Joint Trajectory-Task-Cache Optimization in UAV-Enabled Mobile Edge Networks for Cyber-Physical System

HAIBO MEI [1,2], **KEZHI WANG** [3], **(Member, IEEE), DONGDAI ZHOU** [4],
AND KUN YANG [2,5], **(Senior Member, IEEE)**

[1]School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Zhongshan Institute, University of Electronic Science and Technology of China, Zhongshan 528402, China
[3]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE2 1XE, U.K.
[4]School of Information Science and Technology, Northeast Normal University, Changchun 130024, China
[5]School of Computer Sciences and Electrical Engineering, University of Essex, Colchester CO4 3SQ, U.K.

Corresponding author: Haibo Mei (haibo.mei@uestc.edu.cn)

ABSTRACT This paper studies an unmanned aerial vehicle (UAV)-enabled mobile edge network for Cyber-Physical System (CPS), where UAV with fixed-wing or rotary-wing is dispatched to provide communication and mobile edge computing (MEC) services to ground terminals (GTs). To minimize the energy consumption so as to extend the endurance of the UAV, we intend to jointly optimize its 3D trajectory and the task-cache strategies among GTs to save the energies spent on flight propulsion and GT tasks. Such joint trajectory-task-cache problem is difficult to be optimally solved, as it is non-convex and involves multiple constraints. To tackle this problem, we reformulate the optimizing of task offloading and cache into two tractable linear program (LP) problems, and the optimizing of UAV trajectory into three convex Quadratically Constrained Quadratically Program (QCQP) problems on horizontal trajectory, vertical trajectory and flight time of the UAV respectively. Then a block coordinate descent algorithm is proposed to iteratively solve the formed sub-problems through a successive convex optimization (SCO) process. A high-quality sub-optimal solution to the joint problem then will be obtained, after the algorithm converging to a prescribed accuracy. The numerical results show the proposed solution significantly outperforms the baseline solution.

INDEX TERMS Unmanned aerial vehicle, Internet of Thing, mobile edge computing, 3D trajectory design, cache deployment.

I. INTRODUCTION

Cyber-physical systems (CPS) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. And it is expected that computing and communication capabilities will soon be embedded in all types of objects and structures in the physical environment [1]. However, it is challenging to implement computing and networking technologies to provide an adequate foundation for CPS [2]. In practice, CPS needs reliable wireless network coverage in areas without or with insufficient terrestrial

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Yu.

infrastructures to support the data transmissions. To effectively provide network coverage to CPS, it is beneficial to implement wireless communication using unmanned aerial vehicle (UAV) [3]. Compared to conventional wireless networks, such UAV-enabled wireless communication brings new advantages, such as on-demand and swift deployment, high flexibility with fully-controllable mobility in three dimensional (3D) airspace, and high probability of line-of-sight (LoS) radio frequency links with the Ground Terminals (GTs) [4], [5]. In addition, the UAV-enabled wireless communications deployed for CPS are also in need of efficient techniques to improve the computation capacity and prolong the lifetime of the terminals [6]. This is because CPS usually constitutes a large number of terminals that are typically

constrained by their computation capacity and power. This thus motivates our current work to study UAV-enabled wireless communication working with mobile edge computing (MEC) technique.

MEC is a state-of-art technology that allows wireless terminals to offload the tasks to the edge server instead of locally executing the tasks by the terminals [7]–[10]. MEC thus can greatly improve the computation capacity and energy-efficient performance of those terminals in CPSs that are covered by UAV-enabled wireless network [7] or 4G/5G mobile network [11], [12]. Compared to conventional MEC, the UAV-enabled MEC will better execute the tasks of the GTs by leveraging its high mobility [13]–[17]. Specifically, the UAV that works as the MEC server, can sequentially visit the GTs and receive offloaded tasks only when it moves sufficiently close to each GT. This can ensure high probability of LoS connectivity between UAV and GT and save the transmission energy of all GTs. Moreover, based on the fact that many GTs will request the same popular tasks at different times, UAV can have cache deployed to save tasks in its storage area to avoid redundant data transmissions between itself and the GT. Cache thus can help the UAV-enabled MEC save its energy spent on task offloading [18], [19].

In general, UAV-enabled MEC is particularly useful for the emergency situations, e.g. earthquake, typhoon, where the GTs in CPS having no reach of wireless connection nor power supply. UAV-enabled MEC is also suitable for CPS applications, e.g. intelligent farming, to provide communication and computation services to IoT devices [13]. There has been work studied UAV working with MEC already. In [20], it proposed a UAV-based IoT platform for a surveillance use case, where UAV-enabled MEC is discussed to help task offloading for IoT devices. In [7], the authors discussed the optimal bit allocation in UAV-enabled mobile edge network, considering energy cost. In [21], task offloading was considered in the scenario where UAV works as the cellular user to help execute and offload GT tasks to the base station. It is expected that, as the upcoming of 5G and smart city era, more and more CPS applications with UAV-enabled MEC will appear in short time.

However, one critical issue of UAV-enabled MEC lies in the limited on-board energy of UAVs [22], [23], which needs to be efficiently used to enhance the communication and computation performances and prolong the UAV's mission time. The difficulty is that the energy-efficient wireless communication and computation design with UAV is significantly different from that in conventional terrestrial MEC systems. Therefore, the works on dynamic task offloading [24], [25], resource allocation [26], [27] and cache [28]–[30] for conventional MEC system, can not be directly applied to UAV to support its energy-efficient performance. An initial attempt for designing energy-efficient UAV communication was made in [23], [31]–[35], where UAV trajectory was jointly optimized with the strategies, like resource allocation, user association, power control. In [31], [32], it clarified that the propulsion energy takes most part of the overall energy

consumption of the UAV, and the authors modelled and optimized the propulsion energy of both the fixed-wing and rotary-wing UAVs. The authors in [33]–[35] revealed an interesting trade-off between UAV's energy consumption and that of the GTs it communicating with, base on which the energy-efficient UAV-GT communication was realized via optimized solutions. Furthermore, the work [36], [37] firstly studied cache technique to save the energy consumption and enforce the communication work of the UAV. But the cache deployed in [36], [37] is only for the data dissemination while leaving the more general scenario in CPS unaddressed. The power constraint of UAV also can be released in the more general cases with multiple cooperative UAVs and/or in the presence of ground BSs [21], [38]. However, the more general case is highly non-trivial, and involves additional issues such as UAV collision avoidance, spectrum sharing between UAVs and BSs, and inter-UAV communications.

