Computation offloading game in multiple unmanned aerial vehicle-enabled mobile edge computing networks

Yanling Ren1 | Zhibin Xie1,2 | Zhenfeng Ding3 | Xiyuan Sun1 | Jie Xia1 | Yubo Tian1,2

1 School of Electronic Information, Jiangsu University of Science and Technology, Zhenjiang, China
2 Intelligent Marine Information Sensing and Transmission Laboratory, Zhenjiang, China
3 PetroChina Research Institute of Petroleum Exploration and Development, Beijing, China

Correspondence
Zhibin Xie, School of Electronic Information, Jiangsu University of Science and Technology, Zhenjiang, China.
Email: xiezhibin@just.edu.cn

Abstract
Because of extreme sensitivity to time and energy consumption, many computation- and data-intensive tasks are difficult to implement on mobile terminals and cannot meet the needs of the rapid development of mobile networks. To solve this problem, mobile edge computing (MEC) appears to be a promising solution. In this study, we propose two offloading schemes in the multiple unmanned aerial vehicles (UAVs) enabled MEC network. Their optimisation goals are to minimise the global computing time and energy consumption of all UAVs, respectively. Different from previous research, the UAV can perform tasks locally or offload an appropriate percentage to the desired MEC server in the two proposed schemes. In order to get the minimum global computing time, we prove the existence condition and obtain the optimal offloading proportion. In addition, in order to minimise global energy consumption, we also obtain the optimal offloading proportion and present the optimal transmission power through solving Karush–Kuhn–Tucker conditions. Finally, because UAVs are selfish, we adopt the game theory to get optimal solutions of the proposed offloading strategies. Numerical results verify that the proposed schemes can effectively decrease the global computing time and energy consumption, especially for a large number of UAVs.

1 INTRODUCTION

With high mobility and low cost, unmanned aerial vehicles (UAVs) have been widely used in the past few decades. Initially, UAVs were mainly used in the military field to reduce pilot losses [1]. In recent years, with the ever-decreasing cost and miniaturisation of equipment, UAVs have many new applications in the civilian and commercial fields, such as remote sensing monitoring [2], traffic control [3], cargo transportation [4], and emergency rescue [5].

Among the various UAV-enabled applications, the use of UAVs for achieving high-speed wireless communications was expected to play an important role in future communication systems [6]. In fact, compared to conventional wireless communications, UAV-enabled wireless communications can provide high-speed wireless connectivity in areas without infrastructure coverage [7]. Due to the approximate free-space propagation between the UAV and the cellular network, UAV-enabled wireless communication can always achieve high throughput [8, 9]. In [10], the UAV provided wireless communication is used as a flight base station (BS) for device-to-device communication networks. In [11–13], UAV acted as a relay, which was an effective technical solution for wireless communication between remote or blocked ground terminals. In [14, 15], UAV was used as a mobile data collector to extend network lifetime. Besides, UAVs can be utilised in many communication applications. A new UAV-enabled wireless power transfer system was considered in [16], where a UAV-mounted energy transmitter broadcasts wireless energy to charge receivers on the ground. The issue of secure transmission was considered in [17], where UAV provides video for user equipment (UE) in some small cells. Some researchers have studied the optimisation of the position and trajectory of the UAVs to achieve maximum user coverage [18–21].

The problem of energy consumption in wireless networks always is the focus of research [22–26]. Similarly, due to the limited resource of UAVs, the authors considered the energy consumption and resource allocation in the networks [23]. Recently,
The optimal resource allocation scheme \cite{32, 33}. In \cite{34, 35}, the authors jointly optimised user association, power control, and location planning, and solved the problem of minimising the total sum power of multiple UAVs.

Many studies have been published on the MEC optimisation methods. For different optimisation goals, different offloading optimisation methods can be used in the computational offloading. In \cite{29, 30}, the authors modelled the computational offloading optimisation problem as an integer programming problem to obtain the optimal solution for the task offloading scheme. The authors of \cite{31, 32} both used heuristic algorithms to obtain approximate global optimal solutions. There are also some researchers that used convex optimisation algorithm to solve the resource allocation problem in MEC and obtained the optimal resource allocation scheme \cite{28, 33}. In \cite{34, 35}, the scholars adopted different game theory models to simulate the interaction among the MEC servers and users and proposed the optimal offloading strategies by formulating cost function.

In addition, some scholars pay attention to the problem of computation offloading in UAV-enabled MEC network \cite{36–39}. In \cite{36}, the computation efficiency was maximised by jointly optimising the offloading time, the transmit power of the user and the trajectory of the UAV. In \cite{37}, the authors considered minimising the total energy consumption of the system and meeting the quality-of-service requirements of mobile applications. Although there are some excellent works on UAV-enabled MEC networks in terms of computation offloading, UAVs were mostly deployed as flying MEC platforms to provide computing resources to UEs. In fact, UAVs only have limited hardware and energy resources. When UAVs need to deal with very heavy computing tasks, the calculation of heavy tasks can lead to slower response speed and also be detrimental to battery life, which may eventually affect the success of the tasks. Therefore, if UAVs offload computing tasks to MEC servers, it can effectively reduce computing time and reduce energy consumption. Compared with \cite{36–38}, UAVs are deployed as UEs in \cite{39}. And the authors considered the offloading problem of computing tasks of UAVs and achieved the best balance among energy consumption, time delay and calculation cost. However, they considered the computation offloading as a binary offloading mode and ignored the fact that the computing tasks could be split, which might result in a suboptimal offloading strategy.

In this study, we consider the UAV-enabled MEC network, in which UAVs offload computing tasks to the MEC server. Due to most applications being sensitive to computation time and UAVs have limited hardware and energy resources, we consider two problems of minimising computing time and energy consumption. The optimised time includes the transmission time and the computing execution time of the computing task on the UAV and MEC server. The optimised energy consumption takes into account the energy consumption of all UAVs. In these two problems, the computing tasks can be split, and our purpose is to obtain the optimal offloading proportion. We prove the existence condition of the minimum value in the computing time problem. In addition, we also obtain the optimal transmission power in the energy consumption problem.

