Trajectory Optimization and Resource Allocation for Time Minimization in the UAV-Enabled MEC System

Xin Zhang*, Zheng Chang*, Guopeng Zhang†, Ming Li‡, Yulin Hu§
*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
†School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China
‡Dalian University of Technology, Dalian, China
§Wuhan University, Wuhan, China

2022 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-4266-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCNC51071.2022.9771719

***Abstract*—The unmanned aerial vehicles (UAVs) have been widely used in civilian environments, due to its high flexibility, low cost and ease of deployment. In this paper, an UAV-enabled mobile edge computing (MEC) system is studied, in which the UAV serves as an aerial mobile base station to provide services for a group of ground user equipments (UEs) with computation task requests. We jointly optimize the time allocation, resource allocation and the UAV flying trajectory to minimize the time required for the UAV to complete the task, subject to the constraints of different kinds of resources, energy and velocity. Due to the non-convexity of the formulated problem, we first transform it to a feasibility check problem and then divide it onto three convex optimization subproblems. By using the block coordinate descent method and the successive convex approximate (SCA) method, we propose an efficient iterative algorithm to solve the three subproblems alternately with ensured convergence. Extensive simulation results show that the proposed joint optimization algorithm reduces the task completion time compared with other schemes.**

***Index Terms*—Unmanned aerial vehicle; Mobile edge computing; Trajectory optimization; Completion time.**

I. Introduction

The mobile edge computing (MEC) system is considered as one promising platform for providing computation-intensive services to the users. Recently, integration of unmanned aerial vehicles (UAVs) and MEC receives increasing research interests due to its potential on flexible service provisioning. In the investigation of UAV-aided MEC system, the energy issue is a very important design aspect due to the UAV's energy constraint. The authors studied an UAV-aided MEC system in [1], and its total mobile energy consumption was greatly reduced. In [2], a fixed-wing UAV was used as a flight base station to handle application tasks, in which the energy consumption of terminal equipment is minimized and partial offloading is considered. The task completion time of UAVs is also one of the basic optimization goals in UAV-aided wireless network. The objective of [3] was to minimize the energy consumption and completion time of the UAV. By jointly designing the trajectory, completion time and offloading computing of the UAV, the resource allocation of the MEC system of the UAV was realized. In [4], the authors reduced the total delay energy consumption of the computing task by jointly optimizing computing offloading, content caching and resource allocation.

The trajectory plan of the UAV also has great impact on the service quality of UAV-aided MEC system. In [3], the authors studied the minimization of completion time and energy consumption in the MEC system, and jointly optimized computation offloading, resource allocation, trajectory and completion time. In [5], the authors optimized the trajectory and resource allocation, thus maximizing the energy efficiency in the UAV-aided MEC system. In addition, in order to deal with the limited mobility of users, the spatial distribution estimation technique is used to predict the location of users on the ground, so that the proposed method can still be applied.

In this paper, we consider a single UAV-aided MEC system where the UAV is employed to serve a group of UEs in a given two-dimensional (2D) area. Our goal is to minimize the task completion time by jointly optimizing time allocation, resource allocation and the UAV trajectory. The formulated time minimization problem is first transformed into a feasibility check problem, and then it is further transformed into a convex-like optimization problem. By using the block coordinate descent method and the successive convex approximate method, we propose an efficient iterative algorithm to solve three subproblems with ensured convergence. Extensive simulation experiments results validate the effectiveness and superiority of the proposed algorithm.

II. System Model and Assumption

A. System Model

As shown in Fig. 1, a three-dimensional Cartesian coordinate system is used. In the system one UAV provides services for K UEs, which are represented by $\mathcal{K} = \{1, 2, ..., K\}$, and all the users are randomly distributed in a two-dimensional horizontal region. The height of each device is 0, and its horizontal position is expressed as $\boldsymbol{z}_k = [x_k, y_k]$. In this scenario, the UAV is equipped with edge computing server and can provide computing services to the UEs. The UEs are with computation tasks and can offload them to the UAV via sending computation requests.

