Task-Aware Joint Computation Offloading for UAV-Enabled Mobile Edge Computing Systems

Junshi Hu(✉), Heli Zhang, Xi Li, and Hong Ji

Key Laboratory of Universal Wireless Communications, Ministry of Education,
Beijing University of Posts and Telecommunications,
Beijing, People's Republic of China
{hujunshi,zhangheli,lixi,jihong}@bupt.edu.cn

Abstract. With the emergence of diverse computation-intensive mobile applications (such as virtual reality), demands for data processing from users are rapidly increasing in mobile edge computing (MEC). However, existing mobile edge servers (MES) are susceptible to propagation delays and loss and fail to provide timely and efficient services. Facing this problem, we focus on applying unmanned aerial vehicles (UAVs) equipped with computing resources to provide mobile edge computing offload services for users. UAV as an MES can guarantee low propagation delay and high reliability due to its maneuverability and short-distance line-of-sight communications. In this paper, we study a joint computing offloading problem consideration of user equipments, ground base stations and aerial UAVs. The system provides two offloading methods. The first offloading method is the air-offloading, where a user equipment can offload computing tasks to UAV-enabled MEC servers. The second offloading method is ground-offloading, where a user equipment can offload computing tasks to existing MESs. The task-aware optimization offloading scheme is proposed and it selects local execution or an offloading method based on the latency and energy constraints. Simulation results show that our proposed offloading selection scheme outperforms benchmark schemes. The results demonstrate that the proposed schemes improve quality of service (QoS) and have low task block rate under latency and energy constraints.

Keywords: UAV · Offloading selection · Air-offloading · Ground-offloading · Latency · Energy · MEC

1 Introduction

With recent development of technology and the decline in manufacturing costs, unmanned aerial vehicles (UAVs) have received growing interests in a wide range of applications such as post-disaster estimation, cargo delivery, search and rescue, as well as aerial photography [1]. In future mobile communications, with the

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2020
Published by Springer Nature Switzerland AG 2020. All Rights Reserved
H. Gao et al. (Eds.): ChinaCom 2019, LNICST 312, pp. 102–116, 2020.
https://doi.org/10.1007/978-3-030-41114-5_9

emergence of diverse computation-intensive and latency-critical mobile applications (e.g. video calls, virtual reality, and online game pervasive), the existing mobile edge servers (MES) are difficult to provide satisfactory quality of experience (QoE). UAV-enabled mobile edge computing, which is close to smart mobile devices (SMDs) and possesses stronger and more reliable line-of-sight (LoS), is undoubtedly one of the most critical applications to address the above problem.

Due to the limited battery and low computation capability, it is challenging for mobile devices to execute computation-intensive and latency-sensitive applications [2]. Fortunately, the emergence of MEC technology is promising to address this problem. Mobile users can offload computing tasks to edge networks with enormous computing resources to reduce application latency and save energy. In UAV-enabled networks, UAVs as mobile edge servers are capable of providing timely and efficient services, they provide seamless wireless coverage and ubiquitous computing offload services owing to their high maneuverability and low cost deployment. Meanwhile, flexible connectivity reduces the traffic load at the fixed cloud servers [3]. Therefore, the UAV equipped with a MEC server offers promising advantages compared to the conventional ground cellular network with fixed BSs. The UAV equipped with an MEC server offers promising advantages compared to the conventional ground cellular network with fixed BSs.

Recently, UAV-Enabled MEC provide UEs with mobile edge offload services, which have received extensive attention on the academic community. The work in [4] proposed a deep reinforce learning (DRL)-based scheme to maximize the throughput of offloading tasks offloading with limited UAV energy, and optimized objective function formulated by a semi-Markov decision process (SMDP). The work in [5] proposed a novel penalty dual decomposition (PDD) based algorithm to minimize the sum of the maximum delay among all the mobile users by combining UAV flight Trajectory optimization. The authors use local computing and partial offloading scheme, which each user offloads some tasks to the UAV, and the rest of the tasks are executed locally. In [6], the authors studied an UAV-enabled wireless powered MEC system, and proposed the sequential convex optimization (SCA) techniques to the power minimization problem. The authors formulated a non-convex optimization equation according to constraints on energy harvesting causality and the number of calculated bits. In [7], the authors considered that an unmanned aerial vehicle as edge servers to provide data processing services to the Internet of things devices (IoTDs). The paper minimizes the energy consumption of UAVs subject to the quality service requirement of IoT and the limited computing resource at UAVs. However, the above research only considers UAV-enabled MEC, they ignore ground cellular network with fixed BSs and latency- and energy-awareness for different offload tasks, which will result in loss in UAV-enabled networks performance. Therefore, these aspects will be incorporated into our work of task-aware joint offloading scheme in an MEC system with air-offloading and ground-offloading.