We propose two optimisation schemes, that is, the minimum computing time and the minimum energy consumption schemes. In the minimum computing time problem, we prove the existence conditions of the optimal offloading proportion and also obtain the optimal solution of computing time. In the minimum energy consumption problem, we first obtain the optimal offloading proportion. Then, we obtain the optimal transmission power by solving the Karush–Kuhn–Tucker (KKT) conditions.

3. In order to solve the two schemes effectively, we convert the optimisation problems into the offloading strategy selection problems and present the solutions based on game theory. Simulation results show that the algorithms can achieve convergence and effectively reduce global computing time and energy consumption for UAVs.

The remainder of this study is organised as follows. Section 2 introduces the network scenario and system model. We give two optimal computation offloading strategy in Section 3, which includes two optimisation problems and two offloading algorithms. Numerical simulation results and some theoretical analysis are presented in Section 4. Finally, conclusions are given in Section 5.

2 | SYSTEM MODEL

2.1 | Network model

As shown in Figure 1, we consider a multi-UAV and multi-MEC servers’ network, where \( N \) UAVs are covered by a macro base station (MBS) and \( M \) small base stations (SBSs). In this network, the SBSs and UAVs are evenly distributed. The SBSs are connected to the core network through wired fibers. One lightweight MEC server is deployed around one SBS so that the
SBS has MEC capacity. The sets of UAVs and MEC servers are denoted by $i \in \mathbb{N} = \{1, 2, ..., N\}$ and $j \in \mathbb{M} = \{1, 2, ..., M\}$, respectively. The MBS can provide assist communication among UAVs and SBSs. In order to avoid interference among multiple UAVs, multiple access technology, for example, time division multiple access (TDMA) can be adopted.

If the computing tasks are heavy for UAVs, the UAVs will choose to offload them to the MEC server according to their hardware resources and latency requirements. The UAV first uploads its computing task information and location information to the SBS, and the SBSs within MEC network exchange the information with each other. Then, the SBSs return the information of others to each UAV connected to itself. For any computation task $I_i$ of UAV $i$, it can be described in two terms, that is,

$$I_i = (b_i, \lambda), \quad \forall i \in N,$$

where $b_i$ denotes the data size of the computing task $I_i$ transmitted to the MEC server $j$, and $\lambda_i$ is the required CPU cycles to finish $I_i$.

The UAV $i$ can offload computing tasks to any one MEC server, or perform computing tasks locally without offloading. Let $\Lambda_{i,j} \in \{0, 1\}$ denote the offloading factor between the UAV $i$ and the MEC server $j$, where $\Lambda_{i,j} = 1$ means that the UAV $i$ decides to offload the computing task to the MEC server $j$, otherwise $\Lambda_{i,j} = 0$. Assume that each UAV can only select one MEC server to offload the computing task, that is, $\sum_{j=1}^{M} \Lambda_{i,j} = 1$, $i = 1, ..., N$, and the MEC server $j$ can provide computing service for multiple UAVs at the same time, that is,

$$0 \leq \sum_{i=1}^{N} \Lambda_{i,j} \leq N, \quad j = 1, ..., M.$$

### 2.2 Communication model

Assume that UAV’s location change within each iteration slot can be ignored, compared to the distance from the UAV to the MEC server. Denote that the coordinate of UAV $i$ are $(x_i, y_i, h_i)$ and the coordinate of MEC server $j$ are $(x_j, y_j, 0)$ as shown in Figure 1. The horizontal distance between UAV $i$ and MEC server $j$ is calculated as

$$R_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$  

The data transmission rate $r_i$ of UAV $i$ can be expressed

$$r_i = B \log_2 \left( 1 + \frac{p_i \alpha}{R_{i,j}^2 + b_i^2} \right),$$

where $B$ represents the bandwidth of the selected channel when the task is offloaded, $p_i$ represents the transmission power of UAV $i, \alpha = g_0 / \sigma^2$, $g_0$ as the channel power gain at the reference distance 1 m and set $g_0 = 1.42 \times 10^{-4}$, $\sigma^2$ represents the noise power [28].

### 3 OPTIMAL COMPUTATION OFFLOADING STRATEGY

This section is divided into two sub-sections. The first part describes the problem of the minimum computing time. The computing time includes transmission time and execution time of the tasks on the UAVs and MEC server side. The second section describes the problem of the minimum energy consumption. The energy consumption only considers the energy consumption of all UAVs in the execution of computing tasks.

#### 3.1 The minimum computing time

When UAV $i$ runs a computing task, there are two possible execution locations of the computing tasks. One is performed locally by the UAV $i$, and the other is offloaded to the MEC server. Limited by the hardware resource, it would be the first choice to offload the computing task to the MEC server. However, if UAVs offload all their tasks to MEC server at the same time, it will cause a certain amount of congestion, which increases the task computing time. Therefore, the minimum computing time problem is to find an optimal computation offloading strategy for all UAVs’ computing tasks, which minimises the global task computing time.

#### 3.1.1 Problem formulation

Suppose that the task offloading proportion of UAV $i$ is $x_i$, so the locally computing proportion is $(1 - x_i)$. Then, we know that the local computing time is

$$T_{i,j}^{\text{local}} = (1 - x_i) \frac{f_i}{f_i^{\text{local}}},$$

where $f_i^{\text{local}}$ represents the local computing capacity of the UAV $i$.

If the UAV $i$ selects MEC server $j$ to offload the computing task, the task processing time $T_{i,j}^{\text{offload}}$ mainly contains two parts,
that is, the transmission time and computing execution time of the task processed by the MEC server $j$. Then, the task processing time on the MEC server $j$ side can be expressed as

$$T_{i,j} = x_i \left( \frac{b_i}{r_i} + \frac{s_i}{f_{i,j}} \right),$$  \hspace{1cm} (5)

where $f_{i,j} = f_j / \sum_{i=1}^{N} \lambda_{i,j}$, it represents the computing capacity allocated by the MEC server $j$ to the UAV $i$, and where $\sum_{i=1}^{N} \lambda_{i,j}$ indicates the number of UAVs that all choose to offload the computing task to the MEC server $j$.

