Completion time minimization for UAV-assisted mobile-edge computing systems

Xu, Y., Zhang, T.K., Loo, Jonathan ORCID logoORCID: https://orcid.org/0000-0002-2197-8126, Yang, D.Y. and Xiao, L. (2021) Completion time minimization for UAV-assisted mobile-edge computing systems. IEEE Transactions on Vehicular Technology. ISSN 0018-9545

This is the Accepted Version of the final output.

Completion Time Minimization for UAV-Assisted Mobile-Edge Computing Systems

Yu Xu, *Graduate Student Member, IEEE*, Tiankui Zhang, *Senior Member, IEEE*, Jonathan Loo,
Dingcheng Yang, *Member, IEEE*, and Lin Xiao, *Member, IEEE*

Abstract—The explosive computation demands in the Internet of Things (IoT) have triggered the research interests on unmanned aerial vehicle (UAV) assisted mobile-edge computing (MEC) systems even though there are still many challenges, such as computing delay requirement, multi-UAV cooperation, and resource management. This letter focuses on the computing delay issue in MEC systems assisted by multiple UAVs with the goal of task completion time minimization. In particular, both the partial offloading and binary offloading modes are considered by jointly optimizing time slot size, terminal devices scheduling, computation resource allocation, and UAVs' trajectories. Particularly, an non-LoS channel model is adopted for UAV-ground communication. To handle the formulated problems, we develop alternating optimization algorithms by invoking the successive convex approximation method, Karush-Kuhn-Tucker conditions and penalized method. Numerical results show that the completion time is significantly decreased by the proposed algorithms.

Index Terms—Internet of Things, mobile edge computing, trajectory optimization, UAV communication.

I. INTRODUCTION

Mobile edge computing (MEC) has emerged as a promising technology to cater to the explosive computation demands of many computation-hungry and latency-sensitive applications in Internet of things (IoT), such as augmented reality and automatic pilot [1]. In MEC networks, two offloading modes, partial offloading mode and binary computation mode, are usually adopted according to the application scenario [2]. For the partial offloading mode, the computation task is divided into several parts. One part is computed locally, while the other parts are computed by offloading to MEC server(s). For example, in high computation requirement scenarios such as facial recognition, the lower computation part can be computed locally and the higher computation part is computed by offloading. For the binary offloading mode, the computation task is computed locally or offloaded altogether. For example, if performing the channel state information estimation, the collected raw data samples have to be computed as a whole for ensuring the estimation accuracy [3].

Although a lot of attention has been focused on MEC, the latency in MEC is still an opening problem that are investigated in few works [4]–[7]. Specifically, the authors in [4] and [5] study

Y. Xu and T. Zhang are with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: {xuyu56, zhangtiankui}@bupt.edu.cn). (Corresponding author: Tiankui Zhang);

Jonathan Loo is with the School of Computing and Engineering, University of West London, London W5 5RF, U.K. (e-mail: jonathan.loo@uwl.ac.uk);

D. Yang and L. Xiao are with the Information Engineering School, Nanchang University, Nanchang 330031, China (e-mail: {yangdingcheng, xiaolin}@ncu.edu.cn).

the latency minimization problem in MEC by communication and computation resource allocation. Next, they investigate the latency minimization of all mobile devices in collaboration between cloud computing and edge computing scenarios [6]. The works [4]–[6] only consider the partial offloading mode. To fully utilize the spectrum resource, the work in [7] studies the latency in full-duplex MEC systems. However, the devices in [7] are unable to perform local computing. Note that the MEC servers in these existing latency-related works are deployed fixedly, which leads to many limitations in terms of flexibility. Also, the computation resource in these networks is unavailable in practice for the terminal devices (TDs) located in remote place or "dead" zone, such as mountainous areas.

With the flexibility and low cost, unmanned aerial vehicle (UAV) has applied in varieties of communication scenarios, such as NOMA-UAV systems [8] [9]. Also, UAV has the potential to provide computation for the remote devices [1], which called as UAV-assisted/enabled MEC. By exploiting the mobility, UAV can be utilized to shorten the transmit latency [10]–[13], which provides the potential to tackle with the latency issue in MEC systems mentioned above. Besides the communication time cost, as discussed in [10]–[13], UAV-assisted MEC incurs an additional time overhead on computation that has to be carefully designed since the UAV has limited computation capability. In addition, note that the deterministic LoS channel following the free-space pathloss model links are used as the UAV-ground channel model for the literatures mentioned above, which is practically inaccurate in urban/suburban environment. Moreover, in UAV-assisted MEC systems, the single one UAV case is facing the issue of weak payload, limited coverage and lacking computation capacity. Thus, it is labored even infeasible for one UAV to satisfy the specific scenario with large-scale users or huge amount of tasks.

Motivated by these, we study the latency problem in MEC systems assisted by multiple UAVs in this paper. To be specific, we consider the scenario where multiple UAVs equipped with MEC servers aim to help TDs for computing. Each TD has a certain number of computation requirement that is computed locally and/or offloaded to the UAVs for execution. To the best of our knowledge, the latency in UAV-assisted MEC has not yet been well studied. In summary, the main contributions of this work are presented as follows,

- We propose a new framework of UAV-assisted MEC system to study the latency issue, in which multiple UAVs are adopted to enhance the computation capacity. Particularly, both the partial offloading and binary offloading are considered. To fit in reality, we discard the usual simplified LoS links, and the UAV-ground links are modeled by non-LoS (NLoS) channel model that consists of large-scale fading and small-scale fading.
- We formulate a novel task completion time minimization problem, by jointly optimizing the time slot size, TDs
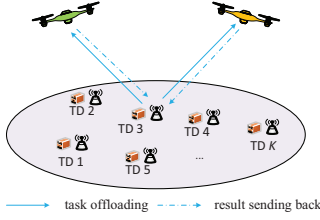
Fig. 1. Illustration of the multiple UAV-assisted MEC systems.

scheduling, computation resource allocation, and UAVs' trajectories. Different from the typical setup of fixed number of time slots and constant length of each time slot, we propose a dynamic method to determine the total number of time slots and the length of each time slot.
- We introduce several techniques, including successive convex approximation (SCA) method, Karush-Kuhn-Tucker (KKT) conditions, penalized method, and bisection method, to optimize the formulated problems according to their characteristics. After that, we propose the effective alternating optimization (AO) algorithms followed by convergence and complexity analysis. Finally, numerical result is presented to validate that our proposed algorithms significantly decrease the completion time.