In this paper, we propose a task-aware MEC based on a joint computation offloading scheme, which including air-offloading and ground-offloading. Users can select to offload different computing tasks to edge servers collocated at base stations or UAV-enabled mobile edge servers or local computing according to available computing and energy resources. The proposed scheme employs the greedy based algorithm to select the best execution location for the current task. Our optimization objective is to minimize the energy consumption of the UE and the UAV under the conditions that satisfy the different latency constraints of each task. Furthermore, the proposed scheme reduces the probability of task blocking and the average delay of handling computing tasks in the system. This scheme is designed for applications, such as images, videos and so on, which requires critical computational latency and computing resources. The obtained results show that our scheme improves quality of servers (QoS) of MEC and extends the operational lifetime of the UE and the UAV to some extent.

The remainder of our work is organized as follows. Section 2 gives the system model and proposed optimization problem. In Sect. 3, the solution to joint offload optimization is presented. Simulation results and discussions are given in Sect. 4. Finally, the conclusions are drawn in Sect. 5.

2 System Model and Problem Formulation

In this section, we introduce the system model and formulate the joint computation offloading optimization problem.

2.1 System Model

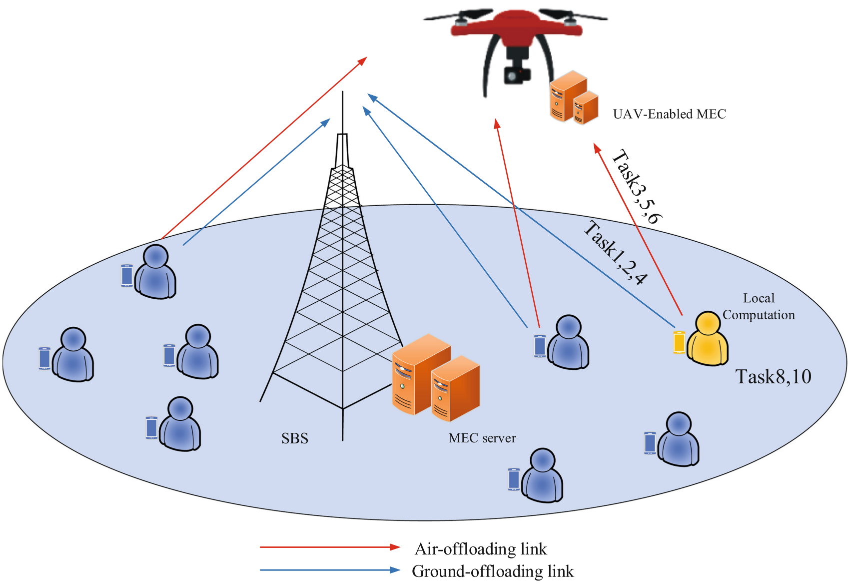


Fig. 1. The system model for task-aware joint computing offloading

As shown in Fig. 1, a joint computing offloading system is considered, which consists of the UAV-enabled MEC system and the ground MEC system. The UAV equipped with MEC servers or ground station connected to cloud servers provide computing offload services for M users. In this paper, the binary offloading paradigm is applied [8]. Each user can select to offload computing tasks to the UAV-enabled MEC server through air offloading or to ground base stations through ground offloading or execute locally. We apply the orthogonal frequency division multiple access (OFDMA) protocol to ensure that all users can offload their tasks to the UAV or the ground base station. Each UE is allowed to occupy one of sub-channels and the sub-channels are not reused, therefore, co-channel interference is ignored. Similar to [9–11], the energy consumption and delay for transmitting the computed results of the UAV and the ground base station are ignored. This paper considers a multitasking joint offloading problem for one user. For one UE, there are multiple different application tasks generated, task m is described by three terms d_m, c_m, t_m^{max}, where d_m is total size of input data of a task in bits, which includes program codes, input files, etc., c_m is the number of CPU cycles required to computing the task offloaded, and t_m^{max} is maximum allowable latency of a task.

2.2 Local Computing

If the task is executed at the UE locally, the execution time is denoted as:

$$t_m^L = \frac{c_m}{f^L}, \tag{1}$$

where f^L is local computing speed of UE in number of CPU cycles per second, and the local energy consumption is denoted as:

$$E_m^L = P_m^L t_m^L, \tag{2}$$

where P_m^L is local power consumption for the CPU to execute tasks.