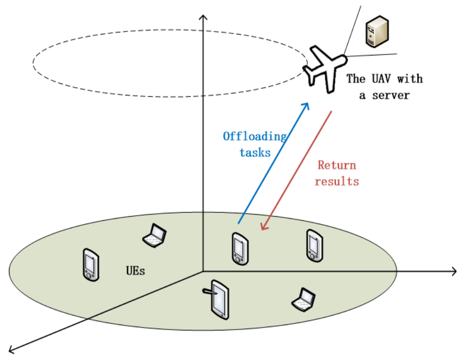

Fig. 1. UAV-enabled mobile edge computing system

B. Communication Model

The UAV flies above the UEs at a fixed height H, and its flight process is divided into N time slots, the duration of which is δ_t. When N is large enough, the position of the UAV in each time slot can be approximately unchanged, so the flight trajectory of the UAV can be defined as a series of discrete position points $\boldsymbol{q}[n] = [x_n, y_n]$. There are the following constraints on its trajectory and velocity:

$$\boldsymbol{q}[1] = \boldsymbol{q}[N], \tag{1}$$

$$v[n] = \frac{\|\boldsymbol{q}[n+1] - \boldsymbol{q}[n]\|}{\delta_t} \leq V_{max}, \forall n = 1, 2, ..., N-1, \tag{2}$$

where (1) indicates that the UAV returns to the initial point after a task completion cycle. (2) indicates the speed constraint for the UAV, where V_{max} represents the maximum flight velocity of the UAV.

Using time division multiplexing (TDMA) technology, in each time slot multiple UEs can share data transmission, so we have

$$\sum_{k=1}^{K} t_k[n] = \delta_t, \forall n. \tag{3}$$

The distance between the UE k and the UAV in the n-th slot is denoted as

$$d_k(\boldsymbol{q}[n]) = \sqrt{\|\boldsymbol{q}[n] - \boldsymbol{z}_k\|^2 + H^2}. \tag{4}$$

The line-of-sight (LoS) link is considered to model the communications between the UAV and UEs, and the channel gain is denoted as

$$g_k(\boldsymbol{q}[n]) = \rho_0 d_k^{-2}(\boldsymbol{q}[n]) = \frac{\rho_0}{\|\boldsymbol{q}[n] - \boldsymbol{z}_k\|^2 + H^2}, \tag{5}$$

where ρ_0 represents the channel gain at a reference distance of 1m. Assume that the transmission power P_k of each UE is fixed, and when offloading, the transmission rate that can be achieved from the UE k to the UAV is expressed as

$$\begin{aligned} r_k(\boldsymbol{q}[n]) &= B \log_2\left(1 + \frac{P_k g_k(\boldsymbol{q}[n])}{\sigma^2}\right) \\ &= B \log_2\left(1 + \frac{\gamma_k}{\|\boldsymbol{q}[n] - \boldsymbol{z}_k\|^2 + H^2}\right), \end{aligned} \tag{6}$$

where B is the channel bandwidth, σ^2 is the additional white Gaussian noise and $\gamma_k = \frac{P_k \rho_0}{\sigma^2}$.

In one flight cycle, it is assumed that the total amount of computing tasks of UE k is D_k. In order to ensure that all UEs can offload all their computing tasks to the UAV, the following constraint is given

$$\sum_{n=1}^{N} r_k(\boldsymbol{q}[n]) t_k[n] \geq D_k, \forall k. \tag{7}$$

C. Computing and Energy Consumption Model

It is assumed that the communication energy between the UE k and the UAV is completely provided by the UE k. Based on the proposed communication model, the communication energy of UE k in the n-th time slot is

$$E_k^{trans}[n] = P_k t_k[n]. \tag{8}$$

The transmit energy of the UE is limited by its maximum amount, which means

$$\sum_{n=1}^{N} E_k^{trans}[n] \leq E_k^{max}, \forall k, \tag{9}$$

where E_k^{max} is the maximum energy constraint of UE k.

The returned result of the server is usually quite small, so we ignore the return transmission here as in [3]. The energy consumed by the UAV mainly includes two parts: flight energy and computing energy. The flight energy model in [5] is adopted to obtain the flight energy of the UAV in each time slot:

$$E_n^{fly} = \frac{1}{2} M v_n^2 \delta_t. \tag{10}$$

According to the power model in [6], we get the computing energy of UAV in each time slot as follow:

$$E_n^c = \sum_{k=1}^{K} r(f_k[n])^3 \delta_t, \tag{11}$$

where r represents the effective switched capacitance and $f_k[n]$ represents the computing resources allocated by the UAV to UE k in time slot n.