For the UAV-enabled MEC, the energy issue will escalate due to the extra energy consumed by the UAV to support MEC computations. Recently, the authors of [14]–[17] have studied resource allocation, trajectory design, user association, and dynamic task offloading to improve the energy-efficient performance of UAV-enabled MEC. But those works haven't solved the energy issue optimally yet. In this paper, to better release the energy issue, we aim to minimize the energy consumptions of the UAV-enabled MEC by solving the joint trajectory-task-cache problem. The proposed solution also intends to satisfy the requirement of each GT on task latency. In specific, a solution to the problem has to: 1) determine whether a GT task should be offloaded to the UAV or not, considering the task requirement and status of the UAV; 2) make decision on whether a UAV has to cache a GT task or not, referring its limited storage capacity and the status of the requesting tasks; 3) design 3D UAV trajectory to control UAV fly energy-efficiently and establish effective UAV-GT links to support satisfying task offloading. According to the study, those three jobs are highly correlated and mutually infect each other. Therefore, the joint problem is complicated and with highly coupling non-convex constraints.

To deal with this multiple-constrained non-convex problem, we involve a set of equality constraints and auxiliary variables, and then transform the problem into a tractable formulation. We first reformulate the task offloading and cache as linear program (LP) problems given the 3D trajectory of the UAV pre-decided. Next, we transform the trajectory optimization into three convex Quadratically Constrained Quadratically Program (QCQP) problems on horizontal trajectory, vertical trajectory and flight time of the UAV respectively. Finally, we propose an iterative block coordinate descent algorithm to solve the formed sub-problems through a successive convex optimization (SCO) process [39], [40]. A high-quality sub-optimal solution to the joint problem then will be obtained after the proposed algorithm converging to a prescribed accuracy. The numerical results show the proposed solution significantly outperform the baseline solution. In summary, the primary contributions of this paper are:
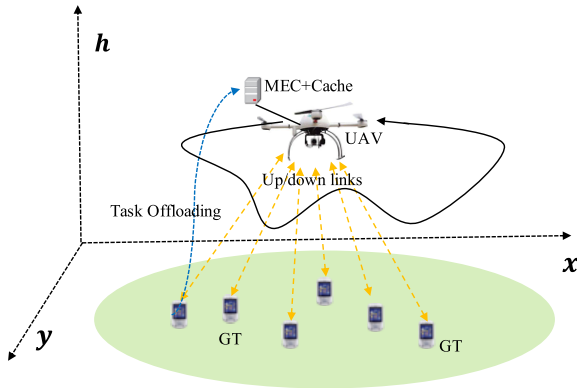
FIGURE 1. Working scenario of the UAV-enabled MEC.

1) It is innovative work to improve the energy-efficient performance of the UAV-enabled MEC by solving the joint trajectory-task-cache problem. We study how the performance of the UAV-enabled MEC being affected by the design of task offloading, cache and 3D UAV trajectory. Then we conclude that the joint problem is multiple constrained and non-convex, which has not been studied yet.

2) It is a pioneer work to optimize the 3D trajectory of the UAV that works as MEC server. Currently, most of the existing works only optimize the UAV trajectory in 2D spaces while leaving UAV latitude fixed in the vertical dimension, which is unrealistic. And some of the recent works optimize the 3D UAV trajectory to enlarge the network throughput. But those works didn't take the energy constraint of the UAV nor MEC into considerations.

The remainder of this paper is organized as follows. In section II, we describe the system model and formulate the problem. In Section III, we present the solution to the optimization problem. In section IV, simulation results and analysis are presented. In Section V, we give conclusions and future work.

II. SYSTEM MODEL AND PROBLEM FORMULATION
A. SYSTEM MODEL
1) ASSUMPTIONS ON UAV AND GTS

This paper considers a typic scenario of UAV-enabled MEC, which is shown in Fig.1. We discretize the UAV path into M line segments, which are represented by $M + 1$ waypoints in 3D coordinates: $\{h_m, q_m\}_{m=1}^{M+1}$. And for the m-th waypoint, h_m is the height and $q_m = \{x_m, y_m\}$ is the horizontal coordinate of the UAV. UAV's height is constrained to $h_{\max} \geq h_m \geq h_{\min}$, where $h_{\min}$ and $h_{\max}$ are the minimum and maximum height of the UAV respectively. Further, one has $\{h_1, q_1\} = \{h_{M+1}, q_{M+1}\}$ to indicate the UAV would fly back to its initial location, which is the common situation in reality. To facilitate the analysis, assume the task offloading will not take place during the stages of UAV taking off and landing, and the UAV will only work as the MEC server when it is

airborne. This assumption is based on the fact that the taking off and landing stages of the UAV only take small portion of the whole mission time. In addition, we impose the following constraints: $\|q_{m+1} - q_m\| \leq \Delta_{\max}^h$, $\|h_{m+1} - h_m\| \leq \Delta_{\max}^v$, $m = 1, \ldots, M$, where $\Delta_{\max}^h$ and $\Delta_{\max}^v$ are appropriately chosen values so that within each line segment, the UAV is assumed to fly with constant horizontal and vertical velocities and the distance between the UAV and each GT is approximately unchanged.