2.3 Ground Mobile Edge Computing

For ground offloading, the goal of this paper is to minimize the energy consumption of mobile devices within the QoS latency time, so we ignore the energy consumption of the SBS. Energy consumption of ground offloading is the energy consumed at UE for offloading the task to the SBS through the wireless channel. Energy consumption of transmitting task m to the SBS is given by

$$E_m^{T_{SBS}} = P_m^T t_m^{UE-SBS}, \tag{3}$$

where P_m^T is data transmission power of the UE, and t_m^{UE-SBS} represents the transmission time of the wireless link, which is denoted as

$$t_m^{UE-SBS} = \frac{d_m}{R_{UE-SBS}}, \tag{4}$$

where R_{UE-SBS} is the data transmission rate of the UE to the SBS, and it can be calculated as

$$R_{UE-SBS} = W_{UE-SBS} \log_2(1 + \frac{P_m^T G_{UE-SBS}}{\sigma^2}), \tag{5}$$

where W_{UE-SBS} denotes the transmission sub-channel of the UE to the SBS, G_{UE-SBS} denotes the channel gain between the UE and the SBS, and σ^2 denotes the noise power at the SBS receiver. The time of the SBS executing task m is expressed as

$$t_m^{SBS} = \frac{c_m}{f^{SBS}}, \tag{6}$$

where f^{SBS} denotes SBS computing speed in number of CPU cycles per second.

2.4 UAV-Enabled Mobile Edge Computing

Without loss of generality, we adopt a three-dimensional (3D) Euclidean coordinate. Each user is fixed at the ground and the location is represented as l, where $l = [x,y]$, the UAV flies at a fixed altitude A_{UAV} during the period when the UAV communicates with the user and the horizontal plane coordinate of the UAV is $\hat{l} = [x_{UAV}, y_{UAV}]$. A LoS model is applied between the UAV and the user. During the finite time, the channel is unchanged. The channel power gain is given as

$$H_{UE-UAV} = \beta_0 D_{UE-UAV}^{-2} = \frac{\beta_0}{A_{UAV}^2 + ||\hat{l} - l||^2}, \tag{7}$$

where β_0 is the channel power gain at a reference distance $d_0 = 1m$. Then, the transmission rate of the UE to the UAV is expressed as

$$R_{UE-UAV} = W_{UE-UAV} \log_2(1 + \frac{P_m^T H_{UE-UAV}}{\hat{\sigma}^2}), \tag{8}$$

where W_{UE-UAV} denotes the transmission sub-channel of the UE to the UAV, H_{UE-UAV} denotes the channel gain between the UE and the UAV, and $\hat{\sigma}^2$ denotes the noise power at the UAV receiver. The time and the energy consumption for offloading the task m of the UE to the UAV are given by

$$t_m^{UE-UAV} = \frac{d_m}{R_{UE-UAV}} \tag{9}$$

and

$$E_m^{T_{UAV}} = P_m^T t_m^{UE-UAV}. \tag{10}$$

The time and energy consumption of the UAV executing task m are expressed as

$$t_m^{UAV} = \frac{c_m}{f^{UAV}}, \tag{11}$$

where f^{UAV} denotes UAV computing speed in number of CPU cycles per second and

$$E_m^{E_{UAV}} = P_m^{UAV} t_m^{UAV}, \tag{12}$$

where P_m^{UAV} is the UAV power consumption for the CPU to execute tasks. In this paper, for deployed UAVs, we ignore the energy consumption at the UAV due to flight.

2.5 Problem Formulation

In this paper, we focus on the energy consumption of UEs and UAVs in the joint offloading selection scheme. Therefore, the optimal problem can be formulated as minimizing the energy consumption of UEs and UAVs with satisfying the delay tolerance for given tasks, and the available resource constraints. For the $task_m$ which is described by three terms (d_m, c_m, t_m^{max}), optimization equation is given by

$$\begin{aligned}
&\min_{\omega,\eta}(\omega E_m^{loc} + \eta E_m^{uav}) \\
\texttt{s.t. } & C1 : t_m \leq t_m^{max}, \quad m = 1, 2, ..., M \\
& C2 : \omega > \eta \quad \forall task_m, m = 1, 2, ..., M \\
& C3 : \omega + \eta = 1 \quad \forall task_m, m = 1, 2, ..., M \\
& C4 : 0 < \omega < 1 \quad \forall task_m, m = 1, 2, ..., M \\
& C5 : 0 < \eta < 1 \quad \forall task_m, m = 1, 2, ..., M \\
& C6 : E_m^{loc_{re}} \geq E_{th}^{loc} \quad \forall task_m, m = 1, 2, ..., M \\
& C7 : E_m^{uav_{re}} \geq E_{th}^{uav} \quad \forall task_m, m = 1, 2, ..., M
\end{aligned} \tag{13}$$

Among all the constraints, $C1$ is to enforce the hard deadline of the task tolerance delay, $C2$, $C3$, $C4$, $C5$ are to ensure that UE energy optimization priority is higher than UAV energy optimization, $C6$ is to ensure that the UE has energy remaining after executing the $task_m$, $C7$ is to ensure that the UAV has energy remaining after executing the $task_m$.