Therefore, the total energy consumption of the UAV is

$$\sum_{n=1}^{N} E_n^{fly} + \sum_{n=1}^{N} E_n^c \leq E_{UAV}^{max}, \tag{12}$$

where E_{UAV}^{max} is the maximum energy constraint of the UAV.

In time slot n, the amount of data calculated by the UAV for UE k is $\frac{f_k[n]\delta_t}{s_k}$, where s_k represents the number of CPU cycles required to calculate each bit of data. In addition, all offloading tasks should be completed in one flight cycle. From this, the following constraint is obtained:

$$\sum_{j=n}^{N} r_k(\boldsymbol{q}[j])t_k[j] \leq \sum_{j=n+1}^{N} \frac{f_k[j]\delta_t}{s_k}, \forall k, \forall n = 1, ..., N-1. \tag{13}$$

UAV can parallelly process the different tasks from UEs [3], but due to the limited computing resources of the UAV, we can get the following constraint in each time slot:

$$\sum_{k=1}^{K} f_k[n] \leq f_{UAV}^{max}, \forall n, \tag{14}$$

where f_{UAV}^{max} denotes the maximum computing frequency of the UAV.

III. Problem Formulation

A. Problem formation

Based on the system model and assumptions, we can formulate the resource allocation and trajectory optimization problem for the considered system. The goal of the problem is to minimize the total task completion time by jointly optimizing offloading time, UAV trajectory, and computing resource. The formulated problem $(\mathbf{P1})$ can be expressed as

$$(\mathbf{P1}): \min_{\{t_k[n]\},\{\boldsymbol{q}[n]\},\{f_k[n]\}} N \tag{15}$$

$$s.t. \quad \boldsymbol{q}[1] = \boldsymbol{q}[N], \tag{15a}$$

$$v[n] \leq V_{max}, \forall n = 1, 2, ..., N-1, \tag{15b}$$

$$\sum_{k=1}^{K} t_k[n] = \delta_t, \forall n, \tag{15c}$$

$$\sum_{n=1}^{N} r_k(\boldsymbol{q}[n])t_k[n] \geq D_k, \forall k, \tag{15d}$$

$$\sum_{n=1}^{N} P_k t_k[n] \leq E_k^{max}, \forall k, \tag{15e}$$

$$\sum_{n=1}^{N} \frac{1}{2} m v_n^2 \delta_t + \sum_{n=1}^{N}\sum_{k=1}^{K} r(f_k[n])^3 \delta_t \leq E_{UAV}^{max}, \tag{15f}$$

$$\sum_{k=1}^{K} f_k[n] \leq f_{UAV}^{max}, \forall n, \tag{15g}$$

$$\sum_{j=n}^{N} r_k(\boldsymbol{q}[j])t_k[j] \leq \sum_{j=n+1}^{N} \frac{f_k[j]\delta_t}{s_k}, \forall k, \forall n = 1, ..., N-1. \tag{15h}$$

It is not difficult to find that since N is an integer and the constraints (15d) and (15h) are non-convex, $(\mathbf{P1})$ is a non-convex optimization problem. Based on these challenges, problem $(\mathbf{P1})$ is difficult to optimize.

B. Problem Solving

Next, an algorithm is proposed to effectively solve the problem $(\mathbf{P1})$. First, we can consider optimizing $\{t_k[n]\}$,$\{\boldsymbol{q}[n]\}$ and $\{f_k[n]\}$ for any given N, and then obtain the optimal solution of N by binary search [7]. Thus, the optimal solution of problem $(\mathbf{P1})$ is completed.

For any given N, the problem $(\mathbf{P1})$ can be turned into the following feasibility check problem:

$$(\mathbf{P2}): \quad \text{Find } t_k[n],\ \boldsymbol{q}[n] \text{ and } f_k[n] \tag{16}$$

$$s.t. \quad (15a) - (15h). \tag{16a}$$

Assuming that the optimal solution of problem $(\mathbf{P1})$ is N^*, then if problem $(\mathbf{P2})$ is feasible for any given N, then $N^* \leq N$, otherwise $N^* > N$. Thus the problem $(\mathbf{P1})$ can be solved by checking the feasibility of the problem $(\mathbf{P2})$ for any given N and then using binary search.