II. SYSTEM MODEL

We consider the UAV-assisted MEC systems that consist of a set $\mathcal{K} \triangleq \{1, 2, ..., K\}$ of ground TDs, in which multiple mobile UAVs, denoted by $\mathcal{M} \triangleq \{1, 2, ..., M\}$, are deployed to assist the TDs to compute the task during a period called as task completion time, as shown in Fig. 1. For ease of discussion, the period is divided into $N > 0$ time slots with each duration of $T[n]$, $n \in \mathcal{N} \triangleq \{1, 2, ..., N\}$. Note that the value of N is a well-chosen deterministic value. Evidently, the task completion time is determined by $\sum_{n=1}^{N} T[n]$. In a 3D Cartesian coordinate system, by giving any time instant t with $0 \leq t \leq \sum_{n=1}^{N} T[n]$, the horizonal locations for TD $k \in \mathcal{K}$ and UAV m are denoted by $\mathbf{w}_k$ and $\mathbf{q}_m(t) = \mathbf{q}_m[n]$, respectively, with $t = \sum_{i=1}^{n} T[i]$, and $\mathbf{w}_k$, $\mathbf{q}_m[n] \in \mathbb{R}^{2 \times 1}$. Note that the horizonal location of each TD can be always known via their GPS information. Denote H as the UAVs' fixed altitude. We assume that the battery power of each UAV is adequate during the mission. The distance between the UAV m and TD k in time slot n can be denoted by $d_{km}[n] = \sqrt{H^2 + ||\mathbf{q}_m[n] - \mathbf{w}_k||^2}$. We consider quasi-static fading channels, and the instantaneous channel gains between the UAVs and TDs are modeled as

$$h_{km}[n] = \sqrt{\beta_0 d_{km}^{-\alpha}[n]} g_{km}[n], \forall k, n, \quad (1)$$

where β_0 is the channel gain at the reference distance $d_0 = 1$ m, $\alpha \geq 2$ denotes the path loss exponent, $g_{km}[n]$ is the complex valued random variable with $\mathbb{E}[|g_{km}[n]|^2] = 1$ accounting for the small-scale fading that is modeled by Rician fading [14], given as $g_{km}[n] = \sqrt{\frac{K_{km}[n]}{K_{km}[n]+1}} g + \sqrt{\frac{1}{K_{km}[n]+1}} \tilde{g}$, where g denotes the LoS channel component with $|g| = 1$, $\tilde{g}$ is a zero-mean unit-variance circularly symmetric complex Gaussian variable that denotes the random scattered component, and $K_{km}[n]$ is the Rician factor.

In order to avoid the complex channel state estimation and feedback, we consider a time division scheme for each TD offloading task to UAVs in turns in each time slot. Let P_k

and B denote the maximum transmit power of TDs and total bandwidth of the system, respectively. In particular, for partial offloading mode, the task of TD k in current time slot is divided into $M + 1$ sections, among which one section is computed locally and the other sections are offloaded to UAVs over M orthogonal sub-carriers with a constant transmit power $P_{kp} = P_k/M$. Let $B_p = B/M$ denote as the bandwidth of each sub-carrier. For binary offloading mode, the task of each TD is executed either at locally or offloaded to no more than one UAV. Thus, the transmit power and available bandwidth in binary offloading mode are given by $P_{kb} = P_k$ and $B_b = B$, respectively. Define σ_0^2 as the noise power at the receiver. Hence, the offloading rate from TD k to the UAV m in time slot n is expressed as

$$\hat{R}_{kmj}[n] = B_j \log_2 \left(1 + \frac{P_{kj}|h_{km}[n]|^2}{\sigma_0^2}\right), j \in \{p, b\}. \quad (2)$$

Note that $\hat{R}_{kmj}[n]$ is intractable due to the random variable $h_{km}[n]$ in (2). One feasible method to tackle this is to adopt the average offloading rate, i.e., $\mathbb{E}[\hat{R}_{kmj}[n]]$, which can be further tackled by the Jensen's inequality. In order to close to reality and embody the effect caused by small-scale fading, we adopt an approximation form by logistic regression method to achieve the effective fading power based on [14], thus the offloading rate of $\hat{R}_{kmj}[n]$ is approximately denoted as $R_{kmj}[n]$, given by

$$R_{kmj}[n] = B_j \log_2 \left(1 + \frac{P_{kj} \gamma_0 v_{km}[n]}{(H^2 + ||\mathbf{q}_m[n] - \mathbf{w}_k||^2)^{\alpha/2}}\right), \quad (3)$$

where $\gamma_0 = \beta_0/\sigma_0^2$, $v_{km}[n] = C_1 + \frac{C_2}{1+e^{-(B_1+B_2 u_{km}[n])}}$, where B_1, B_2, C_1 and C_2 are coefficients related to the logistic model, and $u_{km}[n] = \frac{H}{\sqrt{H^2 + ||\mathbf{q}_m[n] - \mathbf{w}_k||^2}}$ is the sine of the elevation angle. The derivation for (3) is detailedly referred in [14].