3 Solution to the Joint Offloading Optimization Problem

In this section, we first decompose the optimization in (13) into the offloading selection subproblem, which can be converted to an optimal matching problem based on the latency and energy limits of each specific task and solved using the greedy based offload selection algorithm.

For a specific task, the UE broadcasts a request message to all surrounding UAVs and SBSs. As shown in Table 1, the request message contains two main fields, respectively, the location field and the task information field. The location field indicates the current location of the UE. The task information field contains three terms of the offloaded task d_m, c_m, t_m^{max}. As shown in Table 2, all nearby UAVs and SBSs receive the discovery message broadcasted by the UE, and send a response message to the UE. The response message contains three main fields, respectively, the location field, offloading decision, and execution details. The location field in respond message indicates the location of the current task,

Table 1. Request message

Request message		
Identification	Task specification	
Location	Type	Size

Table 2. Response message

Response message			
Identification	Offloading decision	Execution details	
Location	One/Zero	Execution time	Energy consumption

offloading decision indicates whether the task can be offloaded according to the energy limit of the UAV or SBS and the delay limit of the task, execution details contain time and energy consumption required to execute the task offloaded.

For the offloading selection subproblem, a specific task can be executed in three location including local, the UAV base station and the SBS in this paper. The UE reasonably selects the location where the task is offloaded based on the response message. In (13), The energy consumption of task offloading to the local, the UAV, and the SBS is described as follows:

$$\begin{aligned} &\hat{E}_{loc} = E_m^L \\ &\hat{E}_{uav} = \omega E_m^{T_{UAV}} + \eta E_m^{E_{UAV}} \\ &\hat{E}_{sbs} = E_m^{T_{SBS}} \\ &\qquad \texttt{s.t.} \;\; C2, C3, C4, C5 \end{aligned} \tag{14}$$

(a) *The Task Is Executed Locally:* Upon generating a task, the UE broadcasts the request message to seek a suitable execution location based on task energy consumption and QoS latency requirement. The UE executes offloading selection according to received response message. The task is executed locally as described below:

$$\begin{aligned} &\min_{\omega,\eta}(\omega E_m^{loc} + \eta E_m^{uav}) = \hat{E}_{loc} \\ &\texttt{s.t.} \;\; C3, C4, C5 \\ &C1 : t_m^L \le t_m^{max} \\ &C2 : \omega = 1, \eta = 0 \\ &C6 : E_{loc} - E_m^L \ge E_{th}^{loc} \\ &C8 : \hat{E}_{loc} < \hat{E}_{sbs} \quad (if \;\; t_m^{UE-SBS} + t_m^{SBS} \le t_m^{max}) \\ &C9 : \hat{E}_{loc} < \hat{E}_{uav} \quad (if \;\; t_m^{UE-UAV} + t_m^{UAV} \le t_m^{max}). \end{aligned} \tag{15}$$

Within the required QoS latency, if the local calculations have the lowest energy consumption and after the task is completed, the remaining energy resources of the UE are greater than the threshold energy of the UE, which is presented in $C6$, the task is executed by the UE.

(b) *The Task Is Executed at The SBS*: The UE determines whether to offload the task by comparing local execution time with maximum tolerance delay t_m^{max} and the remaining energy of the UE $E_m^{loc_{re}}$ with the threshold of energy of the UE E_{th}^{loc}.

$$\begin{aligned} D_{T-offload} = I(t_m^L, t_m^{max}) &= \begin{cases} 0 \ IF(t_m^L \le t_m^{max}) \\ 1 \ IF(t_m^L > t_m^{max}) \end{cases} \\ D_{E-offload} = I(E_m^{loc_{re}}, E_{th}^{loc}) &= \begin{cases} 0 \ \ IF(E_m^{loc_{re}} \ge E_{th}^{loc}) \\ 1 \ IF(E_m^{loc_{re}} < E_{th}^{loc}). \end{cases} \end{aligned} \tag{16}$$

When the time decision and the energy decision are both positive, the task is executed locally, otherwise the task is offloaded. The task is executed at the SBS as described below:

$$\begin{aligned} &\min_{\omega,\eta}(\omega E_m^{loc} + \eta E_m^{uav}) = \hat{E}_{sbs} \\ &\ \texttt{s.t.} \ \ C3, C4, C5 \\ &\ C1 : t_m^{UE-SBS} + t_m^{SBS} \le t_m^{max} \\ &\ C2 : \omega = 1, \eta = 0 \\ &\ C6 : E_{loc} - E_m^{T_{SBS}} \ge E_{th}^{loc} \\ &\ C10 : \hat{E}_{sbs} < \hat{E}_{uav} \quad (if \ \ t_m^{UE-UAV} + t_m^{UAV} \le t_m^{max}). \end{aligned} \tag{17}$$

When the QoS latency requirement of the task is guaranteed for offloading to the SBS, if the energy consumption is the lowest, and after the task offloading is completed, the remaining energy resources of the UE are greater than the threshold energy of the UE, the task is executed at the SBS.