With such path discretization, the UAV trajectory can be represented by the $M + 1$ waypoints $\{h_m, q_m\}_{m=1}^{M+1}$ together with the duration $\{t_m\}_{m=1}^{M}$ representing the time that the UAV spends within each line segment. Then given $V_{\max}^h$ as the maximum horizontal velocity of the UAV and t_m in the m-th line segment, the horizontal flying velocity of the UAV along the m-th line segment can be denoted as $v_m^h = \frac{\sqrt{\|q_{m+1} - q_m\|^2}}{t_m} \leq V_{\max}^h, m = 1, \ldots, M$. In reality, the fixed-wing UAV is constrained to have a minimum horizontal velocity $v_m^h \geq V_{\min}^h > 0, m = 1, \ldots, M$ as it cannot hover statically at a fixed location and needs to move forward to remain aloft in contrast to the rotary-wing UAV. On the other hand, the vertical flying velocity of the UAV along the m-th line segment can be denoted as $0 \leq v_m^v = \frac{\sqrt{\|h_{m+1} - h_m\|^2}}{t_m} \leq V_{\max}^v, m = 1, \ldots, M$, where $V_{\max}^v$ is the maximum vertical velocity of the UAV. If $v_m^v = 0$, the UAV adopts a steady straight-and-level flight (SLF) in m-th line segment. Furthermore, the total mission completion time is given by $\sum_{m=1}^{M} t_m \leq T_{\max}$, where $T_{\max}$ being the maximum allowed UAV mission time. M is chosen to be sufficiently large so that $M\sqrt{(\Delta_{\max}^h)^2 + (\Delta_{\max}^v)^2} \geq \widehat{D}$, where $\widehat{D}$ is an upper bound of the allowed flying distance of the UAV.

Assume there are $\mathcal{K} = \{1, 2, \ldots, K\}$ GTs on the ground, and the horizontal coordinate of each GT is known in advance as $w_k = [x_k, y_k]^T \in \mathbb{R}^{2 \times 1}, k \in \mathcal{K}$. In CPS, GTs are normally in low mobility, thus we assume the GTs in Fig.1 will stay in static during the UAV mission time. Assume the k-th GT has averagely expected amount of computing task $U_k = (F_k, D_k, T_k)$, where F_k describes the total number of the CPU cycles to be computed; D_k denotes the amount of input data to be transferred through the uplink; T_k denotes the task's completion deadline [11], [13]. We ignore the data transmissions in the downlink from UAV to GT, as it will not cause any energy consumption of the GT. Also the data transmission in downlink is trivial and will not consume significant amount of energy of the UAV. We assume the expected amount of each GT task can be efficiently found numerically or predicted by e.g. Machine Learning [36], during the UAV's mission completion time.

2) COMMUNICATION MODEL

In UAV-enabled mobile edge network, we consider the effect of the environment on the occurrence of LoS, and an air-to-ground channel model in urban environments [32], [41]. The LoS connectivity probability between the UAV and the k-th

GT in path line m is given as

$$p_{km}^{los} = \frac{1}{1 + a \exp\left(-b\left(\arctan\left(\frac{h_m}{d_{km}}\right) - a\right)\right)}, \quad \forall k, m \quad (1)$$

where a and b are constant values that depend on the environment. In this setting, the altitude and antenna heights of the GT are neglected. The LoS connectivity probability between the UAV and the k-th GT in path line m depends on the altitude of the UAV h_m and the horizontal distance between the UAV and the GT denoted as $d_{km} = \sqrt{\|q_m - w_k\|^2}$. In other words, the pathloss of the air-to-ground link depends on the altitude in the vertical dimension, and the distance in the horizontal dimension. Then the pathloss expression becomes

$$l_{km} = 20\log(\sqrt{h_m^2 + d_{km}^2}) + A p_{km}^{los} + C, \quad \forall k, m \quad (2)$$

where A and C are constants such that $A = \eta_{\text{LoS}} - \eta_{\text{NLos}}$ and $C = 20\log\left(\frac{4\pi f_c}{c}\right) + \eta_{\text{NLos}}$; f_c is the carrier frequency (Hz); c is the speed of light (m/s); η_{LoS} and η_{NLoS} (in dB) are respectively the losses corresponding to the LoS and non-LoS connections depending on the environment. Following the pathloss model between UAV and GT, the average achievable rate of k-th GT's uplink, denoted by r_k in bits/second (bps), can be expressed as

$$r_k = \frac{B}{\sum_{m=1}^{M} t_m} \sum_{m=1}^{M} t_m \log_2\left(1 + \frac{p_k 10^{-\frac{l_{km}}{10}}}{B N_0}\right), \quad \forall k \quad (3)$$

where p_k is the uplink transmission power of GT k; B denotes the allocated bandwidth; N_0 denotes the power spectral density of the Additive White Gaussian Noise (AWGN).

3) UAV PROPULSION ENERGY MODEL

For a fixed-wing UAV, the propulsion energy is expressed as

$$E^{\text{f-uav}} = \sum_{m=1}^{M} t_m \left(c_1 \left(v_m^h\right)^3 + \frac{c_2}{v_m^h} + c_3 v_m^v\right) \quad (4)$$

where c_1 and c_2 are the two constant parameters related to the UAV's weight, wing area, and air density [32]; c_3 is the constant parameter related to the UAV's descending/ascending. For a rotary-wing UAV, the propulsion energy is modelled as

$$E^{\text{r-uav}} = \sum_{m=1}^{M} t_m \left(P_0(1 + \frac{3(v_m^h)^2}{U_{\text{tip}}^2}) + \frac{1}{2} d_0 \rho s G(v_m^h)^3 \right.$$
$$\left. + P_1(\sqrt{1 + \frac{(v_m^h)^4}{4v_0^4}} - \frac{(v_m^h)^2}{2v_0^2})^{\frac{1}{2}} + P_2 v_m^v)\right) \quad (5)$$

where P_0 and P_1 are two constants defined, representing the blade profile power and induced power in hovering status, respectively; P_2 is the constant on descending/ascending power. U_{tip} denotes the tip speed of the rotor blade, v_0 is known as the mean rotor induced velocity in hover, d_0 and s are the fuselage drag ratio and rotor solidity, respectively,

and ρ and G denote the air density and rotor disc area, respectively [31].

In general, the propulsion energy of UAV depends on the horizontal and vertical velocities of the UAV in each path segment. For the purpose of exposition and more tractable analysis, we ignore the additional/fewer energy consumption caused by UAV acceleration/deceleration, which is reasonable for scenarios when the UAV maneuvering only takes a small portion of the total operation time. In addition, the propulsion energy on UAV ascending/descending adopts a simple way to vertical velocity v_m^v. An advanced propulsion energy model on UAV ascending/descending will be exploited in future work.