By analyzing the relationship of the constraints, it can be concluded that problem $(\mathbf{P2})$ is equivalent to the following problem $(\mathbf{P3})$, which is to maximize the total amount of data offloaded by all UEs under a given N as in [7].

$$(\mathbf{P3}): \min_{\{t_k[n]\},\{\boldsymbol{q}[n]\},\{f_k[n]\},\mu\geq 0} \mu \tag{17}$$

$$s.t. \quad \sum_{k=1}^{K}\sum_{n=1}^{N} r_k(\boldsymbol{q}[n])t_k[n] \geq \mu, \tag{17a}$$

$$(15a) - (15c), (15e) - (15h). \tag{17b}$$

Assuming that the optimal solution of problem $(\mathbf{P3})$ is μ^*, it is obvious that problem $(\mathbf{P2})$ is feasible if $\mu^* \geq \sum_{k=1}^{K} D_k$, otherwise, problem $(\mathbf{P2})$ is infeasible.

To this end, we only need to solve the problem $(\mathbf{P3})$. However, since the constraints (17a) and (15h) are non-convex, the problem $(\mathbf{P3})$ is still non-convex. In the following part, we propose an effective iterative algorithm to obtain the optimal solution of problem $(\mathbf{P3})$ by jointly optimizing time allocation $\{t_k[n]\}$, UAV trajectory $\{\boldsymbol{q}[n]\}$ and resource allocation $\{f_k[n]\}$.

1) Time Allocation: Given the UAV trajectory and resource allocation, the problem $(\mathbf{P3})$ can be simplified as

$$(\mathbf{P3.1}): \min_{\{t_k[n]\},\mu\geq 0} \mu \tag{18}$$

$$s.t. \quad (15c), (15e), (15h), (17a). \tag{18a}$$

It is evident that the problem $(\mathbf{P3.1})$ is a linear programming problem, which can be solved by standard convex optimization techniques.

2) Trajectory Optimization: Given resource allocation and time allocation, the problem ($\mathbf{P3}$) can be simplified as

$$(\mathbf{P3.2}): \quad \max_{\boldsymbol{q}[n], \mu \geq 0} \quad \mu \tag{19}$$

$$s.t. \quad (15a), (15b), (15f), (15h), (17a). \tag{19a}$$

Since constraints (15h) and (17a) are both non-convex, the problem ($\mathbf{P3.2}$) is still non-convex. To solve this optimization problem, we adopt the successive convex approximate (SCA) technique and propose an effective iterative algorithm.

Assuming $\{\boldsymbol{q}^{(i)}[n]\}$ represents the local point in the r-th iteration. Next, we will get the approximate constraints of (15h) and (17a). In (15h), $r_k(\boldsymbol{q}[n])$ is a non-convex function with respect to $\boldsymbol{q}[n]$, by checking the first-order Taylor expansion of the convex term $H^2 + \|\boldsymbol{q}[n] - \boldsymbol{z}_k\|^2$ with respect to $\boldsymbol{q}[n]$ at the local point $\boldsymbol{q}^{(i)}[n]$, we have

$$H^2 + \|\boldsymbol{q}[n] - \boldsymbol{z}_k\|^2 \geq A_k^{(i)}[n] + 2(B^{(i)}[n])^T \boldsymbol{q}[n], \tag{20}$$

where $B^{(i)}[n] = \boldsymbol{q}^{(i)}[n] - \boldsymbol{z}_k$, $A_k^{(i)}[n] = H^2 + \|\boldsymbol{q}^{(i)}[n] - \boldsymbol{z}_k\|^2 - 2(B^{(i)}[n])^T \boldsymbol{q}^{(i)}[n]$. Based on the above, we obtain the upper bound of $r_k(\boldsymbol{q}[n])$ as

$$\begin{aligned} r_k(\boldsymbol{q}[n]) &\leq B \log_2\left(1 + \frac{\gamma_k}{A_k^{(i)}[n] + 2(B^{(i)}[n])^T \boldsymbol{q}[n]}\right) \\ &= r_{k,up}^{(i)}(\boldsymbol{q}[n]), \end{aligned} \tag{21}$$