(c) *The Task Is Executed at The UAV*: For computation-intensive and latency-sensitive tasks, the UE can not guarantee the Qos and select to offload the task [12]. Compared to offload to the SBS, UAV-enabled MEC provides higher bandwidth and more stable communication link, which can ensure more critical QoS latency requirement of the task. The task is executed at the UAV as described below: executed at the SBS as described below:

$$
\begin{aligned}
&\min_{\omega,\eta}(\omega E_m^{loc} + \eta E_m^{uav}) = \hat{E}_{uav} \\
&\texttt{s.t.}\quad C2, C3, C4, C5 \\
&C1 : t_m^{UE-UAV} + t_m^{UAV} \le t_m^{max} \\
&C6 : E_{loc} - E_m^{T_{UAV}} \ge E_{th}^{loc} \\
&C7 : E_{uav} - E_m^{E_{UAV}} \ge E_{th}^{uav} \\
&C11 : \hat{E}_{uav} < \hat{E}_{sbs} \quad (if \;\; t_m^{UE-SBS} + t_m^{SBS} \le t_m^{max}).
\end{aligned} \tag{18}
$$

When the QoS latency requirement of the task is guaranteed for offloading to the UAV, if the remaining energy of the UE and the UAV are greater than the threshold energy after the task is completed and the energy consumption is the lowest, the task is executed at the UAV.

The detail steps of a task-aware joint computation offloading scheme are described in Algorithm 1.

4 Simulation Results and Discussions

In this section, we examine the performance of the task-aware joint computation offloading scheme based on ground offloading and air offloading. For comparison, we also simulate the following typical scheme: (1) just consider that tasks are processed locally at the UE, which is denoted as scheme(1); (2) consider local execution and the ground-offloading without air-offloading, which is denoted as scheme (2); (3) consider local execution and the air-offloading without ground-offloading, which is denoted as scheme (3); the joint computation offloading scheme proposed in this paper is denoted as scheme (4).

In this simulation, Tables 3 and 4 respectively indicate the size of the considered tasks, a certain value of maximum allowable latency for each task. The computation capability of the UE, the SBS and the UAV are 2.2 GHz, 5 GHz, 4 GHz respectively [13]. UAV is assumed to fly at a fixed altitude A = 80 m. Most other parameters used in the simulation are summarized in Table 5.

Figure 2 illustrates optimal execution location of each task for different applications based on the proposed scheme. The result shows that for computation-intensive or latency-sensitive computing tasks, the UE cannot meet its requirements. And the computing tasks are offloaded to the UAV or UE by air-offloading and ground-offloading according to the delay and energy consumption of each computing task. If there is no air-offloading or ground-offloading, a lot of tasks will be blocked.

Algorithm 1. The Latency-aware and Energy-aware MEC Based on a Joint Computation Offloading Scheme