It is assumed that the total computation task/workload of each TD in bits is D_{req}, and let $D_k[n]$ denote the number of allocated task for TD k in time slot n, satisfying $\sum_{n=1}^{N} D_k[n] = D_{req}$. Let c_k and F_k denote the the required CPU cycles for computing each one bit and the maximum CPU frequency, respectively, at TD k. Denote $\rho_{km}[n]$ as the ratio of the task offloading from TD k in slot n to UAV m, thus the ratio of the task computed locally for TD k in time slot n is denoted by $1 - \sum_{m=1}^{M} \rho_{km}[n]$. Therefore, based on [15], the time for local computing for TD k in time slot n is denoted as

$$t_k^L[n] = \frac{(1 - \sum_{m=1}^{M} \rho_{km}[n]) D_k[n] c_k}{F_k}, \forall k, n. \quad (4)$$

It is evident that for TD k, the amount of task $\rho_{km}[n] D_k[n]$ is executed at the UAV m in time slot n. Let $f_{mk}[n] \geq 0$ denote the CPU frequency of the UAV m to execute computation for TD k in time slot n, which is constrained by the maximum CPU frequency F_m^{max}, i.e., $\sum_{k=1}^{K} f_{mk}[n] \leq F_m^{max}, \forall m, n$. Specifically, the time allocated to schedule TD k in time slot n for offloading to UAV m is denoted by $t_{kmj}^O[n]$, which satisfies $t_{kmj}^O[n] \geq \frac{\rho_{km}[n] D_k[n]}{R_{kmj}[n]}$. Moreover, let c_m and $t_{km}^U[n]$ denote the required CPU cycles for computing each one bit and the time for computing the task from TD k at UAV m, respectively. We assume that the arrived task at the UAV m in current time slot is stored in the buffer and then fully computed in the next time slot. Therefore, we obtain the following constraint,

$$t_{km}^U[n] \geq \frac{\rho_{km}[n-1] D_k[n-1] c_m}{f_{km}[n]}, \forall k, m, n \geq 2. \quad (5)$$

The time for results sending back is omitted due to the much smaller size than the offloaded task [15]. Note that local computing, task offloading and UAVs' computing can be performed parallelly in our design. As a result, the duration of

time slot n should satisfy the following constraint,

$$T[n] \geq \max\left\{\max_{\forall k} t_k^{\mathrm{L}}[n], \max_{\forall m}\sum_{k=1}^{K} t_{kmj}^{\mathrm{O}}[n], \max_{\forall k,m} t_{km}^{\mathrm{U}}[n]\right\}. \quad (6)$$

Note that both the partial offloading mode and binary offloading mode are considered for TDs. For the partial offloading, each TD in time slot n is able to offload discretionary amount of task to the UAVs, i.e., $\rho_{km}[n] \in [0,1]$. While for the binary offloading, the task in time slot n is computed either at the TD or at no more than one UAV, i.e., $\rho_{km}[n] \in \{0,1\}$ and $\sum_{m=1}^{M}\rho_{km}[n] \leq 1$.

To facilitate the assumption that the UAVs' locations are quasi-static within any time slot, the length of each time slot is subject to the constraint $T[n] \leq \Delta_m$, where Δ_m is the allowable maximum length that is predefined.

We formulate the problem that minimizes the task completion time for partial offloading mode, which is expressed as

$$(\mathbf{P}1): \min_{\mathcal{Z}}\sum_{n=1}^{N} T[n]$$

$$\text{s.t. } T[n] \geq \frac{(1-\sum_{m=1}^{M}\rho_{km}[n])D_k[n]c_k}{F_k}, \forall k,n, \quad (7a)$$

$$T[n] \geq \sum_{k=1}^{K}\frac{\rho_{km}[n]D_k[n]}{R_{kmp}[n]}, \forall m,n, \quad (7b)$$

$$T[n] \geq \frac{\rho_{km}[n-1]D_k[n-1]c_m}{f_{km}[n]}, \forall k,m,n \geq 2, \quad (7c)$$

$$0 \leq \rho_{km}[n] \leq 1, \sum_{m=1}^{M}\rho_{km}[n] \leq 1, \forall k,m,n, \quad (7d)$$

$$\sum_{k=1}^{K} f_{mk}[n] \leq F_m^{\max}, \forall m,n, \quad (7e)$$

$$D_k[n] \geq 0, \sum_{n=1}^{N} D_k[n] \geq D_{\mathrm{req}}, \forall k, \quad (7f)$$

$$0 \leq T[n] \leq \Delta_m, f_{mk}[n] \geq 0, \forall k,m,n, \quad (7g)$$

$$\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\| \leq V_{\max}T[n], \forall m,n, \quad (7h)$$

$$\|\mathbf{q}_m[n] - \mathbf{q}_l[n]\|^2 \geq d_m^2, \forall m,l \in \mathcal{M}, n, m \neq l, \quad (7i)$$

$$\mathbf{q}_m[1] = \mathbf{q}_m[N], \quad (7j)$$

where $\mathcal{Z} \triangleq \{T[n], D_k[n], \rho_{km}[n], f_{mk}[n], \mathbf{q}_m[n]\}$, (7f) denotes the minimum task requirement for TD k. The constraint in (7h) indicates that the UAVs' flight distance in one time slot cannot exceed a maximum distance, and $V_{\max}$ denotes the maximum speed of each UAV. Constraint (7i) denotes that any two UAVs always need to keep a minimum secure distance d_m for collision avoidance. The constraint in (7j) means that the UAVs need to fly back to their initial locations after task completion for the sake of recycling. Note that we can readily apply this work to the case with different initial and final locations without any impact on the following analysis. It is worth noting that (7a)-(7c) are equivalent to (6). Evidently, problem $\mathbf{P}1$ is non-convex due to the non-convexity in (7a)-(7c) and (7i).

Similarly, the completion time minimization problem for binary offloading mode is formulated as

$$(\mathbf{P}2): \min_{\mathcal{Z}}\sum_{n=1}^{N} T[n]$$

$$\text{s.t. } (7a), (7c)\text{-}(7j),$$

$$T[n] \geq \sum_{k=1}^{K}\frac{\rho_{km}[n]D_k[n]}{R_{kmb}[n]}, \forall m,n, \quad (8a)$$

$$\rho_{km}[n] \in \{0,1\}, \forall k,m,n. \quad (8b)$$

Evidently, problem $\mathbf{P}2$ is also non-convex due to the non-convexity in the constraints (7a), (7c), (7i), (8a), and (8b).