Because of simultaneously happening of the UAV and MEC server computation, the task computing time of the UAV $i$ is

$$T_{i} = \max\{T_{i,local}, T_{i,cloud}\},$$  \hspace{1cm} (6)

Based on Equation (6) and offloading factor $\lambda_{i,j}$, the computing time of UAV $i$ can be rewritten as

$$T_{i}(\lambda_{i,j}) = \begin{cases} 0, & \text{if } \lambda_{i,j} = 0, \\ T_{i,j}, & \text{if } \lambda_{i,j} = 1. \end{cases}$$  \hspace{1cm} (7)

This indicates that the computing time is valid when the UAV $i$ has established the offloading relationship with the MEC server $j$.

Our final optimisation goal is to minimise the task computing time of all UAVs; so let $T = \sum_{i=1}^{N} \sum_{j=1}^{M} T_{i}(\lambda_{i,j})$. The minimum computing time problem can be described as follows:

$$\begin{align*}
\min_{\{x_i, \lambda_{i,j}\}} & \sum_{i=1}^{N} \sum_{j=1}^{M} T_{i}(\lambda_{i,j}) \\
\text{s.t.} & \sum_{j=1}^{M} f_{i,j} \leq f_{j}^{\max}, \\
& T_{i,j} \leq T_{i}^{\max}, \\
& 0 < x_i \leq \frac{f_{i}^{\max}}{f_{j}^{\max}}, \\
& \frac{b_{i}}{r_{i}} \geq 0.
\end{align*}$$  \hspace{1cm} (8)

In Equation (8), the first constraint indicates that the sum of the computing resources allocated by the MEC server $j$ for the UAVs does not exceed its maximum computing capacity; the second constraint indicates that the total computing time is lower than the maximum tolerable execution time $T_{i}^{\max}$ of the UAV $i$; the third constraint represents the range of the uplink transmission power of the UAV $i$; the fourth constraint indicates that the UAV $i$ has a certain computing capacity $f_{j}^{local}$.

### 3.1.2 Optimal offloading proportion for the minimum computing time

After one MEC server is selected, the proportion $x_i$ of offloading task determines $T_{i}^{local}$ and $T_{i}^{cloud}$. According to Equation (6), we can know that the task computing time $T_{i}$ of the UAV $i$ depends on the maximum value of the local computing time $T_{i}^{local}$ and the MEC server-side computing time $T_{i}^{cloud}$.

#### Lemma 1

If and only if $T_{i}^{local} = T_{i}^{cloud}$, the optimal $x_i$ exists and the task computing time $T_{i}$ reaches a minimum value, and the optimal offloading proportion $x_i$ as below:

$$x_i^* = \frac{x_i}{\frac{b_{i}}{r_{i}} + \frac{f_{i}^{\max}}{f_{j}^{\max}} + \frac{b_{i}}{f_{j}^{local}}},$$  \hspace{1cm} (9)

**Proof:**

The computing time of the UAV $i$ has three possible situations, that is, $T_{i}^{local} > T_{i}^{cloud}$, $T_{i}^{local} < T_{i}^{cloud}$, $T_{i}^{local} = T_{i}^{cloud}$.

**Case 1:** When $T_{i}^{local} > T_{i}^{cloud}$, there is $T_{i}^{\ast} = T_{i}^{local}$, then $(1 - x_i)\frac{b_{i}}{f_{i}^{local}} > x_i\left(\frac{b_{i}}{r_{i}} + \frac{f_{i}^{\max}}{f_{j}^{\max}}\right)$, it follows that $x_i < \frac{\frac{b_{i}}{r_{i}} + \frac{f_{i}^{\max}}{f_{j}^{\max}}}{\frac{b_{i}}{f_{i}^{local}} + \frac{b_{i}}{f_{j}^{local}}}$.

Assume $\varphi = \frac{\frac{b_{i}}{f_{i}^{local}} + \frac{b_{i}}{r_{j}^{local}}}{\frac{b_{i}}{f_{i}^{local}} + \frac{b_{i}}{f_{j}^{local}}}$, then $x_i < \varphi$. We obtain $T_{i}^{local} > (1 - \varphi)\frac{b_{i}}{f_{i}^{local}}$.

**Case 2:** When $T_{i}^{local} < T_{i}^{cloud}$, there is $T_{i}^{\ast} = T_{i}^{cloud}$, similar to case 1, we have $T_{i}^{cloud} > \varphi\left(\frac{b_{i}}{r_{i}} + \frac{f_{i}^{\max}}{f_{j}^{\max}}\right)$.

**Case 3:** When $T_{i}^{local} = T_{i}^{cloud}$, there is $x_i = \varphi$. So, we obtain $T_{i}^{local} = (1 - \varphi)\frac{b_{i}}{f_{i}^{local}}$ and $T_{i}^{cloud} = \varphi\left(\frac{b_{i}}{r_{i}} + \frac{b_{i}}{f_{j}^{local}}\right)$.

In summary, the optimal $T_{i}^{\ast}$ exists in case 3, that is, when $T_{i}^{local} = T_{i}^{cloud}$, the task computing time $T_{i}$ have a minimum value.

Based on Equation (8), the problem (7) can be converted to

$$\begin{align*}
\min_{\{x_i, \lambda_{i,j}\}} & \sum_{i=1}^{N} \sum_{j=1}^{M} (1 - x_i) \frac{b_{i}}{f_{j}^{local}} \\
\text{s.t.} & \sum_{j=1}^{M} f_{i,j} \leq f_{j}^{\max}, \\
& T_{i} \leq T_{i}^{\max}, \\
& 0 < x_i \leq \frac{f_{i}^{\max}}{f_{j}^{\max}}, \\
& \frac{b_{i}}{r_{i}} \geq 0.
\end{align*}$$  \hspace{1cm} (10)

In order to solve Equation (9), we need to obtain the optimal offloading proportion $x_i$ according to the distance among the UAV and the BS and the computing capacity of the MEC server. Therefore, the problem (9) can be transformed to an offloading strategy problem.