4) COMPUTATION MODEL

According to task offloading, if task U_k is offloaded to the UAV, it will allocate f_k^o CPU cycles to the task. And if task U_k is executed locally, we assume f_k^l denotes the CPU cycles allocated to the task by the k-th GT. The latency of U_k is then denoted as

$$L_k = (1 - a_k)\frac{F_k}{f_k^l} + a_k \left(\frac{F_k}{f_k^o} + \frac{D_k}{r_k}\right), \quad \forall k \quad (6)$$

where $0 \leq a_k \leq 1$ is the offloading indicator from k-th GT to the UAV; $a_k = 1$ denotes that GT k decides to offload all the task to the UAV, while $a_k = 0$ indicates that GT k decides to conduct the task all by itself. If $0 < a_k < 1$, the k-th GT will offload portion of its task to the UAV. Assume the UAV has limited computation capacity C^o, then one has $\sum_{k=1}^{K} a_k f_k^o \leq C^o$. In (6), it shows that if task U_k is offloaded to the UAV, the task latency is then composed by the time spent by the UAV on computing the task and the time spent on the data transmission in the uplink. In contrast, if the task is not offloaded but executed locally, the task latency only involves the time spent by the GT on executing the task locally. In addition, we assume that the GT can locally finish the task before the deadline, i.e. $\frac{F_k}{f_k^l} < T_k$.

To execute task U_k, the energy consumption of the UAV and GT are respectively formulated as

$$E_k^u = a_k E_k^o, \quad \forall k \quad (7)$$

$$E_k^{gt} = a_k \frac{D_k}{r_k} p_k + (1 - a_k)E_k^l, \quad \forall k \quad (8)$$

where $E_k^o = \varphi(f_k^o)^{\vartheta - 1} F_k$ and $E_k^l = \varphi(f_k^l)^{\vartheta - 1} F_k$ are the computation related energy consumption of the UAV and GT respectively; φ is the effective switched capacitance and $\vartheta \geq 1$ is the positive constant as mentioned in [11], [13]. In (8), it denotes that if task U_k is offloaded, GT k only consumes its energy on transmitting the data through uplink. On the other hand, if task U_k is not offloaded, GT k only consumes its energy on locally executing the task. And in (7), it demonstrates that the UAV will consumes its energy on executing the task U_k, only if the task being offloaded.

5) CACHE MODEL

The process of task caching is as follows. GT first requests the task that needs to be offloaded. If the task been cached in the UAV, GT then does not need to transfer data to the UAV. When the UAV finishes task processing, it only has to transmit the result to the GT [18], [19]. With cache deployed, the latency and the energy consumption of U_k can be formulated as

$$L_k^c = (1 - x_k)L_k + x_k a_k \frac{F_k}{f_k^o}, \quad \forall k \tag{9}$$

$$E_k^c = (1 - x_k)(\alpha E_k^{gt} + E_k^u) + x_k E_k^u, \quad \forall k \tag{10}$$

where $0 \le x_k \le 1$ is the caching decision variable; $x_k = 1$ denotes that GT k is fully cached in the UAV, while $x_k = 0$ denotes the task is not cached at all. If $0 \le x_k \le 1$, it means portion of the task being cached in the UAV. One has $\alpha > 0$ as the tradeoff between the energy consumptions of GT and UAV on serving task U_k. Assume the cache storage in UAV is limited, and set to be $\sum_{k=1}^{K} x_k D_k \le C^c$. In (9), it denotes that if task U_k is fully cached in the UAV, the task latency is then only composed by the time spent by the UAV on computing the task. And in (10), it shows that if task U_k is cached, the energy consumption only involves the one spent by the UAV on executing the task. In contrast, if task is not cached, the latency and energy consumption of task U_k will be the same as the ones formulated in (6) (7) and (8). To this end, we can find that cache can effectively save the latency and energy caused by the data transmission during task offloading.

B. PROBLEM FORMULATION

Let $\mathbf{A} = \{a_k, k \in \mathcal{K}\}$, $\mathbf{X} = \{x_k, k \in \mathcal{K}\}$, $\mathbf{Q} = \{q_m\}_{m=1}^{M+1}$, $\mathbf{H} = \{h_m\}_{m=1}^{M+1}$, and $\mathbf{T} = \{t_m\}_{m=1}^{M}$, the optimization problem is formulated as

$$\mathcal{P}: \min_{\mathbf{A}, \mathbf{X}, \mathbf{Q}, \mathbf{H}, \mathbf{T}} \left(E^{\text{uav}} + \beta \sum_{k=1}^{K} E_k^c \right) \tag{11}$$

$$\text{s.t. } 0 \le a_k \le 1, \ 0 \le x_k \le 1, \quad \forall k; \tag{11.1}$$

$$\sum_{k=1}^{K} x_k D_k \le C^c; \tag{11.2}$$

$$\sum_{k=1}^{K} a_k f_k^o \le C^o; \tag{11.3}$$

$$T_k \ge L_k^c, \forall k; \tag{11.4}$$

$$\sum_{m=1}^{M} t_m \le T_{\max}; \tag{11.5}$$

$$\{q_1, h_1\} = \{q_{M+1}, h_{M+1}\}; \tag{11.6}$$

$$V_{\min}^h \le \frac{\|q_{m+1} - q_m\|}{t_m} \le V_{\max}^h, \quad m = 1, .., M; \tag{11.7}$$