where $r_{(k,up)}^{(i)}(\boldsymbol{q}[n])$ is convex with respect to $\boldsymbol{q}[n]$. Replacing $r_k(\boldsymbol{q}[n])$ in (15h) with $r_{(k,up)}^{(i)}(\boldsymbol{q}[n])$, we get the following approximate convex constraint:

$$\sum_{j=n}^{N} r_{(k,up)}^{(i)}(\boldsymbol{q}[n]) t_k[j] \leq \sum_{j=n+1}^{N} \frac{f_k[j]\delta_t}{s_k}, \forall n = 1, 2, ..., N-1. \tag{22}$$

Even though $r_k(\boldsymbol{q}[n])$ is not convex with respect to $\boldsymbol{q}[n]$, it is convex with respect to $\|\boldsymbol{q}[n] - \boldsymbol{z}_k\|^2$. Since any convex function is globally lower-bounded by its first-order Taylor expansion at any point, the lower bound of $r_k(\boldsymbol{q}[n])$ at $q^{(i)}[n]$ in the i-th iteration can be obtained as

$$\begin{aligned} r_k(\boldsymbol{q}[n]) &\geq C_k^{(i)}[n] - D_k^{(i)}[n](\|\boldsymbol{q}[n] - z_k\|^2 - \|\boldsymbol{q}^{(i)}[n] - z_k\|^2) \\ &= r_{k,low}^{(i)}(\boldsymbol{q}[n]), \end{aligned} \tag{23}$$

where $C_k^{(i)}[n]$ and $D_k^{(i)}[n]$ are the coefficients of Taylor expansion and $r_{k,low}^{(i)}(\boldsymbol{q}[n])$ with respect to $\boldsymbol{q}[n]$ is a concave function. Replacing $r_k(\boldsymbol{q}[n])$ in (17a) with $r_{k,low}^{(i)}(\boldsymbol{q}[n])$, we get the following approximate convex constraint:

$$\sum_{k=1}^{K} \sum_{n=1}^{N} r_{k,low}^{(i)} \boldsymbol{q}[n] t_k[n] \geq \mu. \tag{24}$$

Converted to (22) and (24), the problem (**P3.2**) can be approximately the following convex optimization problem (**P3.3**), which can be solved by standard convex optimization techniques.

$$(\mathbf{P3.3}): \quad \max_{\boldsymbol{q}[n], \mu \geq 0} \quad \mu \tag{25}$$

$$s.t. \quad (15a), (15b), (15f), (22), (24). \tag{25a}$$

Assuming that $\{\boldsymbol{q}^{(i)*}[n]\}$ represents the optimal trajectory of the problem ($\mathbf{P3.3}$) at the point $\{\boldsymbol{q}^{(i)}[n]\}$, the following iterative Algorithm 1 can be obtained and used to solve the problem (**P3.2**). In each iteration, the UAV trajectory is updated to be $\{\boldsymbol{q}^{(i)*}[n]\}$, i.e. $\boldsymbol{q}^{(i+1)}[n] = \boldsymbol{q}^{(i)*}[n]$, and $\{\boldsymbol{q}^{(0)}[n]\}$ represents the initial UAV trajectory. The proposed Algorithm 1 is described in the following table.

Algorithm 1 Solve the problem (**P3.2**) using SCA method

1: Initialization: Given the UAV trajectory $\{\boldsymbol{q}^{(0)}[n]\}$, denote the number of iterations $i = 0$.
2: **repeat**
3: With given $\{\boldsymbol{q}^{(i)}[n]\}$, solve the problem (**P3.3**) to obtain the optimal solution $\{\boldsymbol{q}^{(i)*}[n]\}$.
4: Update $\{\boldsymbol{q}^{(i+1)}[n]\}$=$\{\boldsymbol{q}^{(i)*}[n]\}$
5: Update $i = i + 1$.
6: **until** The optimal value converges within a given threshold or the maximum number of iterations is reached.

3) Resource Allocation: Given the UAV trajectory and time allocation, the problem ($\mathbf{P3}$) can be simplified as

$$(\mathbf{P3.4}): \quad \max_{f_k[n], \mu \geq 0} \quad \mu \tag{26}$$

$$s.t. \quad (15f), (15g), (15h), (17a). \tag{26a}$$

where (15g),(15h),(17a) with respect to $f_k[n]$ are linear constraints, while (15f) with respect to $f_k[n]$ is a convex constraint. Therefore, the problem (**P3.4**) is a convex optimization problem, which can be solved by standard convex optimization techniques.