#### 3.1.3 Optimal offloading strategy for the minimum computing time

To solve the offloading strategy problem, we adopt the game approach to find the optimal strategy and the minimum computing time. Game theory is considered as a powerful tool to analyse the conflicts among multiple game players that are supposed to realise their own interests.

We adopt a dynamic non-cooperative game to obtain efficient multi-UAV computation offloading decisions. The dynamic
game refers to the fact that the players can observe the actions of the players who acted before them and make their own decision accordingly. In the non-cooperative game, the players do not have a common agreement, and each player pursues the best personal decision.

Let $a_i$ indicate the offloading strategy of UAV $i$, and let $a_{i,j}$ indicate the offloading strategy of the other UAVs excluding UAV $i$. The goal of the game is to minimise the computing time of each UAV, that is,

$$\min_{\{a_{i,j}=0,1\}} T_j(a_i,a_{\sim i}), \quad \forall i \in N, j \in M, \quad (11)$$

where $T_j(a_i,a_{\sim i})$ is the computing time function of UAV $i$ based on the current strategies, and it can be expressed as

$$T_j(a_i,a_{\sim i}) = \sum_{j=1}^{M} T_{ij}(a_{ij}), \quad \forall i \in N. \quad (12)$$

The offloading strategy problem can be described as a distributed offloading strategy game, which can be denoted by

$$\Gamma_1 = (\{a_i\}_{i \in N}, \{T_j(a_i,a_{\sim i})\}_{i \in N}), \quad (13)$$

where $N$ represents all game players, $\{a_i\}_{i \in N}$ is the strategy set of player $i$, $\{T_j(a_i,a_{\sim i})\}_{i \in N}$ is the computing time function of UAV $i$ in the game.

Nash equilisation (NE) is a state of non-cooperation game, in which no player can improve its utility by changing its decision. According to the NE existence theorem, the games where the players have a finite pure strategy set, there is at least one NE point [40]. Our offloading strategy game has finite players and offloading strategy space, and each player can only choose a pure strategy from a limited set of offloading strategy. Thus, our game has the NE. That is, in each iteration, UAV determines whether there is a better decision based on the existing decisions of other UAVs then requests to update it. But in each iteration, the network only updates one UAV decision. Based on the premise that each UAV is rational, the game will reach NE over a limited iteration. In this state of NE, no UAV can further reduce computing time by changing its strategy. The strategy of the equilibrium point is denoted as

$$a_{\text{ime}}^* = (a_1^{\text{ime}}, \ldots, a_i^{\text{ime}}, \ldots, a_{i+1}^{\text{ime}}, \ldots, a_K^{\text{ime}}), \quad (14)$$

where $a_{\text{ime}}^*$ is the optimal strategy for the minimum computing time. In addition, the optimal computing time $T_j$ can be expressed as

$$T_j(a_i^{\text{ime}},a_{\sim i}) \leq T_j(a_i,a_{\sim i}). \quad (15)$$

In order to get the equilibrium point strategy $a_{\text{ime}}^*$, we propose an effective algorithm, that is, MCTA. In this algorithm, it is assumed that UAVs can get all the information, including the distance, the computing capacity of the MEC servers, and the size of computing tasks. First, each UAV selects the nearest

**Algorithm 1 Minimum Computing Time Algorithm**

1. Initialise: $N, M, B, p_i, \sigma_i, b_i, s_i, f_{\text{max}}$; an offloading strategy space of the UAVs($a_i,a_{\sim i}$); the initial offloading strategy of the UAV $i$, that is,$(a_i,a_{\sim i})$; the initial computing time $T_i(a_i,a_{\sim i})$ based on the initial strategy; the update set $D = \emptyset$ of the UAV $i$; the computation time update set $\mathbb{F} = \emptyset$.
2. for each iteration $l$
3. for all UAV $i$
4. change strategy, compute $x_i$, $T_i(\lambda_{ik})$, and compute whether $T_i(\lambda_{ik}) < T_i(\lambda_{ik})$ exists;
5. if $T_i(a_i,a_{\sim i}) < T_i(a_i,a_{\sim i})$ then
6. save $\lambda_{ik}$ to $D$, save $T_i(a_i,a_{\sim i})$ to $T$
7. end if
8. end for
9. while $\mathbb{F} = \emptyset$ do
10. find the minimum time $T_{i_{\text{time}},a_{\sim i}}$ from $T_i$;
11. update UAV’s strategy, i.e., $a_i = a_{i_{\text{time}}}$;
12. end while
13. while $(a_i, a_{\sim i}) \neq (a_i, a_{\sim i})$ do
14. return step 3;
15. end while
16. end for

MEC server as the initial strategy. Then, within each iteration, all UAVs calculate whether a better strategy exists and determine whether it is globally optimal. If $T_i(a_i, a_{\sim i}) < T_i(a_i, a_{\sim i})$, store it to the updated $D_i, T_i$. When the $D_i$ is not empty, the system updates the optimal strategy and broadcasts to other UAVs. Finally, all UAVs have no better choice than current strategies, which indicates that the network has reached the NE, and the proposed algorithm has converged to the global optimal solution.

### 3.2 The minimum energy consumption

Due to the limited energy resource of UAV, we consider the problem of energy consumption in this section. Similar to the minimum computing time problem, UAV can offload any proportion of computing tasks to the MEC server. In addition, in order to minimise energy consumption, we also consider the transmission power of UAV. The minimum energy consumption problem is to find the optimal computation offloading strategy so as to meet the real-time requirements of the computing task while minimising the energy consumption of all UAVs.

#### 3.2.1 Problem formulation

If the UAV $i$ chooses local computing, the energy consumption can be given by

$$E_i^{\text{local}} = s_i \cdot \mathcal{E}_{\text{local}}, \quad (16)$$

where $s_i$ is the distance between the UAV $i$ and the MEC server, and $\mathcal{E}_{\text{local}}$ is the energy consumption of local computing.
where $\varepsilon_{\text{local}}$ denotes the consumed energy per CPU cycle of UAV $i$.