$$0 \le \frac{\|h_{m+1} - h_m\|}{t_m} \le V_{\max}^v, m = 1, .., M; \tag{11.8}$$

$$\|q_{m+1} - q_m\| \le \Delta_{\max}^h, \quad m = 1, \ldots, M; \tag{11.9}$$

$$\|h_{m+1} - h_m\| \le \Delta_{\max}^v, \quad m = 1, \ldots, M; \tag{11.10}$$

$$h_{\min} \le h_m \le h_{\max}, \ \forall m; \tag{11.11}$$

where $E^{\text{uav}} = E^{\text{f-uav}}$ for fixed-wing, and $E^{\text{uav}} = E^{\text{r-uav}}$ for rotary-wing UAV; $\beta > 0$ is the tradeoff between the energy consumptions on UAV propulsion and serving GT tasks; (11.1) denotes the constraints of the variables a_k and x_k on task offloading and cache, each of which is decimal between 0 and 1. In reality, if the GT tasks are atom and cannot be divided, the optimizations of the variables a_k and x_k in (11.1) become 0-1 dynamic program problems. In this case, we can further employ the method in [33] to convert a_k and x_k to be decimal between 0 and 1. (11.2) and (11.3) are the constraints on the limited computation and storage capacities of the UAV. (11.4) is the constraint on the service requirement of each task, i.e. task's latency being lower than its completion deadline; (11.5) is the constraint on UAV mission completion time; (11.6) $\sim$ (11.11) are the constraints on the trajectory of the rotary-wing or fixed-wing UAV in horizontal and vertical dimensions respectively.

In (11), it denotes that cache is highly co-related to the task offloading, which makes the design of these two strategies a complicated dynamic programming problem. The task-cache strategies are also effected by the design of UAV trajectory that determines the UAV-GT links. On the other hand, it is a non-convex problem to optimize UAV trajectory in terms of its locations in horizontal and vertical dimensions. This is because the locations of UAV in both of the two dimensions are jointly constrained with respect to the LoS probabilities as defined in (1). Also the velocities of the UAV in both dimensions should neither be too fast nor too slow to save the propulsion energies defined in (4) and (5). In addition, even in an implicit manner, the energy consumption on UAV propulsion E^{uav} is in a complicated non-linear form that is closely related to different styles of UAVs. To this end, it is difficult to solve $\mathcal{P}$ in its current form.

III. PROPOSED SOLUTION

To solve problem $\mathcal{P}$, we intend to optimize the task offloading and cache jointly with the design of UAV trajectory. The process is formulated as a block coordinate descent algorithm. In each iteration of the algorithm, we first optimize the task offloading and cache strategies with pre-defined UAV trajectory $\mathbf{Q}, \mathbf{H}, \mathbf{T}$. Then we optimize the UAV trajectory given $\mathbf{A}, \mathbf{X}$. This is similar to the SCO solution proposed in [42], [43], which can decrease the complexity of the problem.

Based on this SCO method, in sub-section A, we optimize the task offloading and cache variables given the UAV trajectory pre-defined. In this sub-section, the task-cache problem is simplified as two LP problems and get directly solved. In sub-section B, we reform the UAV trajectory problem into three convex QCQP problems and get them solved. Next, based on the works in sub-sections A and B, we design an

overall algorithm in sub-section *C* to find the sub-optimal solution to the whole trajectory-task-cache problem via a SCO procedure. In the end, we analyze the complexity and convergency of the overall algorithm in sub-section *D*.

A. OPTIMIZE TASK OFFLOADING AND CACHE GIVEN UAV TRAJECTORY

Given pre-defined UAV trajectory, the problem on optimizing task offloading and cache can be simplified as

$$\min_{\mathbf{A}, \mathbf{X}} \left(\sum_{k=1}^{K} E_k^c \right) \quad \text{s.t. (11.1); (11.2); (11.3); (11.4);} \quad (12)$$

This problem however is not jointly convex with respect to **A** and **X**, and cannot be directly solved. The non-convexity is due to the existence of the second order term of $a_k x_k$ in (11.4). To linearize the second order term, we give an theorem,

Theorem 1: With the UAV trajectory determined, the data rate of the uplink from UAV to each GT k will be known as R_k. Then, in order to ensure the GT task U_k finished before its completion deadline, the task offloading variable a_k and caching variable x_k of task U_k have to be jointly constrained to

$$a_k(1 - x_k) \leq \begin{cases} H_k & \text{if } H_k > 0 \\ 1 & \text{if } H_k = 0 \end{cases} \quad (13)$$

where $H_k = \min \left\{ \max \left\{ \dfrac{T_k - \frac{F_k}{f_k^l}}{\frac{D_k}{R_k} + \frac{F_k}{f_k^o} - \frac{F_k}{f_k^l}}, 0 \right\}, 1 \right\}$. □

Proof: see Appendix. □

According to the theorem, if the UAV-GT link has a low data rate R_k causing the UAV not able to finish the whole task before its completion deadline, i.e. $\frac{D_k}{R_k} + \frac{F_k}{f_k^o} > T_k > \frac{F_k}{f_k^l}$, one will have $H_k < 1$. Under this situation, if the task not cached, i.e. $x_k = 0$, GT k only can offload portion of its task to the UAV to avoid intolerable task latency, which is denoted as $a_k \leq H_k < 1$. On the contrary, if the UAV-GT link has data rate R_k higher enough to have $\frac{D_k}{R_k} + \frac{F_k}{f_k^o} \leq T_k$, it will lead to $H_k = 1$. In this case, the GT can possibly offload all of its task to the UAV and get it executed in time even the task not cached in the UAV, which is denoted as $a_k \leq H_k = 1$. This follows the real situation that high qualify UAV-GT links can better support the task offloading. Moreover, considering task cache, the theorem indicates that one should make $(1 - x_k)$ as small as possible, i.e, x_k as high as possible, to ensure $a_k(1 - x_k) \leq H_k$. This reflects the real situation that the UAV should cache as much the offloaded GT tasks as possible in its storage area to short the task latency.