Based on the above results, an iterative algorithm is proposed by using the block coordinate descent method. In each iteration, time allocation $t_k[n]$, UAV trajectory $\boldsymbol{q}[n]$ and resource allocation $f_k[n]$ are optimized alternately. The proposed Algorithm 2 is described in the following table.

Based on the above analysis, the global time minimization algorithm is proposed by applying binary search. The proposed Algorithm 3 is described in the following table.

IV. Simulation Results and Discussions

In this section, we set specific parameters to carry out simulation experiments and obtain relevant experimental data. We consider a system with $K = 6$ UEs that are randomly distributed within a geographic area of size 1×1 km^2. The UAV is assumed to fly at a fixed altitude $H = 100$ m and its weight is $M = 10$ kg. The duration of each time slot is $\delta_t = 0.5$ s. The data uploading power between each UE and UAV is $P_k = 0.1$ W. We set $E_k^{max} = 30$ J and $E_{UAV}^{max} = 3.6 \times 10^6$ J. The required number of CPU cycles to process each bit is $s_k = 1000$. The channel bandwidth is $B = 10$ MHz. The channel power gain at the reference

Algorithm 2 Solve the problem (**P3**) using the block coordinate descent method

1: Initialization: Given the trajectory $\{\boldsymbol{q}^{(0)}[n]\}$, resource allocation $\{f_k^{(0)}[n]\}$, the number of iterations $i = 0$.
2: **repeat**
3: Solve the problem (**P3.1**) for the given $\{\boldsymbol{q}^{(i)}[n]\}$ and $\{\boldsymbol{q}^{(i)}[n]\}$, and denote the optimal solution as $\{t_k^{(i+1)}[n]\}$.
4: Solve the problem ($\mathbf{P3.2}$) applying Algorithm 1 for the given $\{\boldsymbol{q}^{(i)}[n]\}$, $\{\boldsymbol{q}^{(i)}[n]\}$, and $\{t_k^{(i+1)}[n]\}$, and denote the optimal solution as $\{\boldsymbol{q}^{(i+1)}[n]\}$.
5: Solve the problem ($\mathbf{P3.4}$) for the given $\{\boldsymbol{q}^{(i+1)}[n]\}$ and $\{t_k^{(i+1)}[n]\}$, and denote the optimal solution as $\{f_k^{(i+1)}[n]\}$.
6: Update $i = i + 1$.
7: **until** The increasing of the optimal value is below the threshold ϵ

Algorithm 3 Solve the problem (**P1**) by binary search, i.e. the global time minimization algorithm

1: Initialization: Given the number of time slots N.
2: **repeat**
3: Algorithm 2 is applied to obtain the optimal value μ^* for the given number of time slots N.
4: Judge the size of μ^* and $\sum_{k=1}^{K} D_k$, and then redetermine the value of N by binary search.
5: **until** μ^* for all values less than N is less than $\sum_{k=1}^{K} D_k$.

distance of 1 m is $\rho_0 = -50$ dB and the noise power at each UE is $\sigma^2 = -110$ dB. The effective switched capacitance $r = 1 \times 10^{-27}$. The maximum instantaneous frequency of the server carried by te UAV is $f_{UAV}^{max} = 0.1$ MHz.

Fig. 2 depicts the UAV flight trajectory optimization process under different number of time slots. First, in each optimized trajectory, the UAV is always trying to fly close to the UEs, because close proximity leads to faster data transmission rate and better communication quality. Second, when the number of time slots is fixed, the optimized trajectory of the UAV gradually approaches each UE from the initial circle. In order to maximize the total amount of offloading tasks, its final trajectory tends to converge between the positions of some UEs. Third, more time slots will promote the optimized UAV trajectory to expand, so as to achieve a larger coverage area.

Fig. 3 shows the time allocation when $N = 141$. Observed that different UEs cannot offload tasks at the same time due to the TDMA scheme. In each time slot, only when the UAV is relatively close to the UE, the offloading time proportion of the UE is the highest. When the UAV is far away from the device, the offloading time proportion of the UE gradually decreases and approaches zero. The reason is that the short distance brings faster transmission rate and better channel quality.