III. PROPOSED JOINT OPTIMIZATION ALGORITHM

In this section, we propose the efficient AO algorithms to solve the problem $\mathbf{P}1$ and $\mathbf{P}2$. For clarity, we first discuss the partial offloading mode. Then the analysis for binary mode is presented. Finally, we present the proposed algorithms.

A. Partial Offloading Mode

For the partial offloading, we solve the problem $\mathbf{P}1$ by tackling with the non-convexity in (7a)-(7c) and (7i). Define $\tilde{\mathcal{Z}}_1 \triangleq \{\rho_{km}[n], D_k[n]\}$, $\tilde{\mathcal{Z}}_2 \triangleq \{f_{km}[n]\}$, and $\tilde{\mathcal{Z}}_3 \triangleq \{\mathbf{q}_m[n]\}$. Then, the problem is decomposed and solved by three steps in an alternating manner.

1) Step 1: Optimizing problem $\mathbf{P}1$ by fixed $\tilde{\mathcal{Z}}_2$ and $\tilde{\mathcal{Z}}_3$.
Lemma 1: The non-convex constraint (7a) can be converted as convex form that is expressed as

$$T[n]s_k \geq \frac{(\tilde{\rho}_k[n]+D_k[n])^2}{4} + \frac{(\tilde{\rho}_k^r[n]-D_k^r[n])^2}{4}$$
$$- \frac{(\tilde{\rho}_k[n]-D_k[n])(\tilde{\rho}_k^r[n]-D_k^r[n])}{2}, \quad (9)$$

where $s_k = F_k/c_k$, $\tilde{\rho}_k[n] \geq 1 - \sum_{m=1}^{M}\rho_{km}[n]$ is an introduced auxiliary variable, $\tilde{\rho}_k^r[n]$ and $D_k^r[n]$ are the given feasible points at the rth iteration.

Proof: Considering $\tilde{\rho}_{km}[n]D_k[n]$ of the right-hand side of (7a), it can be rewritten as difference-of-convex form, i.e.,

$$\tilde{\rho}_{km}[n]D_k[n] = \frac{(\tilde{\rho}_{km}[n]+D_k[n])^2}{4} - \frac{(\tilde{\rho}_{km}[n]-D_k[n])^2}{4}. \quad (10)$$

Next, we adopt SCA method to deal with (10). Define $\mathbf{z} \triangleq \{\tilde{\rho}_{km}[n], D_k[n]\}$, and $f(\mathbf{z}) \triangleq (\tilde{\rho}_{km}[n] - D_k[n])^2$. Obviously, $f(\mathbf{z})$ is convex *w.r.t.* $\mathbf{z}$. Thus, we can obtain its lower-bound function by the first-order Taylor expansion at $\mathbf{z}^r$, expressed as

$$f^{lb}(\mathbf{z}^r,\mathbf{z}) = f(\mathbf{z}^r) + 2\Re\{\nabla f^H(\mathbf{z}^r)(\mathbf{z}-\mathbf{z}^r)\}$$
$$= (\tilde{\rho}_{km}[n]-D_k[n])(\tilde{\rho}_{km}^r[n]-D_k^r[n])$$
$$- (\tilde{\rho}_{km}^r[n]-D_k^r[n])^2. \quad (11)$$

By substituting $f(\mathbf{z})$ with $f^{lb}(\mathbf{z}^r,\mathbf{z})$ into (10) and via minor transformation, (9) is obtained. This proof is completed. ∎

Remark 1: Lemma 1 provides a technique for approximate convex variant. When the points $\tilde{\rho}_{km}^r[n]$ and $D_k^r[n]$ become stable in iterations, the inequality $\tilde{\rho}_{km}[n] \geq 1 - \sum_{m=1}^{M}\rho_{km}[n]$ and the constraint (9) must hold with equality. Thus, at least a sub-optimal solution is guaranteed to achieve in this step.

Let $\nu_{kmn}^{pr} = \left(B_p\log_2\left(1+\frac{P_{kp}\gamma_0 v_{km}^r[n]}{H^2+\|\mathbf{q}_m[n]-\mathbf{w_k}\|^{\alpha/2}}\right)\right)^{-1}$, $\varsigma_{kmn}^r = f_{km}^r[n]/c_m$, and $\chi_{km}^r[n] = \rho_{km}^r[n] - D_k^r[n]$. According to Lemma 1, the constraints (7b) and (7c) can be approximately converted into the following convex forms, expressed as

$$4T[n] \geq \sum_{k=1}^{K}\sum_{m=1}^{M}\nu_{kmn}^{pr}\left((\rho_{km}[n]+D_k[n])^2 + (\chi_{km}^r[n])^2\right.$$
$$\left. - 2(\rho_{km}[n]-D_k[n])\chi_{km}^r[n]\right), \quad (12)$$

$$4T[n]\varsigma_{kmn}^r \geq (\rho_{km}[n-1]+D_k[n-1])^2 + (\chi_{km}^r[n-1])^2$$
$$- 2(\rho_{km}[n-1]-D_k[n-1])\chi_{km}^r[n-1]. \quad (13)$$

Accordingly, by substituting (7a)-(7c) with (9), (12) and (13), the problem $\mathbf{P}1$ in Step 1 becomes convex that is efficiently solved by the convex optimization tools.

2) Step 2: Optimizing problem $\mathbf{P}1$ by fixed $\tilde{\mathcal{Z}}_1$ and $\tilde{\mathcal{Z}}_3$
In this step, the problem $\mathbf{P}1$ is convex *w.r.t.* $f_{km}[n]$. Therefore, via the KKT conditions, we obtain the closed-form solution of $f_{km}[n]$ in Lemma 2.