Based on the theorem, if we bring (13) into (12), the original problem can be simplified as

$$\min_{\mathbf{A}} \sum_{k=1}^{K} \hat{E}_k^c \quad \text{s.t. (11.1); (11.3);} \quad (14)$$

where $\hat{E}_k^c$ is converted from E_k^c in (10). Specifically, given R_k being the known data rate of the uplink from the

UAV to GT k, one has $E_k^c = a_k E_k^o + \alpha (1 - x_k) E_k^l + \alpha a_k (1 - x_k) \left(\frac{D_k}{R_k} p_k - E_k^l \right)$ in (10). Then, if we replace the term $a_k(1 - x_k)$ of E_k^c by H_k, one has $\hat{E}_k^c = a_k E_k^o + \frac{\alpha H_k E_k^l}{a_k} + \alpha H_k \left(\frac{D_k}{R_k} p_k - E_k^l \right)$. Due to the theorem, it is obvious that $\hat{E}_k^c$ is the upper bound of E_k^c, i.e. $E_k^c \leq \hat{E}_k^c$. Therefore, if we find the optimal **A** to ensure $\hat{E}_k^c$ minimized, such obtained **A** will be also the optimal solution to the problem in (12). We can find that the second order term of $a_k x_k$ disappears in $\hat{E}_k^c$, and (14) therefore is a LP problem. Then **A** can be directly obtained by solving the LP problem. To do so, we follow the monotonicity of $\hat{E}_k^c$ and take its deviation, then the optimized **A** can be obtained as: $a_k = \min\{ \sqrt{\frac{\alpha H_k E_k^l}{E_k^o}}, H_k \}, \forall k$.

After **A** obtained, to let the UAV cache as much GT task as possible, the cache deployment **X** can be obtained by solving the LP problem as,

$$\max_{\mathbf{X}} \sum_{k=1}^{K} x_k a_k D_k \quad \text{s.t. (11.1); (11.2); (13);} \quad (15)$$

where the LP problem can be solved by the CVX tool [44].

B. OPTIMIZE UAV TRAJECTORY GIVEN TASK OFFLOADING AND CACHE

With obtained **A** and **X**, the UAV's trajectory is to be optimized to minimize the propulsion energy, while guaranteeing the GT-UAV uplinks to have satisfied data rate. Then the problem on optimizing the 3D trajectory of the UAV can be simplified as

$$\min_{\mathbf{Q}, \mathbf{H}, \mathbf{T}} E^{\text{uav}} \quad \text{s.t.(11.5)} \sim (11.11); \ r_k \geq R_k', \forall k; \quad (16)$$

where $R_k' = \frac{(a_k - a_k x_k) D_k}{(a_k - a_k x_k) \frac{F_k}{f_k^l} - \left(a_k \frac{F_k}{f_k^o} - x_k \frac{F_k}{f_k^l} \right) + T_k - \frac{F_k}{f_k^l}}$, which is obtained following (9) and (11.4). The problem in (16) is non-convex and cannot be solved directly. We intend to release the non-convexities and divide the problem into three sub-problems on horizontal trajectory **Q**, vertical trajectory **H** and flight time **T** respectively. The obtained convex sub-problems then can be solved by convex optimization technique under through a successive convex optimization (SCO) procedure.

1) OPTIMIZE UAV HORIZONTAL TRAJECTORY

Given **H** and **T**, the problem on optimizing UAV horizontal trajectory **Q** can be formulated as

$$\min_{\mathbf{Q}} \max_{\forall m} \left\| \|q_{m+1} - q_m\|^2 - (t_m v_h^*)^2 \right\| \quad (17)$$

$$\text{s.t. (11.6), } r_k \geq R_k', \quad \forall k; \quad (17.1)$$

$$\|q_{m+1} - q_m\|^2 \geq \max\{((1 - \theta)t_m v_h^*)^2, (t_m V_{\min}^h)^2\} \quad (17.2)$$

$$\|q_{m+1} - q_m\|^2 \leq \min\{((1 + \theta)t_m v_h^*)^2, (\Delta_{\max}^h)^2\}; \quad (17.3)$$

where v_h^* is the horizontal velocity in optimal that leads to the minimum of the propulsion power in horizontal dimension.

For fixed-wing UAV, by solving the convex expression of $E^{\text{f-uav}}$ in (4), a closed-form expression of v_h^* can be obtained as: $\min\left(\max\left(\sqrt[4]{\frac{c_2}{3c_1}}, V_{\min}^h\right), V_{\max}^h\right)$. For rotary-wing, v_h^* is difficult to obtain due to the complicated expression of $E^{\text{r-uav}}$ in (5), however it can be efficiently found numerically [31]. In (17), θ is the slack variable to allow the UAV to adopt a trajectory with horizontal velocity v_m^h between $(1-\theta)v_h^*$ and $(1+\theta)v_h^*$. This enables the UAV to approach the optimized velocity v_h^* to have sub-minimized propulsion energy in horizontal dimension, while supporting the UAV having requested data rate linked to GTs. However, (17) is still a non-convex problem due to the non-convexities in (17.1) and (17.2).

To release the non-convexity in (17.1), we can reform the left side of (17.1) to

$$r_k = \frac{B}{\sum_{m=1}^M t_m} \sum_{m=1}^M t_m \log_2 \times \left(1 + 10^{-\frac{Ap_{km}^{los}}{10}} \times \frac{\kappa_{km}}{h_m^2 + d_{km}^2}\right)$$
(18)

where $\kappa_{km} = \frac{p_k}{BN_0} 10^{-\frac{C}{10}}$. Because $p_{km}^{los} \geq 0$, and $A < 0$, i.e. $\eta_{\text{LoS}} < \eta_{\text{NLos}}$, (18) can be transformed to

$$r_k \geq \frac{B}{\sum_{m=1}^M t_m} \sum_{m=1}^M t_m \log_2(1 + 10^{-\frac{A}{10}} \times \frac{\kappa_{km}}{h_m^2 + d_{km}^2})$$
(19)