Fig. 4 shows the computing resource allocation when $N = 141$. Each UE will get continuous computing resources allocation, and after the UAV receives the data of the UE,

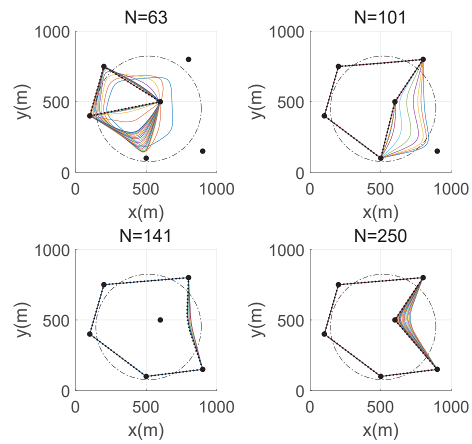

Fig. 2. The optimization process of UAV trajectory. Six solid black dots represent six UEs, dotted line represents the final optimal trajectory, solid lines represent trajectories in the process of optimization iteration and dash-dot-lines represent the initial circular trajectory.

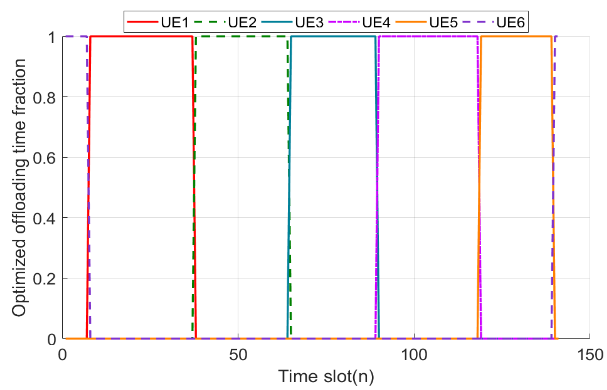

Fig. 3. Optimized offloading time allocation with $N = 141$ among different UEs

it will then allocate more computing resources to the corresponding UE, which is shown in the diagram that every curve has a peak, and UAV will allocate less computing resources to the UE in most of the time outside the peak.

Fig. 5 shows the change of the UAV flight velocity. First, for the fixed number of time slots, we can see clearly that when the UAV approaches each UE, it will greatly reduce its velocity, and keep close to the velocity of zero which allows the UAV to hover over the UE for a long time and transmit data at a high rate. After leaving one UE, the UAV will fly at the maximum velocity, so it can get to the next UE as quickly as possible. Second, as the number of time slots increases, the frequency of UAV flying at low velocity increases. This is because the UAV can cover a larger area and be close to more UEs for task offloading with more given time.

Fig. 6 shows the relationship between the iterative optimized task bits and the number of iterations. In particular, the dash-dot-line represents the maximum amount of offloading tasks obtained from the outer iteration. No matter the inner or the outer layers, with the increase of the number of iterations,

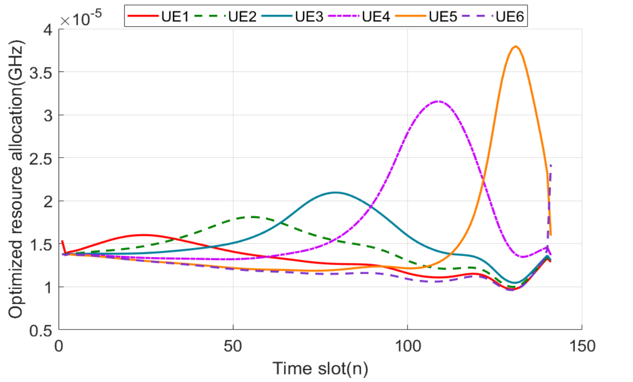

Fig. 4. Optimized resource allocation with $N = 141$ among different UEs

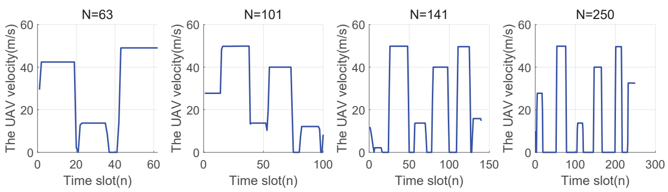

Fig. 5. UAV flight velocity

the optimized offloading amount will continue to show an upward trend and eventually converge. This also verifies the convergence and effectiveness of the proposed algorithm.