Lemma 2: The optimal solution of $f_{km}[n]$ by solving the problem **P**1 is given by

$$f_{km}^*[n] = \begin{cases} 0, & \text{if } \rho_{km}^r[n-1] = 0, \\ \frac{\sqrt{\rho_{km}^r[n-1]D_k^r[n-1]}F_m^{max}}{\sum_{k=1}^K \sqrt{\rho_{km}^r[n-1]D_k^r[n-1]}}, & \text{otherwise.} \end{cases} \quad (14)$$

Proof: For the case of $\rho_{km}^r[n-1] = 0$, namely there is no task bits from TD k at the buffer of UAV m in current time slot n, thus the UAV m dose not allocate any computation resource to TD k, i.e., $f_{km}^*[n] = 0$. For $\rho_{km}^r[n-1] \neq 0$, define $\Xi \triangleq \{T[n], f_{km}[n], \lambda_{kmn}, \mu_{mn}\}$, the Lagrange function of **P**1 is expressed as $L(\Xi) = \sum_{n=1}^N T[n] + \sum_{k=1}^K \sum_{m=1}^M \sum_{n=2}^N \lambda_{kmn} \left(\frac{\rho_{km}[n-1]D_k[n-1]c_m}{f_{km}[n]} - T[n] \right) + \sum_{m=1}^M \sum_{n=2}^N \mu_{mn} \left(\sum_{k=1}^K f_{mk}[n] - F_m^{max} \right)$, with $\lambda_{kmn}, \mu_{mn} \geq 0$. As the problem **P**1 is convex, the Slaters condition is satisfied. By applying KKT conditions, we obtain $\frac{\partial L(\Xi)}{\partial f_{km}[n]} = 0$, $\lambda_{kmn}(\frac{\rho_{km}[n-1]D_k[n-1]c_m}{f_{km}[n]} - T[n]) = 0$, $\mu_{mn}(\sum_{k=1}^K f_{mk}[n] - F_m^{max}) = 0$. Based on these equations, the final result $f_{km}^*[n]$ is derived. This proof is completed. ∎

Remark 2: Lemma 2 reveals that the optimal CPU frequency for executing the task of TD k at the UAV m is absolutely determined by the offloaded task in the last time slot. Specifically, the larger value of the offloaded task bits of TD k is, the more frequency is supposed to be allocated to execute for TD k.

*3) Step 3: Optimizing problem **P**1 by fixed $\tilde{\mathcal{Z}}_1$ and $\tilde{\mathcal{Z}}_2$*

For fixed $\tilde{\mathcal{Z}}_1$ and $\tilde{\mathcal{Z}}_2$, all the constraints of problem **P**1 are affine besides the constraint (7b) and (7i). Note that constraint (7b) is a quite complex function w.r.t. $\mathbf{q}_m[n]$. First, we convert it into more tractable forms based on Lemma 3.

Lemma 3: By introducing the nonnegative auxiliary parameters π_{kmn}, x_{kmn}, y_{kmn}, s_{kmn} and v_{kmn}, the constraint (7b) is equivalent to the following constraints,

$$T[n] \geq \sum_{k=1}^K \frac{\rho_{km}[n]D_k[n]}{\pi_{kmn}}, \quad (15a)$$

$$\pi_{kmn} \leq B_p \log_2 \left(1 + \frac{P_{kp}\gamma_0 C_1 + \frac{P_{kp}\gamma_0 C_2}{x_{kmn}}}{y_{kmn}^{\alpha/2}} \right), \quad (15b)$$

$$x_{kmn} \geq 1 + e^{-s_{kmn}}, \quad (15c)$$

$$y_{kmn} \geq ||\mathbf{q}_m[n] - \mathbf{w_k}||^2 + H^2, \quad (15d)$$

$$s_{kmn} \leq B_1 + B_2 u_{km}[n], \quad (15e)$$

Proof: In order to prove the equivalence between (7b) and (15), we just need to show that the inequalities in (15) meet with equalities. For (15b), it has to meet with equality, otherwise we can increase the value of π_{kmn}, thus further decrease $T[n]$, which is expected. Similarly, it is known that the rest of inequalities in (15) also must hold with equalities and the details are omitted here. Hence, the constraint (7b) is equivalent (15). This proof is completed. ∎

Note that constraints (15b) and (15e) is still non-convex. Nevertheless, the right-hand side of (15b) is a joint convex function w.r.t. x_{kmn} and $y_{kmn}^{\alpha/2}$, and $u_{km}[n]$ in (15e) is convex w.r.t. $||\mathbf{q}_m[n] - \mathbf{w_k}||^2 + H^2$, which motivates us to invoke the SCA method to tackle with them. Specifically, for given feasible points x_{kmn}^r and $y_{kmn}^{\alpha/2,r}$, we get the lower-bound function of the right-hand side of (15b) as follows,

$$\varphi_{km}^{lb}[n] = A_{1km}^r[n] - A_{2km}^r[n](x_{kmn} - x_{kmn}^r)$$
$$- A_{3km}^r[n]\left(y_{kmn}^{\alpha/2} - y_{kmn}^{\alpha/2,r} \right), \quad (16)$$

where $A_{1km}^r[n] = B_p \log_2 \left(D_{1km}^r[n] + \frac{D_{2km}^r[n]}{x_{kmn}^r} \right)$, $A_{2km}^r[n] = $

$\frac{B_p \log_2(e) D_{2km}^r[n]}{x_{kmn}^r \left(D_{1km}^r[n] x_{kmn}^r + D_{2km}^r[n] \right)}$, $D_{1km}^r[n] = 1 + P_{kp}\gamma_0 C_1 / y_{kmn}^{\alpha/2,r}$, $D_{2km}^r[n] = P_{kp}\gamma_0 C_2 / y_{kmn}^{\alpha/2,r}$, $A_{3km}^r[n] = \frac{B_p \log_2(e) E_{km}^r[n]}{y_{kmn}^{\alpha/2,r} \left(y_{kmn}^{\alpha/2,r} + E_{km}^r[n] \right)}$, and $E_{km}^r[n] = P_{kp}\gamma_0 C_1 + P_{kp}\gamma_0 C_2 / x_{kmn}^r$. Then, we obtain the lower bound of $u_{km}[n]$ with $\mathbf{q}_m^r[n]$, given as