If the UAV $i$ chooses to offload the computing task with the proportion $x_i$ to the MEC server $j$, then the energy consumption can be given by

$$E_{i,j}^{\text{cloud}} = p_i \cdot \frac{x_i b_j}{r_i} + (1 - x_i)s_j \cdot \varepsilon_{\text{local}},$$

(17)

where the first item is the energy consumption of the UAV, which transmits the computing task. The second item is the energy consumption of the UAV, which handles the rest of the computing task.

Based on Equation (17) and offloading factor $\lambda_{i,j}$, the energy consumption of UAV $i$ can be rewritten as

$$E_i(\lambda_{i,j}) = \begin{cases} 0, & \text{if } \lambda_{i,j} = 0, \\ E_{i,j}^{\text{cloud}}, & \text{if } \lambda_{i,j} = 1. \end{cases}$$

(18)

Our final optimisation goal is to minimise the energy consumption of all UAVs; so let $E = \sum_{i=1}^{N} \sum_{j=1}^{M} E_i(\lambda_{i,j})$. The minimum energy consumption problem can be mathematically formulated as

$$\begin{align*}
\min_{\{\lambda_{i,j}, p_i, \forall i\}} & \quad E \\
\text{s.t.} & \quad E_{i,j}^{\text{cloud}} < E_i^{\text{local}} \\
& \quad T_i \leq T_i^{\text{max}} \\
& \quad \sum_{j=1}^{M} f_{i,j} \leq f_j^{\text{max}} \\
& \quad 0 < p_i \leq p_i^{\text{max}} \\
& \quad \sum_{j=1}^{M} \lambda_{i,j} = 1.
\end{align*}$$

(19)

In Equation (19), the first constraint indicates that the UAV’s energy consumption for offloading computing task should be less than executing locally; the second constraint indicates that the total execution time is lower than the maximum tolerable time $T_i^{\text{max}}$ of each UAV $i$; the third constraint represents the sum of the computing resources allocated by the MEC server $j$ for the UAVs do not exceed its maximum computing capacity; the fourth constraint denotes the range of the uplink transmission power of the UAV $i$; the fifth constraint indicates that the UAV $i$ can only select one MEC server to offload the computing task.

### 3.2.2 Optimal offloading proportion for the minimum energy consumption

For a given UAV offloading factor $\lambda_{i,j}$ and transmission power $p_i$ of problem (19), we can get the optimal solution of the offloading proportion $x_i$ as follows.

The partial derivative of Equation (17) is shown below:

$$\frac{\partial E_{i,j}^{\text{cloud}}}{\partial x_i} = \frac{b_j}{r_i} - s_j \cdot \varepsilon_{\text{local}}.$$

(20)

According to the first constraint $E_{i,j}^{\text{cloud}} < E_i^{\text{local}}$ of problem (19), we can get

$$p_i \cdot \frac{b_j}{r_i} - s_j \cdot \varepsilon_{\text{local}} < 0,$$

(21)

and implies that the function is a decreasing function respect to $x_i$.

**Lemma 2.** For the certain offload strategy and the transmission power of each UAV, the optimal offloading proportion $x_i$ can be expressed as

$$x_i = \min \left\{ \frac{T_i^{\text{max}}}{\frac{b_j}{r_i} + \frac{s_j}{\lambda_{i,j}}}, 1 \right\}.$$  

(22)

**Proof:** According to Equations (20) and (21), we can know that the larger the value of the offloading proportion $x_i$, the less the energy consumption of the UAV $i$. The range of offloading proportion $x_i$ is from 0 to 1, thus $x_i \leq 1$.

According to the second condition of Equation (19), we have $T_{i,j}^{\text{cloud}} \leq T_i^{\text{max}}$; so $x_i \leq \frac{T_i^{\text{max}}}{\frac{b_j}{r_i} + \frac{s_j}{\lambda_{i,j}}}$. Therefore, the optimal offloading proportion $x_i$ can be expressed as $x_i = \min \left\{ \frac{T_i^{\text{max}}}{\frac{b_j}{r_i} + \frac{s_j}{\lambda_{i,j}}}, 1 \right\}$.  

### 3.2.3 Optimal transmission power for the minimum energy consumption

For a certain UAV offloading factor $\lambda_{i,j}$ and offloading proportion $x_i$ of problem (19), we can get the optimal solution of the transmission power $p_i$ of each UAV $i$ as follows.

According to Equations (3) and (17), let $\alpha = \frac{\alpha}{R_i s + H_i}$; then the energy consumption of UAV $i$ can be expressed as

$$E_i = p_i \cdot \frac{x_i b_j}{\text{Blog}_2(1 + p_i \cdot \alpha)} + (1 - x_i)s_j \cdot \varepsilon_{\text{local}}.$$  

(23)

According to Equation (22), the energy consumption is only related to the first term. Let $e_i = p_i \cdot \frac{x_i b_j}{\text{Blog}_2(1 + p_i \cdot \alpha)}$, the problem (19) can be simplified as

$$\begin{align*}
\min_{\{p_i\}} & \quad \sum_{i=1}^{N} e_i \\
\text{s.t.} & \quad T_i \leq T_i^{\text{max}} \\
& \quad 0 < p_i \leq p_i^{\text{max}}.
\end{align*}$$  

(24)
We rewrite the objective function through variable substitution, that is, let \( \tau_i = \frac{\beta_i}{\log_2(1 + \beta_i \cdot \bar{\alpha})} \) and \( \bar{\alpha} = \frac{1}{\beta_i} (2 \tau_i - 1) \) and \( \log_2(1 + \beta_i \cdot \bar{\alpha}) = \frac{\beta_i}{\tau_i} \). Then, the problem (24) can be formulated as

\[
\min_{\tau_i} \sum_{i=1}^{N} \frac{\tau_i}{\bar{\alpha}} (2\beta_i / \tau_i - 1)
\]
\[\text{s.t. } \tau_i \leq \tau_i^{\max} - \frac{\beta_i}{\beta_i}, \quad \tau_i \geq \frac{\beta_i}{\log_2(1 + \beta_i^{\max} \cdot \bar{\alpha})}.
\]  

(25)

For each UAV \( i \), if \( y = \frac{\tau_i}{\bar{\alpha}} (2\beta_i / \tau_i - 1) \), the first derivative of the objective function exists, and the second derivative is

\[
\frac{\partial^2 y}{\partial \tau_i^2} = \frac{\beta_i \cdot \ln 2}{\bar{\alpha}} \cdot 2 \beta_i / \tau_i^2.
\]

(26)

Due to \( \frac{\partial^2 y}{\partial \tau_i^2} > 0 \), problem (25) is a convex problem. Then, the KKT conditions are a necessary and sufficient condition for the optimal solution.