Then (19) can be simply denoted as

$$r_k \geq \frac{B}{\sum_{m=1}^M t_m} \sum_{m=1}^M t_m \log_2 (1 + \xi_{km} f(v_{km}));$$
(20)

where $\xi_{km} = \frac{\kappa_{km}}{h_m^2} 10^{-\frac{A}{10}}$; $f(x) = \frac{1}{1+x}$ and $v_{km} = \frac{d_{km}^2}{h_m^2}$. Obviously, the right side of the inequality in (20) is non-linear to v_{km}. It motivates us to replace this non-linearity by more tractable convex function derived from the Taylor expansion at a given local point [39]. Specifically, with given local point $v_{km}^l = \frac{(d_{km}^l)^2}{h_m^2}$, we have the following inequality

$$\sum_{m=1}^M t_m \log_2 (1 + \xi_{km} f(v_{km})) \geq \sum_{m=1}^M t_m (J_{km}^l \left(v_{km} - v_{km}^l\right) + S_{km}^l)$$
(21)

where

$$J_{km}^l = \frac{-\xi_{km}}{\ln 2 \left((1 + v_{km}^l)^2 + \xi_{km} (1 + v_{km}^l)\right)},$$

$$S_{km}^l = \log_2 \left(1 + \xi_{km} f(v_{km}^l)\right).$$

And the local point $v_{km}^l = \frac{(d_{km}^l)^2}{h_m^2}$ is directly obtained given current status of the horizontal trajectory $\{q_{m+1}^l, q_m^l\}$. Then, (17.1) can be expressed in a linear form as

$$r_k^{bl} = \frac{B}{\sum_{m=1}^M t_m} \sum_{m=1}^M t_m (J_{km}^l \left(v_{km} - v_{km}^l\right) + S_{km}^l) \geq R_k'$$
(22)

Further, we intend to release the non-convexity in (17.2). To do so, given local points $\{q_{m+1}^l, q_m^l\}$ as the horizontal trajectory to be updated, we define the following inequality by applying first-order Taylor expansion of the left side of (17.2) as

$$\|q_{m+1} - q_m\|^2 \geq -\left\|q_{m+1}^l - q_m^l\right\|^2 + 2\left(q_{m+1}^l - q_m^l\right)^T (q_{m+1} - q_m) = q_m^{bl}$$
(23)

Then constraint (17.2) can be converted as

$$q_m^{bl} \geq \max\{((1-\theta)t_m v_h^*)^2, (t_m V_{\min}^h)^2\}$$
(24)

To this end, the problem in (17) can be finally redefined as a convex Quadratically Constrained Quadratically Program (QCQP) problem and solved directly by CVX tool, as

$$\min_{\mathbf{Q}} \max_{\forall m} \quad \left\|q_m^{bl} - (t_m v_h^*)^2\right\|$$
s.t. (11.6); (17.3); (22); (24); (25)

2) OPTIMIZE UAV VERTICAL TRAJECTORY

Given $\mathbf{Q}$ and $\mathbf{T}$, the problem on optimizing UAV vertical trajectory $\mathbf{H}$ can be formulated as

$$\min_{\mathbf{H}} \max_{\forall m} \quad \|h_{m+1} - h_m\|$$
s.t. (11.6); (11.8); (11.10); (11.11); $\hat{r}_k^{bl} \geq R_k'$, $\forall k$ (26)

where $\hat{r}_k^{bl} = \frac{B}{\sum_{m=1}^M t_m} \sum_{m=1}^M t_m \left(Y_{km}^l \left(\psi_{km} - \psi_{km}^l\right) + Z_{km}^l\right)$, $\psi_{km} = \frac{h_m^2}{d_{km}^2}$; $\psi_{km}^l = \frac{(h_m^l)^2}{d_{km}^2}$; $Y_{km}^l = \frac{-\hat{\xi}_{km}}{\ln 2\left((1+\psi_{km}^l)^2 + \hat{\xi}_{km}(1+\psi_{km}^l)\right)}$, $Z_{km}^l = \log_2\left(1 + \hat{\xi}_{km} f(\psi_{km}^l)\right)$; $\hat{\xi}_{km} = \frac{\kappa_{km}}{d_{km}^2} 10^{-\frac{A}{10}}$. Obviously, $\hat{r}_k^{bl}$ is obtained following the identical procedure as the one obtaining r_k^{bl} in (22), but taking h_m, $\forall k$ as the variables. (26) is to optimize the UAV to approach SLF, i.e. $\|h_{m+1} - h_m\| \to 0$, $\forall m$ to minimize the propulsion energy in vertical dimension, while ensuring acceptable data rate of UAV-GT uplinks. Obviously (26) is a QCQP problem and can be solved directly by CVX tool.

3) OPTIMIZE UAV FLIGHT TIME

Given $\mathbf{Q}$ and $\mathbf{H}$, the problem on optimzing UAV flight time $\mathbf{T}$ in each path line can be formulated as

$$\min_{\mathbf{T}} \max_{\forall m} \quad \left\|(t_m v_h^*)^2 - \|q_{m+1} - q_m\|^2\right\|$$
(27)
s.t. (11.5); (11.7); (11.8);

$$\sum_{m=1}^M t_m \leq \sum_{m=1}^M t_m^l$$
(27.1)

$$\sum_{m=1}^M t_m r_{km} \geq \frac{\sum_{m=1}^M t_m^l}{B} R_k';$$
(27.2)

where $r_{km} = \log_2\left(1 + \frac{p_k 10^{\frac{-l_{km}}{10}}}{BN_0}\right)$; (27.1) denotes the UAV flight time should be decreased to be lower than current

time $\sum_{m=1}^{M} t_m^l$. According to (3) and (27.1), one could have $r_k = \frac{B}{\sum_{m=1}^{M} t_m} \sum_{m=1}^{M} t_m r_{km} \geq \frac{B}{\sum_{m=1}^{M} t_m^l} \sum_{m=1}^{M} t_m r_{km}$. Thus the requirement of the UAV-GT links on data rate can be converted to $\frac{B}{\sum_{m=1}^{M} t_m^l} \sum_{m=1}^{M} t_m r_{km} \geq R_k'$, and denoted as (27.2). Obviously, (27) is to optimize the flight time t_m of the UAV in each path line to let the UAV approach the optimal horizontal velocity v_h^* to minimize propulsion energy in horizontal dimension, while ensures each GT task finished before its completion deadline. Because (27.1) constrains the flight time to be short, thus (27) does guarantee the propulsion energy of the UAV to be minimized. In addition, it can be noticed in (4) and (5), that we simply assume the propulsion energy in vertical dimension only correlates to the distance on UAV descending/ascending, the flight time optimization in (27) thus does not need to take the vertical velocity of the UAV into consideration.