$$u_{km}^{lb}[n] = u_{1km}^r[n]$$
$$- u_{2km}^r[n] \left(||\mathbf{q}_m[n] - \mathbf{w_k}||^2 - ||\mathbf{q}_m^r[n] - \mathbf{w_k}||^2 \right), \quad (17)$$

where $u_{1km}^r[n] = \frac{H}{\sqrt{H^2 + ||\mathbf{q}_m^r[n] - \mathbf{w_k}||^2}}$, and $u_{2km}^r[n] = \frac{H}{2(H^2 + ||\mathbf{q}_m^r[n] - \mathbf{w_k}||^2)^{3/2}}$. As a result, the constraints (15b) and (15e) are respectively transformed as

$$\pi_{kmn} \leq \varphi_{km}^{lb}[n], \forall k, m, n. \quad (18)$$

$$s_{kmn} \leq B_1 + B_2 u_{km}^{lb}[n]. \quad (19)$$

In (7i), we find that $||\mathbf{q}_m[n] - \mathbf{q}_l[n]||^2$ is jointly convex w.r.t. $\mathbf{q}_m[n]$ and $\mathbf{q}_l[n]$, which motivates us to gain its lower bounded function by using the first order Taylor expansion at given local points $\mathbf{q}_m^r[n]$ and $\mathbf{q}_l^r[n]$, denoted by

$$\psi_{m,l}^{lb}[n] = ||\mathbf{q}_m^r[n] - \mathbf{q}_l^r[n]||^2 +$$
$$2 \left(\mathbf{q}_m^r[n] - \mathbf{q}_l^r[n] \right)^\dagger \left(\mathbf{q}_m[n] + \mathbf{q}_l^r[n] - \mathbf{q}_l[n] - \mathbf{q}_m^r[n] \right), \quad (20)$$

where $(\cdot)^\dagger$ denotes the transpose operation.

Therefore, by substituting (7b) with (15a), (15c), (15e), (18) and (19), and (7i) with (20), the problem **P**1 in Step 3 is convex that can be efficiently solved by the convex optimization tools.

Remark 3: The main thought of trajectory optimization in Step 3 is that the primal non-convex problem is first approximately converted into a convex form by introducing the auxiliary variables. Then, the trajectory is updated in each iteration based on SCA method, until a stable local optimal point is reached.

B. Binary Offloading Mode

For the binary offloading mode, i.e., **P**2, it belongs to a mixed integer non-convex problem. In order to derive the solution, a penalized algorithm that includes double loops is proposed.

Lemma 4: With given penalty parameter ϱ and feasible point $\rho_{km}^r[n]$, the solution to problem **P**2 for the binary offloading is obtained by solving the following penalized problem **P**2.1,

$$(\mathbf{P}2.1): \min_{\tilde{\mathcal{Z}}} \left(\sum_{n=1}^N T[n] + \varrho \sum_{k=1}^K \sum_{m=1}^M \sum_{n=1}^N |(2\rho_{km}^r[n] - 1)\rho_{km}[n] - (\rho_{km}^r[n])^2|^2 \right)$$

s.t. $(7a), (7c), (7e)-(7j), (8a).$

Proof: First, the constraint $\rho_{km}[n] \in \{0, 1\}$ can be written as the equivalent form of $\rho_{km}[n](\rho_{km}[n] - 1) = 0$ that holds if and only if $\rho_{km}[n]$ equals to 0 or 1. Thus, we denote the penalty term as $\varrho \sum_{k=1}^K \sum_{m=1}^M \sum_{n=1}^N |\rho_{km}[n](\rho_{km}[n] - 1)|^2$ involved in the objective function. For case of $\rho_{km}[n] \notin \{0, 1\}$, the increasing penalty parameter ϱ is imposed to the objective function of problem **P**2.1, which conflicts with the goal of problem. Therefore, $\rho_{km}[n](\rho_{km}[n] - 1) = 0$ must hold at the optimal solution, i.e., $\varrho \gg 1$. Then, we apply SCA method to handle the non-convexity of the introduced penalty term. Via the first-order Taylor expansion for the penalty term at $\rho_{km}^r[n]$, $|(2\rho_{km}^r[n] - 1)\rho_{km}[n] - (\rho_{km}^r[n])^2|^2$ is obtained. Thus, the proof is completed. ∎

It is worth mentioning that the penalized algorithm consists of double loops. The primal variables are optimized in the inner loop, and the penalty parameter is updated with a step size

$c > 1$ in the outer loop. Note that besides the objective function, problem $\mathbf{P}2.1$ is actually identical to problem $\mathbf{P}1$ for partial offloading mode, indicating that problem $\mathbf{P}2.1$ can be solved by the foregoing Step 1 to Step 3 with any given penalty parameter.

C. Procedure of Proposed Algorithms

Based on Step 1-3 above, we propose the effective AO algorithms to solve problems $\mathbf{P}1$ and $\mathbf{P}2.1$, which are summarized in Algorithm 1 and Algorithm 2, respectively. In particular, the number of time slots N is a given deterministic value that, however, has a great impact on the algorithm complexity. To reduce the complexity, we propose a bisection method to obtain a minimum feasible N before starting the algorithms, which is presented in Algorithm 3. Note that each step presented above serves the lower bound for the previous step so that the objective value is non-increasing in iterations. Also, it is evident that the objective function is lower-bounded by a finite value, i.e., 0. Hence, the proposed algorithms are guaranteed to converge.

Algorithm 1 Proposed AO algorithm for partial offloading

1: Initialize $\tilde{\mathcal{Z}}_1$, $\tilde{\mathcal{Z}}_2$, $\tilde{\mathcal{Z}}_3$. Set accuracy ε, iteration $r = 0$, give the maximum number of iterations $I_{\max}$.
2: Determine the value of N via Algorithm 3.
3: **repeat**
4:     Update $\tilde{\mathcal{Z}}_1$ by fixed $\tilde{\mathcal{Z}}_2$ and $\tilde{\mathcal{Z}}_3$ in Step 1.
5:     Update $\tilde{\mathcal{Z}}_2$ by fixed $\tilde{\mathcal{Z}}_1$ and $\tilde{\mathcal{Z}}_3$ in Step 2.
6:     Update $\tilde{\mathcal{Z}}_3$ by fixed $\tilde{\mathcal{Z}}_1$ and $\tilde{\mathcal{Z}}_2$ in Step 3.
7:     Update $r \leftarrow r + 1$.
8: **until** reach the accuracy or exceed the value of $I_{\max}$.