The Lagrange function of problem (25) can be obtained as

\[
L(\tau, \mu, \nu) = \sum_{i=1}^{N} \frac{\tau_i}{\bar{\alpha}} (2\beta_i / \tau_i - 1) + \sum_{i=1}^{N} \mu_i \left( \tau_i - \tau_i^{\max} + \frac{\beta_i}{f_{i,j}} \right) + \sum_{i=1}^{N} \nu_i \left( \frac{\beta_i}{\log_2(1 + \beta_i^{\max} \cdot \bar{\alpha})} - \tau_i \right),
\]

where the variables \( \mu_i, \nu_i \) are all non-negative coefficients representing the Lagrange multipliers. The KKT conditions are as follows for \( \forall i \).

\[
\frac{\partial L}{\partial \tau_i} = \frac{1}{\bar{\alpha}} \left[ \left( 1 - \frac{\beta_i \cdot \ln 2}{\tau_i} \right) 2 \beta_i / \tau_i - 1 \right] + \mu_i - \nu_i = 0,
\]

\[
\mu_i \left( \tau_i - \tau_i^{\max} + \frac{\beta_i}{f_{i,j}} \right) = 0,
\]

\[
\nu_i \left( \frac{\beta_i}{\log_2(1 + \beta_i^{\max} \cdot \bar{\alpha})} - \tau_i \right) = 0,
\]

\[\mu_i \geq 0, \quad \nu_i \geq 0.
\]

(28)

The optimal \( \tau_i^* \) can be obtained by Equation (27). Then \( \beta_i^* \) can be derived by

\[
\beta_i^* = \frac{1}{\bar{\alpha}} (2 \beta_i / \tau_i^* - 1).
\]

(29)

Based on Equations (22) and (29), the problem (19) based on given UAV’s offloading strategy can be effectively solved by iteratively solving the offloading proportion \( \chi \) and transmission power \( \rho_i \). The specific algorithm is shown in Procedure 1.

3.2.4 Optimal offloading strategy for the minimum energy consumption

Next, we adopt the game approach to find the optimal strategy and achieve minimum energy consumption for all UAVs.

Similar to the minimum computing time problem, the game goal of this part is to minimise the energy consumption of each UAV, that is,

\[
\min_{\rho_i \in [0,1]} E_i(a_i, a_{-i}), \quad \forall i \in N, \forall j \in M,
\]

(30)

where \( E_i(a_i, a_{-i}) = \sum_{j=1}^{M} E_i(a_i, a_{-i}), \forall i \in N \) is the energy consumption for UAV \( i \) based on the current strategies.

The game can be denoted by

\[
\Gamma_2 = (K, \{A_i\}, \{E_i(a_i, a_{-i})\}_{i \in N}),
\]

(31)

where \( \{E_i(a_i, a_{-i})\}_{i \in N} \) is the energy consumption function of UAV \( i \) in the game. Similar to Equation (14), the game in Equation (31) also have the NE over a limited number of iterations. In the state of NE, no UAV can further reduce energy consumption by changing its strategy.

Similarly, in order to get the equilibrium point strategy \( \delta_{\text{energy}}^* \), we propose an effective algorithm, that is, MECA. First, each UAV selects the nearest MEC server as the initial strategy. Within each iteration, the UAV \( i \) can change strategy and iteratively solve the optimal offloading proportion \( \chi_i \) and optimal transmission power \( \rho_i \). If \( E(a_i', a_{-i}) < E(a_i, a_{-i}) \), store the updated \( D', E \). Then, the system updates the optimal strategy and broadcasts to all UAVs. At last, when the \( D' \) is empty, it indicates that the proposed algorithm has converged to the global optimal solution.

4 NUMERICAL RESULTS AND DISCUSSION

In this section, we evaluate the performance of the proposed two algorithms, that is, the MCTA and MECA. At the same
Algorithm 2 Minimum Energy Consumption Algorithm

1. Initialise: An offloading strategy space of the UAVs \((a_i, a_{\text{cloud}})\); set the initial offloading strategy of the UAVs \((a_i, a_{\text{cloud}})\); the initial energy consumption \(E_i(a_i, a_{\text{cloud}})\) based on the initial strategy; the update set \(\mathcal{B} = \emptyset\) of the UAV; the energy consumption update set \(\mathcal{E} = \emptyset\); 2. for each iteration \(l\) 3. for all UAV \(i\) do 4. execute Procedure 1 to compute \(x_i, f_i, E_i^{\text{cloud}}\); 5. compute whether \(E_i^{\text{cloud}}(\lambda_{i,k}) < E_i^{\text{cloud}}(\lambda_{i,j})\) exists; 6. if \(E(a_{i,l}^*, a_{\text{cloud}}) < E(a_{i,l}, a_{\text{cloud}})\) then 7. save \(\lambda_{i,k}\) to \(D^l\) and save \(E(a_{i,l}^*, a_{\text{cloud}})\) to \(\mathcal{E}\) 8. end if 9. end for 10. while \(\mathcal{E} = \emptyset\) do 11. find the minimum energy consumption \(E(a_{i,\text{energy}}^*, a_{\text{cloud}})\) from the set of \(\mathcal{E}\); 12. update UAVs strategy, let \(a_i = a_{i,l}^*\); 13. end while 14. while \((a_{i,l}^*, a_{\text{cloud}}) \neq (a_{i}, a_{\text{cloud}})\) do 15. return step 5; 16. end while 17. end for