1: **Initialization**:
a)Set f^L, f^{UAV}, $\{P_m^L, P_m^T, P_m^{UAV}\}$, $m \in M$, ω, η, β_0, A_{UAV}, W_{UE-SBS}, E_{loc}, E_{uav}, E_{th}^{loc}, E_{th}^{uav}
b)Calculate G_{UE-SBS}, H_{UE-UAV}
c)Initialize three terms (d_m, c_m, t_m^{max}) of $task_m$, $m \in M$, according to different applications;
2: **for** $m = 1, \cdots, M$ **do**
3: Calculate t_m^L, E_m^L, t_m^{UE-SBS}, t_m^{SBS}, E_m^TSBS, t_m^{UE-UAV}, t_m^{UAV}, E_m^EUAV, E_m^TUAV
4: **if** $t_m^L \leq t_m^{max}$ & $E_{loc} - E_m^L \geq E_{th}^{loc}$ **then**
5: **if** $t_m^{UE-SBS} + t_m^{SBS} \leq t_m^{max}$ & $E_{loc} - E_m^TSBS \geq E_{th}^{loc}$ **then**
6: **if** $t_m^{UE-UAV} + t_m^{UAV} \leq t_m^{max}$ & $E_{loc} - E_m^TUAV \geq E_{th}^{loc}$ & $E_{uav} - E_m^EUAV \geq E_{th}^{uav}$ **then**
7: Compare the size of $\hat{E}_{loc}, \hat{E}_{uav}, \hat{E}_{sbs}$
8: Select the optimal offloading location with the lowest energy consumption in line 7
9: **else**
10: Compare the size of $\hat{E}_{loc}, \hat{E}_{sbs}$
11: Select the optimal offloading location with the lowest energy consumption in line 10
12: **end if**
13: **else if** $t_m^{UE-UAV} + t_m^{UAV} \leq t_m^{max}$ & $E_{loc} - E_m^TUAV \geq E_{th}^{loc}$ & $E_{uav} - E_m^EUAV \geq E_{th}^{uav}$ **then**
14: Compare the size of $\hat{E}_{loc}, \hat{E}_{uav}$
15: Select the optimal offloading location with the lowest energy consumption in line 14
16: **else**
17: E=$\hat{E}_{loc}$
18: The task offloads to local
19: **end if**
20: **else if** $t_m^{UE-SBS} + t_m^{SBS} \leq t_m^{max}$ & $E_{loc} - E_m^TSBS \geq E_{th}^{loc}$ **then**
21: **if** $t_m^{UE-UAV} + t_m^{UAV} \leq t_m^{max}$ & $E_{loc} - E_m^TUAV \geq E_{th}^{loc}$ & $E_{uav} - E_m^EUAV \geq E_{th}^{uav}$ **then**
22: Compare the size of $\hat{E}_{uav}, \hat{E}_{sbs}$
23: Select the optimal offloading location with the lowest energy consumption in line 22
24: **else**
25: E=$\hat{E}_{sbs}$
26: The task offloads to the SBS
27: **end if**
28: **else if** $t_m^{UE-UAV} + t_m^{UAV} \leq t_m^{max}$ & $E_{loc} - E_m^TUAV \geq E_{th}^{loc}$ & $E_{uav} - E_m^EUAV \geq E_{th}^{uav}$ **then**
29: E=$\hat{E}_{uav}$
30: The task offloads to the UAV
31: **else**
32: The task is blocked
33: **end if**
34: **end for**
35: Output: the location where the task is offloaded and average energy consumption and latency for tasks execution

Figure 3 illustrate the blocking rate and average latency of handling for each scheme. And we can find that the proposed scheme can achieve the lowest blocking rate and average delay compared other schemes. In Fig. 3, we can find that the number of blocked tasks is reduced consideration of ground-offloading besides the local computing or air-offloading besides the local computing, but there

Table 3. The size of different tasks

Task	Task(1)	Task(2)	Task(3)	Task(4)	Task(5)
d_m(KB)	100	270	320	370	600
Task	Task(6)	Task(7)	Task(8)	Task(9)	Task(10)
d_m(KB)	780	850	900	910	1000

Table 4. QoS latency for different tasks

Task	Task(1)	Task(2)	Task(3)	Task(4)	Task(5)
t_m^{max}(ms)	440	520	260	420	490
Task	Task(6)	Task(7)	Task(8)	Task(9)	Task(10)
t_m^{max}(ms)	500	470	560	600	200

Table 5. Simulation parameters

Parameters	Values
Sub-channel bandwidth W_{UE-SBS}	2.5 MHz
Sub-channel bandwidth W_{UE-UAV}	4 MHz
Processing power for P_m^L, P_m^{UAV}	[1,5, 0.8] W
Tx power	24 dBm
Bandwidth	20 MHz
Small cell path loss (in dB)	$147.4 + 43.3 log_{10}$(R)
UAV channel power gain β_0	-50 dB
Computing capability for f^L, f^{SBS}, f^{UAV}	[2.2, 5, 4] GHz
Height of UAVs	80 m
Noise power spectral density (PSD)	-174 dBm/Hz
Threshold level of energy of UE E_{th}^{loc}	0.2 WH
Threshold level of energy of UE E_{th}^{uav}	1 WH

are still some blocked tasks. The proposed scheme in this paper joint ground-offloading and air-offloading achieves the best performance. Meanwhile, it can be observed that the proposed scheme can achieve the lowest average latency based on latency-awareness and energy-awareness of tasks. Each task is offloaded to the best execution location to meet QoS latency requirement.

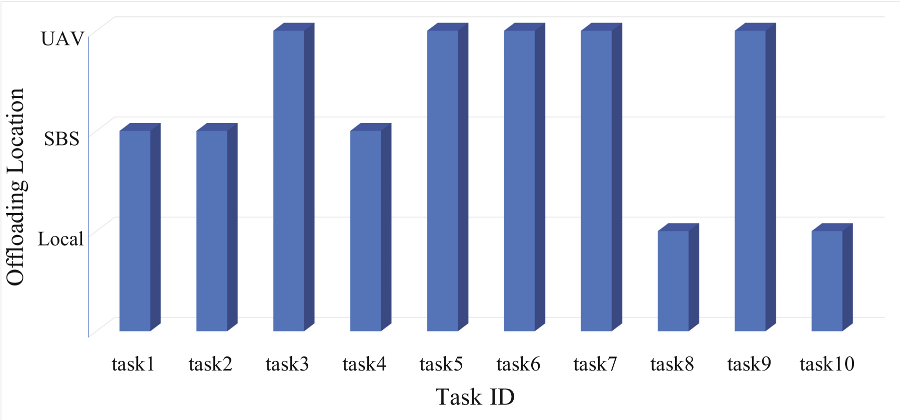


Fig. 2. Offloading location of each task.