Algorithm 2 Proposed AO algorithm for binary offloading

1: Initialize $\tilde{\mathcal{Z}}_1$, $\tilde{\mathcal{Z}}_2$, $\tilde{\mathcal{Z}}_3$. Set penalty parameter ϱ^l and accuracy ε, iteration $l = 0$, give the maximum number of iterations $I_{\max}$, penalty step $c < 1$.
2: Determine the value of N via Algorithm 3.
3: **repeat**
4:     Solve $\mathbf{P}2.1$ via Step 1 to Step 3 with given ϱ^l.
5:     Update $\varrho^{l+1} \leftarrow c\varrho^l$.
6:     Update $l = l + 1$.
7: **until** reach the accuracy or exceed the value of $I_{\max}$.

Algorithm 3 A bisection method to determine value of N

1: Set a sufficiently large number of time slots N_R that makes the target problem feasible, and let $N_L = 0$.
2: **repeat**
3:     Calculate $N = \lceil \frac{N_L + N_R}{2} \rceil$.
4:     **if** target problem is feasible with N **then**
5:         Update $N_R = N$.
6:     **else**
7:         Update $N_L = N$.
8:     **end if**
9: **until** reach the expected accuracy.

The complexity of Algorithm 1 in each iteration mainly lies in the step 4 to step 6 of this algorithm. Moreover, the number of optimization variables of Algorithm 1 mainly depends on

K, M and N. Based on [12] [16], the total complexity of Algorithm 1 for the partial offloading mode can be roughly given by $\mathcal{O}(LK^3M^3N^3 + KMN\log_2(1/\epsilon_b))$, where L denotes the number of iterations of Algorithm 1, and ϵ_b denotes the accuracy for the bisection search. Similarly, the total complexity of Algorithm 2 for the binary offloading mode can be roughly given by $\mathcal{O}(L_iL_oK^3M^3N^3 + KMN\log_2(1/\epsilon_b))$, where L_i and L_o denote the number of iterations of inner loop and that of outer loop, respectively. It is obvious that the algorithm complexity of the binary offloading mode is higher than that of the partial offloading mode. In practice, compared with the binary offloading mode, the partial offloading mode is more dynamical and flexible to exploit the network resources with a lower algorithm complexity, but it requires a more complex circuit and protocol [3].

IV. NUMERICAL RESULTS

In this section, we present the numerical results. We consider $M = 2$ UAVs serve $K = 6$ TDs at $H = 100$ m with the maximum speed $V_{\max} = 30$ m/s. The referenced channel gain and noise power are $\beta_0 = -60$ dB and $\sigma_0^2 = -110$ dBm, respectively. Based on [14], let $B_1 = -4.3221$, $B_2 = 6.0750$, $C_1 = 0$, $C_2 = 1$, and the path loss exponent $\alpha = 2.3$. Set $c_k = c_m = 10^3$ cycles/bit and the maximum CPU frequency $F_k = 0.5$ GHz, $F_m^{\max} = 10$ GHz. In addition, the maximum time slot length is set as $\Delta_m = 1$ s, the accuracy and maximum iterations are set as $\varepsilon = 10^{-3}$ and $I_{\max} = 50$, respectively. The initial value of penalty parameter is set as $\varrho^0 = 0.01$, and the step $c = 1.5$. The transmit power $P_k = 20$ dBm, and the bandwidth $B = 2$ MHz. The secure distance between UAVs is $d_m = 10$ m. In addition, for the initialization, the trajectories of UAVs are set as circle flight with different radius of 100 m and 80 m, the CPU frequency at UAVs is equally allocated for each TD, and the task partition is set as $\rho_{k1}[n] = \rho_{k2}[n] = 0.4$.
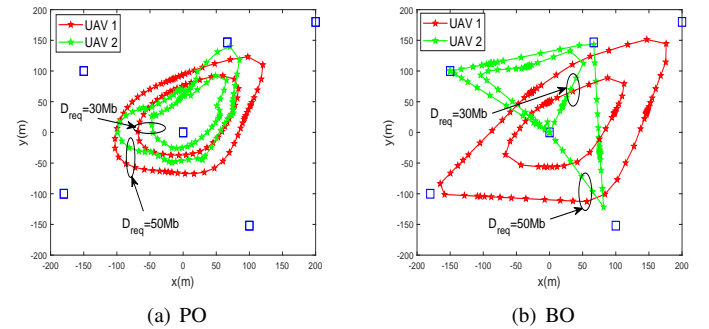

(a) PO

(b) BO

Fig. 2. UAVs' trajectories under different computation requirements.

In Fig. 2, we present the UAVs' trajectories for the the partial offloading (PO) and binary offloading (BO) under $D_{\text{req}} = 30$ Mb and $D_{\text{req}} = 50$ Mb. To reflect the location changes, the trajectories are sampled every one time slot and marked with "$\star$"s. We observe that with D_{req} increasing, the UAVs flies closer to the remote TDs so as to contribute to build high-quality links, thus further decreases the offloading time. It is expected that the UAVs' trajectories for the PO and BO are diverse due to the different scheduling in the two modes.