In order to verify the effectiveness of our proposed algorithms, we first compare the MCTA and MECA with their benchmark algorithms, respectively. The benchmarks in this experiment are exhaustive search algorithms, and the results are shown in Figures 2(a) and (b). The exhaustive search algorithm is mainly used to traverse the set of MEC servers selected by the UAVs, that is, \(j \in M = \{1, 2, \ldots, M\}\). The number of MEC servers is set to 4, and the number of UAV is changed from 1 to 8. It can be found that in Figure 2(a), when the number of UAV is 5, the minimum computing time of the MCTA and the exhaustive algorithm is 0.000215. In Figure 2(b), when the number of UAV is 6, the difference in value between the MECA and exhaustive algorithm is only 0.001. Therefore, the computing time and energy consumption of our proposed algorithms are very consistent with the benchmark algorithms. In addition, the proposed algorithms have less complexity since the complexity of the two exhaustive algorithms are \(O(M^N)\), which indicates that our algorithms can provide the optimal solution for our problems.

Next, we present the convergence of the two algorithms for different numbers of UAVs as shown in Figures 3(a) and (b). The number of UAV is \(N = 20, 35, 50\), and the number of MEC server is 8. It can be easily seen that the computing time or energy consumption always decreases gradually as the number of iterations increases. After a limited number of iterations, the two algorithms can achieve numerical convergence. In addition, the minimum computing time and the minimum energy consumption of the two algorithms are related to the number of UAVs. The larger the number of UAV, the greater the value of the minimum computing time and energy consumption. But the needed iterations are similar when the two algorithms reach convergence, so the MCTA and MECA apply to any number of UAV.

In order to verify the performance of our proposed algorithms, we simulated a comparison with other offloading algorithms as shown in Figures 4(a) and (b). First, Figure 4(a) shows the comparison results of the binary offloading algorithm (BOA), local computing algorithm (LCA), all offloading algorithm (AOA) and the MCTA. In order to make a fair...
comparison, other algorithms also adopt game theory to obtain the optimal strategy. At this time, the number of MEC server is set to 10. Under the BOA, UAV's computing tasks are not split, and either all are offloaded to the MEC server or are executed locally. The LCA is that the computing tasks are performed locally, while the AOA is that the UAV offloads all computing tasks to the MEC server. It can be seen from Figure 4(a) that there is just a little difference among the minimum computing time of the four algorithms when the number of UAV is five. When the number of UAV reaches 50, the proposed MCTA reduces the computing time by about 45%, 53%, and 64% compared to LCA, BOA and AOA, respectively. Therefore, the MCTA can effectively reduce the computing time, especially for a large number of UAVs. This is because the MCTA can flexibly choose any percentage of computing tasks to offload.

Similarly, Figure 4(b) shows the comparison results between LCA, AOA and the proposed MECA. The other strategies also adopt game theory to obtain the optimal strategy. At this time, the number of MEC servers is set to 8. In the Figure 4(b), when the number of UAV is 20, one can see that the MECA reduce the energy consumption about 58% and 75% compared with AOA and LCA, respectively. It is clear that the proposed MECA shows the least energy consumption for the same number of UAVs. This is because the MECA can not only offload the computing tasks to any MEC server but also has the flexibility to choose any percentage of computing tasks to offload. Numerical results not only demonstrate the need to use multiple MEC servers for computing offloading in UAV-enabled MEC network but also demonstrate that the proposed MECA can effectively reduce the energy consumption of all UAVs.

5 | CONCLUSION

Most applications of UAV-enabled MEC networks are sensitive to time and energy consumption, but UAVs have limited resources. In this study, we consider the two problems of minimizing global computing time and energy consumption for UAVs. Different from the previous works, the computing tasks can be split to make full use of the computing resources of the networks. We prove the existence conditions of the minimum computing time and obtain the optimal solution of the offloading proportion. Moreover, we also consider the impact
of transmission power on the offloading proportion and obtain the minimum energy consumption. Next, in order to obtain the optimal offloading strategy, we propose the effective MCTA and MECA based on game theory. Simulations show that the proposed algorithms have less complexity and are suitable for the scenario of a large number of UAVs. In future work, we will focus on the joint optimisation problem of computing time and energy consumption and consider the situation in which the MEC server allocates computing capacity under the influence of UAV network uncertainty.

