhong, P. Liu, F. Zhou, Y. Wang, Resource allocation for a UAV-enabled mobile-edge computing system: Computation efficiency maximization, IEEE Access (2019).
[20] Zhenzhen Hu, Fanzi Zeng, Zhu Xiao, Bin Fu, Hongbo Jiang, Hongyang Chen, Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems, IEEE Trans. Netw. Sci. Eng. (2021) 1.

[21] Jiao Zhang, Li Zhou, Qi Tang, Edith C.-H. Ngai, Xiping Hu, Haitao Zhao, Jibo Wei, Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing, IEEE Internet Things J. 6 (2) (2019) 3688–3699.
[22] Tiankui Zhang, Yu Xu, Jonathan Loo, Dingcheng Yang, Lin Xiao, Joint computation and communication design for UAV-assisted mobile edge computing in IoT, IEEE Trans. Ind. Inf. 16 (8) (2020) 5505–5516.
[23] Fuhui Zhou, Yongpeng Wu, Rose Qingyang Hu, Yi Qian, Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems, IEEE J. Sel. Areas Commun. 36 (9) (2018) 1927–1941.
[24] T. Cover, J. Thomas, J. Wiley, Elements of Information Theory, Elements of information theory, 2003.
[25] Jian Li, Mugen Peng, Yuling Yu, Zhiguo Ding, Energy-efficient joint congestion control and resource optimization in heterogeneous cloud radio access networks, IEEE Trans. Veh. Technol. 65 (12) (2016) 9873–9887.
[26] Konstantinos Psychas, Javad Ghaderi, Randomized algorithms for scheduling multi-resource jobs in the cloud, IEEE/ACM Trans. Netw. 26 (5) (2018) 2202–2215.
[27] Yuan Liu, Ke Xiong, Qiang Ni, Pingyi Fan, Khaled Ben Letaief, UAV-assisted wireless powered cooperative mobile edge computing: Joint offloading, CPU control, and trajectory optimization, IEEE Internet Things J. 7 (4) (2020) 2777–2790.
[28] Tao Zheng, Jian Wan, Jilin Zhang, Congfeng Jiang, Gangyong Jia, A survey of computation offloading in edge computing, in: 2020 International Conference on Computer, Information and Telecommunication Systems (CITS), 2020, pp. 1–6.
[29] Zhenjie Tan, Hua Qu, Jihong Zhao, Shiyu Zhou, Wenjie Wang, UAV-aided edge/fog computing in smart IoT community for social augmented reality, IEEE Internet Things J. 7 (6) (2020) 4872–4884.
[30] Xiaochen Zhang, Jiao Zhang, Jun Xiong, Li Zhou, Jibo Wei, Energy-efficient multi-UAV-enabled multiaccess edge computing incorporating NOMA, IEEE Internet Things J. 7 (6) (2020) 5613–5627.
[31] Zhe Yu, Yanmin Gong, Shimin Gong, Yuanxiong Guo, Joint task offloading and resource allocation in UAV-enabled mobile edge computing, IEEE Internet Things J. 7 (4) (2020) 3147–3159.
[32] Zhanjun Liu, Xin Tan, Miaowen Wen, Shupeng Wang, Chengchao Liang, An energy-efficient selection mechanism of relay and edge computing in UAV-assisted cellular networks, IEEE Trans. Green Commun. Netw. 5 (3) (2021) 1306–1318.
[33] Kaiyuan Zhang, Xiaolin Gui, Dewang Ren, Defu Li, Energy–latency tradeoff for computation offloading in UAV-assisted multiaccess edge computing system, IEEE Internet Things J. 8 (8) (2021) 6709–6719.
[34] Liang Zhang, Nirwan Ansari, Latency-aware IoT service provisioning in UAV-aided mobile-edge computing networks, IEEE Internet Things J. 7 (10) (2020) 10573–10580.
[35] Liang Zhang, Nirwan Ansari, Optimizing the operation cost for UAV-aided mobile edge computing, IEEE Trans. Veh. Technol. 70 (6) (2021) 6085–6093.
[36] Shiqiang Wang, Rahul Urgaonkar, Murtaza Zafer, Ting He, Kevin Chan, Kin K. Leung, Dynamic service migration in mobile edge computing based on Markov decision process, IEEE/ACM Trans. Netw. 27 (3) (2019) 1272–1288.
[37] Ye Yu, Xiangyuan Bu, Kai Yang, Hongyuan Yang, Xiaozheng Gao, Zhu Han, Uav-aided low latency multi-access edge computing, IEEE Trans. Veh. Technol. 70 (5) (2021) 4955–4967.
[38] Xintong Qin, Zhengyu Song, Yuanyuan Hao, Xin Sun, Joint resource allocation and trajectory optimization for multi-UAV-assisted multi-access mobile edge computing, IEEE Wirel. Commun. Lett. 10 (7) (2021) 1400–1404.
[39] Yan Kyaw Tun, Yu Min Park, Nguyen H. Tran, Walid Saad, Shashi Raj Pandey, Choong Seon Hong, Energy-efficient resource management in UAV-assisted mobile edge computing, IEEE Commun. Lett. 25 (1) (2021) 249–253.
[40] Yinglei Teng, Mengting Liu, F. Richard Yu, Victor C.M. Leung, Mei Song, Yong Zhang, Resource allocation for ultra-dense networks: A survey, some research issues and challenges, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2134–2168.
[41] Lixing Chen, Sheng Zhou, Jie Xu, Energy efficient mobile edge computing in dense cellular networks, in: 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–6.
[42] Ying Qiao, Deyu Zhang, Ju Ren, Yaoxue Zhang, Stochastic resource management for mobile edge computing in 5g networks, in: 2018 IEEE International Conference on Communication Systems (ICCS), 2018, pp. 378–383.
[43] Xingqiu He, Sheng Wang, Peer offloading in mobile-edge computing with worst case response time guarantees, IEEE Internet Things J. 8 (4) (2021) 2722–2735.
[44] Amine Abouaomar, Soumaya Cherkaoui, Zoubeir Mlika, Abdellatif Kobbane, Resource provisioning in edge computing for latency-sensitive applications, IEEE Internet Things J. 8 (14) (2021) 11088–11099.
[45] Kai Peng, Jiangtian Nie, Neeraj Kumar, Chao Cai, Jiawen Kang, Zehui Xiong, Yang Zhang, Joint optimization of service chain caching and task offloading in mobile edge computing, Appl. Soft Comput. 103 (2021) 107142.
[46] Deyu Zhang, Long Tan, Ju Ren, Mohamad Khattar Awad, Shan Zhang, Yaoxue Zhang, Peng-Jun Wan, Near-optimal and truthful online auction for computation offloading in green edge-computing systems, IEEE Trans. Mob. Comput. 19 (4) (2020) 880–893.
[47] Chen-Feng Liu, Mehdi Bennis, Mérouane Debbah, H. Vincent Poor, Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing, IEEE Trans. Commun. 67 (6) (2019) 4132–4150.