The convergence of the proposed Algorithm 1 and Algorithm 2 under different task requirements is illustrated in Fig. 3. It shows that the proposed algorithms significantly decrease the task completion time. Note that, at the beginning, Algorithm
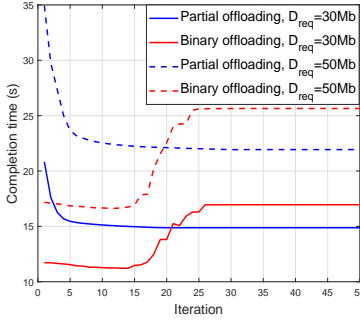
Fig. 3. Convergence performance of the proposed algorithms.
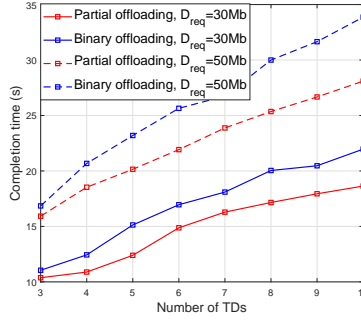

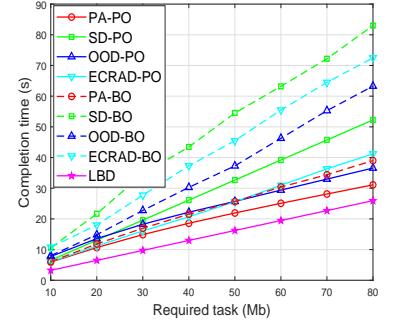Fig. 4. Completion time with varying number of TDs.

Fig. 5. Minimum completion time with varying task requirements.

2 has a sufficient degree of freedom for optimization because the low-valued penalty term has little effect on the objective value. With the increasing of penalty parameter in iterations, the penalty term is playing a increasing impact, thus the objective value is rising until each TD fully executes binary offloading.

In order to illustrate the general applicability of the proposed algorithms, we compare the task completion time with different number of TDs under different computation requirement in Fig. 4. It observes that our proposed AO algorithms are feasible for different network size. Moreover, the task completion time for the binary offloading increases faster than that for the partial offloading, which indicates that the partial mode outperforms the binary mode, as expected. This figure also validates that the proposed AO algorithms are feasible for the real applications with large number of TDs.

Fig. 5 shows the task completion time versus different values of D_{req} for the PO and BO, including the following special designs for comparison:

- Proposed algorithm (PA): Algorithm 1 and Algorithm 2.
- Static design(SD): the UAVs are fixed at two points.
- Equal computation resource allocation design (ECRAD): The CPU frequency of each UAV is equally allocated to each TD in each time slot.
- Only offloading for computing design (OOCD): the computation task is executed only at the UAVs by offloading.
- Lower bound design (LBD): the UAVs' speed is set as $V_{max} \rightarrow \infty$ under partial offloading mode, thus the UAVs are over each TD all the time with the distance of H.

From Fig. 5, LBD actually gives an ideal value that measures the optimality of our proposed algorithm. It is observed that the partial offloading mode outperforms the binary offloading mode because of the flexible allocation for task. Besides, PA for these two offloading modes outperforms the corresponding special cases, which validates the performance gain brought by our proposed algorithms. Moreover, it is observed that OOCD achieves a good performance because of the powerful computation ability of the UAVs. From the fact that PA overmatches the SD and ECRAD, it indicates that the important role in trajectory plan and computation resource optimization.

V. CONCLUSIONS

This paper investigated a problem for MEC systems assisted by multiple UAVs with the goal of task completion time minimization. The NLoS channel model was adopted for UAV-ground links. We developed effective alternating optimization algorithms for two offloading modes. Numerical results showed the convergence of our proposed algorithms, and also validated that the completion time can be significantly decreased. This paper filled the gap of the latency study in multi-UAV assisted MEC systems. This paper also laid a foundation for more interesting and complex scenarios in future study, such as the scenario with probabilistic LoS model.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.,* vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[2] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.,* vol. 36, no. 9, pp. 1927–1941, Sept. 2018.

[3] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wireless Comm.,* vol. 27, no. 1, pp. 140–146, Feb. 2020.

[4] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.,* vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[5] J. Ren, G. Yu, Y. Cai, Y. He, and F. Qu, "Partial offloading for latency minimization in mobile-edge computing," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[6] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Techn.,* vol. 68, no. 5, pp. 5031–5044, May 2019.

[7] W. He, Y. Zhang, Y. Huang, D. He, Y. Xu, G. Yun-feng, and W. Zhang, "Latency minimization for full-duplex mobile-edge computing system," in *Proc. IEEE Int. Conf. Commun.,* Shanghai, China, 2019, pp. 1–6.

[8] N. Zhao, Y. Li, S. Zhang, Y. Chen, W. Lu, J. Wang, and X. Wang, "Security enhancement for NOMA-UAV networks," *IEEE Trans. Veh. Techn.,* vol. 69, no. 4, pp. 3994–4005, Apr. 2020.

[9] X. Liu, J. Wang, N. Zhao, Y. Chen, S. Zhang, Z. Ding, and F. R. Yu, "Placement and power allocation for NOMA-UAV networks," *IEEE Wireless Commun. Lett.,* vol. 8, no. 3, pp. 965–968, Jun. 2019.

[10] Q. Song, S. Jin, and F. Zheng, "Completion time and energy consumption minimization for UAV-enabled multicasting," *IEEE Wireless Commun. Lett.,* vol. 8, no. 3, pp. 821–824, Jun. 2019.

[11] H. Wang, J. Wang, G. Ding, J. Chen, F. Gao, and Z. Han, "Completion time minimization with path planning for fixed-wing UAV communications," *IEEE Trans. Wireless Commun.,* vol. 18, no. 7, pp. 3485–3499, Jul. 2019.

[12] C. Zhan and Y. Zeng, "Completion time minimization for multi-UAV-enabled data collection," *IEEE Trans. Wireless Commun.,* vol. 18, no. 10, pp. 4859–4872, Oct. 2019.

[13] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in UAV-enabled multicasting," *IEEE Trans. Wireless Commun.,* vol. 17, no. 4, pp. 2233–2246 Apr. 2018.

[14] C. You and R. Zhang, "3D trajectory optimization in Rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.,* vol. 18, no. 6, pp. 319–3207, Jun. 2019.

[15] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.,* vol. 6, no. 3, pp. 4188–4200, Jun. 2019.

[16] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press, 2004