REFERENCES

1. Beard, R., et al.: Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. Proc. IEEE. 94(7), 1306–1324 (2006)
2. Berni, J., et al.: Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. IEEE Trans. Geosci. Remote Sens. 47(3), 722–738 (2009)
3. Reshma, R., Ramesh, T., Sathishkumar, P.: Security situational aware intelligent road traffic monitoring using UAVs. In: International Conference on VLSI Systems, Architectures, Technology and Applications, Bangalore, India, pp. 10–12 (2016)
4. Han, K., et al.: Unmanned aerial vehicle (UAV) cargo system: Senior design capstone project. In: IEEE Systems and Information Engineering Design Symposium, Charlottesville, VA, USA, pp. 16–16 (2004)
5. Wang, C., et al.: The adaptive vortex search algorithm of optimal path planning for forest fire rescue UAV. In: IEEE Advanced Information Technology, Electronic and Automation Control Conference, Chongqing, China, pp. 12–14 (2018)
6. Zeng, Y., Zhang, R., Lim, T.J.: Wireless communications with unmanned aerial vehicles: Opportunities and challenges. IEEE Commun. Mag. 54(5), 36–42 (2016)
7. Gupta, L., Jain, R., Vaszluis, G.: Survey of important issues in UAV communication networks. IEEE Commun. Surv. Tutorials 18(2), 1123–1152 (2016)
8. Amorim, R., et al.: Radio channel modeling for UAV communication over cellular networks. IEEE Wireless Commun. Lett. 6(4), 514–517 (2017)
9. Hua, M., et al.: Energy-Efficient optimization for UAV-Aided cellular offloading. IEEE Wirel. Commun. Lett. 8(3), 769–772 (2019)
10. Mozaffari, M., et al.: Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs. IEEE Trans. Wireless Commun. 15(6), 3949–3963 (2016)
11. Kong, L., et al.: Autonomous relay for millimeter-wave wireless communications. IEEE J. Sel. Areas Commun. 35(9), 2127–2136 (2017)
12. Fan, R., et al.: Optimal node placement and resource allocation for UAV relaying network. IEEE Commun. Lett. 22(4), 808–811 (2018)
13. Zhang, S., et al.: Joint trajectory and power optimization for UAV relay networks. IEEE Commun. Lett. 22(1), 161–164 (2018)
14. Zhan, C., Zeng, Y., Zhang, R.: Energy-efficient data collection in UAV enabled wireless sensor network. IEEE Wireless Commun. Lett. 7(3), 328–331 (2018)
15. Gong, J., et al.: Aviation time minimization of UAV for data collection from energy constrained sensor networks. In: IEEE Wireless Communications and Networking Conference, Barcelona, Spain, pp. 15–18 (2018)
16. Xu, J., Zeng, Y., Zhang, R.: UAV-enabled wireless power transfer: Trajectory design and energy optimization. IEEE Trans. Wireless Commun. 17(8), 5092–5106 (2018)
17. Zhao, N., et al.: Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment. IEEE Trans. Commun. 66(5), 2281–2294 (2018)
18. Al-Hourani, A., Kandeepan, S., Lardner, S.: Optimal LAP altitude for maximum coverage. IEEE Wireless Commun. Lett. 3(6), 569–572 (2014)
19. Zeng, Y., Zhang, R.: Energy-efficient UAV communication with trajectory optimization. IEEE Trans. Wireless Commun. 16(6), 3747–3760 (2017)
20. Zhang, S., Zeng, Y., Zhang, R.: Cellular-enabled UAV communication: Trajectory optimization under connectivity constraint. In: IEEE International Conference on Communications, Kansas City, MO, USA, pp. 20–24 (2018)
21. Khawaja, A., et al.: Optimum deployment of multiple UAVs for coverage area maximization in the presence of co-channel interference. IEEE Access 7, 85203–85212 (2019)
22. Xie, Z., et al.: The computation and analysis of energy-efficient multirelay and multihop communication scheme in wireless sensor networks. Int. J. Commun. Syst. 31(7), e3435.1–e3435.12 (2018)
23. Wang, K., Yang, K., Magarawalage, C.: Joint energy minimization and resource allocation in C-RAN with mobile cloud. IEEE Trans. Cloud Comput. 6(3), 760–770 (2018)
24. Wang, P., et al.: Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems. IEEE Internet Things J. 6(2), 2872–2884 (2019)
25. Wang, P., et al.: HetMEC: Latency-optimal task assignment and resource allocation for heterogeneous multi-layer mobile edge computing. IEEE Trans. Wireless Commun. 18(10), 4942–4956 (2019)
26. Wang, L., et al.: RL-based user association and resource allocation for multi-UAV enabled MEC. In: International Wireless Communications & Mobile Computing Conference, Tangier, Morocco, pp. 24–28 (2019)
27. Motlagh, N.H., Baghaa, M., Taleh, T.: UAV-based IoT platform: A crowd surveillance use case. IEEE Commun. Mag. 55(2), 128–134 (2017)
28. Yang, Z., et al.: Energy efficient resource allocation in UAV-enabled mobile edge computing networks. IEEE Trans. Wireless Commun. 18(9), 4576–4589 (2019)
29. Zhou, W., et al.: Joint offloading decision and resource allocation for multi-user NOMA-MEC systems. IEEE Access 7, 181100–181116 (2019)
30. Wang, W., Zhou, W.: Computational offloading with delay and capacity constraints in mobile edge. In: IEEE International Conference on Communications, Paris, France, pp. 1–6 (2017)
31. Deng, M., Tian, H., Fan, B.: Fine-granularity based application offloading policy in cloud-enhanced small cell networks. In: IEEE International Conference on Communications Workshops, Kuala Lumpur, Malaysia, pp. 1–6 (2016)
32. Zhang, H., et al.: Computation offloading considering front-haul and back-haul in small-cell networks integrated with MEC. In: IEEE Conference on Computer Communications Workshops, Atlanta, GA, USA, pp. 1–6 (2017)
33. Zhou, F., et al.: UAV-enabled mobile edge computing: Offloading optimization and trajectory design. In: 2018 IEEE International Conference on Communications, Kansas City, MO, USA, pp. 1–6 (2018)
34. Messous, M., et al.: Computation offloading game for an UAV network in mobile edge computing. In: IEEE International Conference on Communications, Paris, France, pp. 1–6 (2017)
35. Liu, M., Liu, Y.: Price-based distributed offloading for mobile-edge computing with computation capacity constraints. IEEE Wireless Commun. Lett. 7(3), 420–423 (2018)
36. Zhang, X., et al.: Resource allocation for a UAV-enabled mobile-edge computing system: Computation efficiency maximization. IEEE Access 7, 113345–113354 (2019)
37. Jeong, S., Simeone, O., Kang, J.: Mobile edge computing via a UAV mounted cloudlet: Optimization of bit allocation and path planning. IEEE Trans. Veh. Technol. 67(5), 2049–2063 (2018)
38. Zhou, F., et al.: Computation rate maximization in UAV-enabled wireless powered mobile-edge computing systems. IEEE J. Sel. Areas Commun. 36(9), 1927–1941 (2018)
39. Messous, M., et al.: A game theory based efficient computation offloading in an UAV network. IEEE Trans. Veh. Technol. 68(5), 4964–4974 (2019)
40. Nash, J.F., Equilibrium points in N-person games. Proc. Natl. Acad. Sci. U.S.A. 36(1), 48–49 (1950)

How to cite this article: Ren Y, Xie Z, Ding Z, Sun X, Xia J, Tian Y. Computation offloading game in multiple unmanned aerial vehicle-enabled mobile edge computing networks. IET Commun. 2021;15:1392–1401. https://doi.org/10.1049/cmu2.12102