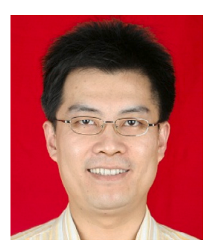


Dr. Jie Lin is currently an Associate Professor in the School of Computer Science and Technology from Xi'an Jiaotong University. He received the B.S. and Ph.D. degrees in the Department of computer science and technology from Xi'an Jiaotong University in 2009, and 2013, respectively. His research interests include Edge Computing, Internet of Things, and Security. He is a recipient of a Best Paper Award from IEEE IPCCC 2016.

Lin Huang is currently pursuing the master degree at the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an. She received the B.S. degree in the Department of computer science and technology from Xi'an Jiaotong University in 2018. She current research interests include Edge computing and Internet of Things.



Dr. Hanlin Zhang received his B.S. degree in Software Engineering from Qingdao University in 2010. He received the M.S. degree in applied information technology and Ph.D. degree in information technology from Towson University, MD, US, in 2011, 2016, respectively. He is currently working at Qingdao University as an assistant professor in the College of Computer Science and Technology. His research interests include Cloud Computing Security, Blockchain Technology and Internet of Things Security.



Dr. Xinyu Yang received the Diploma in Computer Science and Technology from the Xi'an Jiaotong University of China in 2001 and the Bachelor, Master, and Ph.D. degrees from Xi'an Jiaotong University in 1995, 1997, and 2001. He is currently a professor with the School of Computer Science and Technology, Xi'an Jiaotong University. His research interests include Wireless Communication, Mobile Ad Hoc Networks, and Network Security.

Dr. Peng Zhao is currently an Associate Professor in the School of Computer Science and Technology from Xi'an Jiaotong University. He received the B.S. and Ph.D. degrees in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2007 and 2013,respectively. From 2013 to 2014, he was a Research Fellow with Nanyang Technological University, Singapore. He is currently an Assistant Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University. His current research interests include Edge Computing, Network Security, and Deep Learning.