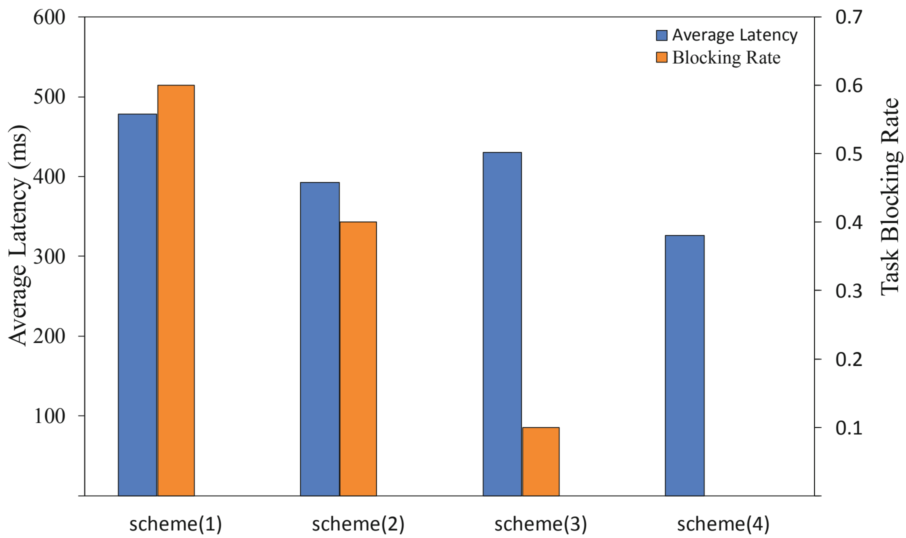
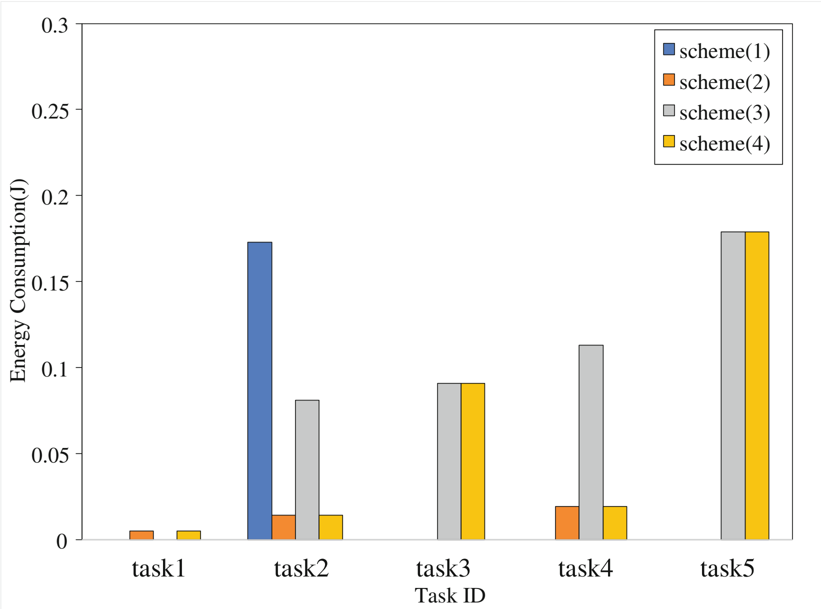
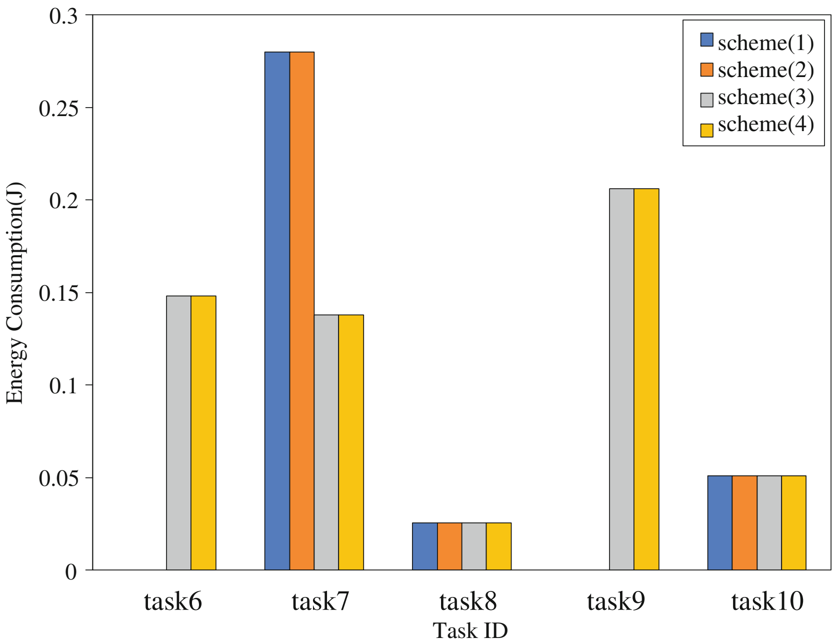