Finally, we compare the performance of the proposed co-optimization algorithm with that of the other schemes.

1) *Proposed joint optimization algorithm:* The proposed joint optimization algorithm is used to optimize time, resource allocation and UAV flight trajectory.
2) *Circular UAV flight trajectory:* Optimize time and resource allocation with the circular UAV flight trajectory.
3) *Equal computing resource allocation:* Optimize time allocation with the simple circular UAV flight trajectory and the equal computing resource allocation.

Fig. 7 shows the relationship between the minimum task completion time $T = N\delta_t$ of the three schemes and the total input bits $\sum_{k=1}^{K} D_k$. First, the minimum completion time for all three schemes increases as the total amount of input data increases. The performance of the proposed joint optimization algorithm is obviously better than the other two, especially when the amount of input data gradually increases, the gap between the proposed algorithm and the other two schemes is getting bigger. Second, the performance of the algorithm becomes better with increasing optimization variables. Third, the UAV trajectory, compared with the other two variables, has a more significant effect on the overall performance of the algorithm.

V. Conclusion

In this paper, we study the time minimization problem in UAV-enabled MEC system. With the objective to minimize the task completion time, we jointly optimize the time, computing resources and trajectory of UAV. By using the block coordinate descent method and the convex optimization

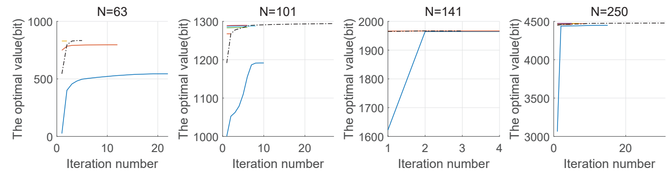

Fig. 6. Iterative optimization process of offloading task amount. Each solid line represents the maximum amount of offloading task obtained by the inner iteration. The blue solid line represents the inner optimization result in the first outer iteration.

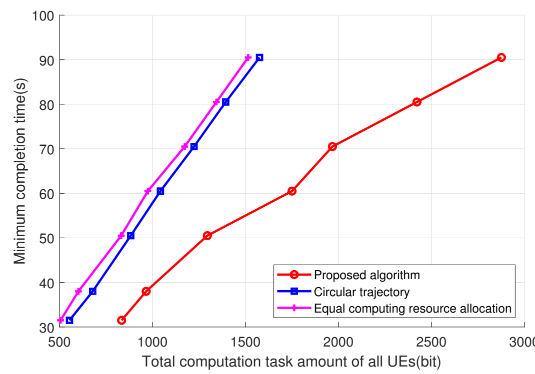

Fig. 7. Comparison of algorithm performance

technique, we propose an efficient iterative algorithm with ensured convergence. Extensive simulations are conducted and performance evaluations show that the joint optimization of multiple variables can greatly reduce the task completion time of UAV.

VI. Acknowledgment

This work has been partly supported by NSFC (No. 62071105)

References

[1] S. Jeong, O. Simeone, J. Kang, "Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049-2063, 2018.

[2] M. Hua, Y Wang, C Li, Y Huang, L Yang, "UAV-Aided Mobile Edge Computing Systems With One by One Access Schem," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 569-572, 2019.

[3] C. Zhan, H. Hu, X. Sui, Z. Liu, D. Niyato, "Completion Time and Energy Optimization in the UAV-Enabled Mobile-Edge Computing System," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7808-7822, 2020.

[4] J. Zhang, *et al*, "Joint Resource Allocation for Latency-Sensitive Services Over Mobile Edge Computing Networks With Caching," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4283-4294, 2019.

[5] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, X. Shen, "Energy-Efficient UAV-Assisted Mobile Edge Computing: Resource Allocation and Trajectory Optimizations," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3424-3438, 2020.

[6] Y. Mao, J. Zhang, K. Letaief, "Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590-3605, 2016.

[7] X. Cao, J. Xu, and R. 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*, 2018.