Fig. 3. Blocking rate and average latency of tasks handling for different schemes of each task.

Figure 4 shows energy consumption of each task for different schemes. And we can find that the proposed scheme can minimize the energy consumption while satisfying the QoS latency. The calculation task blocking rate even dropped to 0%. It prolongs the lifetime of the UAV and the UE while improving the performance of UAV-enabled mobile edge computing systems.



(a)



(b)

Fig. 4. Energy consumption of each task for different schemes.

5 Conclusion

In this paper, we focus on a latency-aware and energy-aware mobile edge computing for time critical applications based on a joint computation offloading scheme. The proposed scheme decides the location of task execution based on the available resources and the required QoS constraints. For different tasks with different QoS latency constraints and resource requirement, they are decided to execute locally or offload to UAV-enabled MEC servers by air-offloading method or offload to the SBS with conventional MESs by ground-offloading method. The simulation results demonstrate that our scheme achieves higher efficiency in terms of latency and blocking probability. Meanwhile, the proposed scheme minimizes the energy consumption of UEs and UAVs under the conditions that satisfy the different delay limits of each task.

Acknowledgement. This work is jointly supported by National Natural Science Foundation of China (Grant No. 61671088), and the National Natural Science Foundation of China (Grant No. 61771070).

References

1. Gupta, L., Jain, R., Vaszkun, G.: Survey of important issues in UAV communication networks. IEEE Commun. Surv. Tutor. **18**(2), 1123–1152 (2016). Secondquarter
2. Fan, L., Yan, W., Chen, X., Chen, Z., Shi, Q.: An energy efficient design for UAV communication with mobile edge computing. China Commun. **16**(1), 26–36 (2019)
3. Ruan, L., et al.: Energy-efficient multi-UAV coverage deployment in UAV networks: a game-theoretic framework. China Commun. **15**(10), 194–209 (2018)
4. Li, J., Liu, Q., Wu, P., Shu, F., Jin, S.: Task offloading for UAV-based mobile edge computing via deep reinforcement learning. In: 2018 IEEE/CIC International Conference on Communications in China (ICCC), pp. 798–802, August 2018
5. Hu, Q., Cai, Y., Yu, G., Qin, Z., Zhao, M., Li, G.Y.: Joint offloading and trajectory design for UAV-enabled mobile edge computing systems. IEEE Internet Things J. **6**(2), 1879–1892 (2019)
6. Zhou, F., Wu, Y., Sun, H., Chu, Z.: UAV-enabled mobile edge computing: offloading optimization and trajectory design. In: 2018 IEEE International Conference on Communications (ICC), pp. 1–6, May 2018
7. Sharma, V., You, I., Jayakody, D.N.K., Reina, D.G., Choo, K.R.: Neural-blockchain based ultra-reliable caching for edge-enabled UAV networks. IEEE Trans. Ind. Inform. **15**, 5723–5736 (2019)
8. Du, Y., Wang, K., Yang, K., Zhang, G.: Energy-efficient resource allocation in UAV based MEC system for IoT devices. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, December 2018
9. Zhou, F., Wu, Y., Hu, R.Q., Qian, Y.: Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems. IEEE J. Sel. Areas Commun. **36**(9), 1927–1941 (2018)
10. Bai, T., Wang, J., Ren, Y., Hanzo, L.: Energy-efficient computation offloading for secure UAV-edge-computing systems. IEEE Trans. Veh. Technol. **68**, 6074–6087 (2019)

11. Zhang, J., et al.: Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing. IEEE Internet Things J. **6**(2), 3688–3699 (2019)
12. Qian, Y., Wang, F., Li, J., Shi, L., Cai, K., Shu, F.: User association and path planning for UAV-aided mobile edge computing with energy restriction. IEEE Wirel. Commun. Lett. **8**, 1312–1315 (2019)
13. Khuwaja, A.A., Chen, Y., Zhao, N., Alouini, M., Dobbins, P.: A survey of channel modeling for UAV communications. IEEE Commun. Surv. Tutor. **20**(4), 2804–2821 (2018). Fourthquarter