C. OVERALL ALGORITHM DESIGN

Using the results obtained in the previous two subsections, the overall block coordinate descend algorithm to solve $\mathcal{P}$ can be designed as Algorithm 1. At step 1, the UAV trajectory is firstly initialized to fly a circular trajectory through the method proposed in [33]. The UAV trajectory can also be initialized to travel to each places of interest with the shortest distance, which can be found by solving the traveling salesman problem (TSP) [45]. Based on the initialized trajectory, the UAV trajectory can either lead to coverage fairness (circular trajectory) or geographically close to the GTs (TSP trajectory). At step 1, it also assumes the UAV stays in each of the initialized path line segment in a fixed duration $\{t_0\}_{m=1}^{M}$ and flies with height $\{h_m^0\}_{m=1}^{M+1}$. After initialization, the algorithm mainly runs a loop with finite iterations from step 2 to step 8. In each iteration, it sequentially solves the problems in (25), (26) and (27) to find sub-optimal UAV trajectory. At step 3, it optimizes the task offloading and cache by solving the problem in (14) and (15). Then, under through such block coordinate descend algorithm, it gradually allows the UAV serving GTs with lower energy consumption. After the algorithm converging to a prescribed accuracy within finite iterations, a desirable sub-optimal solution of problem $\mathcal{P}$ will eventually be found.

D. COMPLEXITY AND CONVERGENCE OF THE OVERALL ALGORITHM

Generally, in each iteration of Algorithm 1, the UAV horizontal trajectory, vertical trajectory, and flight time are sequentially optimized using the convex solver based on the interior-point method, and thus each of their individual complexities can be resented by $O((KM)^{3.5}\log(1/\epsilon))$, given the solution accuracy of $\epsilon > 0$ [46]. In addition, at step 3, the time complexity on optimizing task offloading and cache variables $\mathbf{A}$, $\mathbf{X}$ can be denoted as $O(K)$ and $O((K)^{3.5}\log(1/\epsilon))$ respectively. Then accounting for the block coordinate descendant iterations of Algorithm 1 with the complexity in the order of

---

Algorithm 1 Solve the Trajectory-Task-Cache Problem

1   Initialize UAV trajectory as: $\mathbf{A}^0, \mathbf{X}^0, \mathbf{Q}^0, \mathbf{H}^0, \mathbf{T}^0$, $i = 0$;
2   **repeat**
3   Obtain $\mathbf{A}^{i+1}, \mathbf{X}^{i+1}$ by soving (14)(15), given $\mathbf{Q}^i$, $\mathbf{H}^i$, $\mathbf{T}^i$;
4   Obtain $\mathbf{Q}^{i+1}$ by solving (25), given $\mathbf{H}^i$, $\mathbf{T}^i, \mathbf{A}^{i+1}, \mathbf{X}^{i+1}$;
5   Obtain $\mathbf{H}^{i+1}$ by solving (26), given $\mathbf{Q}^{i+1}$, $\mathbf{T}^i, \mathbf{A}^{i+1}, \mathbf{X}^{i+1}$;
6   Obtain $\mathbf{T}^{i+1}$ by solving (27), given $\mathbf{Q}^{i+1}$, $\mathbf{H}^{i+1}, \mathbf{A}^{i+1}, \mathbf{X}^{i+1}$;
7     $i = i + 1$;
8   **until** *Converge to a prescribed accuracy*;
9   Return $\mathbf{A} = \mathbf{A}^i, \mathbf{X} = \mathbf{X}^i, \mathbf{Q} = \mathbf{Q}^i, \mathbf{H} = \mathbf{H}^i, \mathbf{T} = \mathbf{T}^i$;

$\log(1/\epsilon)$, the total computation complexity of Algorithm 1 is $O(KM)^{3.5}\log^2(1/\epsilon)$.

Afterward, we concern the convergence of Algorithm 1. Let $E(\mathbf{A}^i, \mathbf{X}^i, \mathbf{Q}^i, \mathbf{H}^i, \mathbf{T}^i)$ denote the object value of $\mathcal{P}$ in the i-th iteration of Algorithm 1. Then, in the $(i+1)$-th iteration of Algorithm 1, one has

$$
\begin{aligned}
E(\mathbf{A}^i, & \mathbf{X}^i, \mathbf{Q}^i, \mathbf{H}^i, \mathbf{T}^i) \\
&\overset{(a)}{\geq} E(\mathbf{A}^{i+1}, \mathbf{X}^{i+1}, \mathbf{Q}^i, \mathbf{H}^i, \mathbf{T}^i) \\
&\overset{(b)}{=} E^{bl}(\mathbf{A}^{i+1}, \mathbf{X}^{i+1}, \mathbf{Q}^i, \mathbf{H}^i, \mathbf{T}^i) \\
&\overset{(c)}{\geq} E(\mathbf{A}^{i+1}, \mathbf{X}^{i+1}, \mathbf{Q}^{i+1}, \mathbf{H}^i, \mathbf{T}^i) \\
&\overset{(d)}{=} E^{bl}(\mathbf{A}^{i+1}, \mathbf{X}^{i+1}, \mathbf{Q}^{i+1}, \mathbf{H}^i, \mathbf{T}^i) \\
&\overset{(e)}{\geq} E(\mathbf{A}^{i+1}, \mathbf{X}^{i+1}, \mathbf{Q}^{i+1}, \mathbf{H}^{i+1}, \mathbf{T}^i) \\
&\overset{(f)}{=} E^{bl}(\mathbf{A}^{i+1}, \mathbf{X}^{i+1}, \mathbf{Q}^{i+1}, \mathbf{H}^{i+1}, \mathbf{T}^i) \\
&\overset{(g)}{\geq} E(\mathbf{A}^{i+1}, \mathbf{X}^{i+1}, \mathbf{Q}^{i+1}, \mathbf{H}^{i+1}, \mathbf{T}^{i+1}) \quad (28)
\end{aligned}
$$

where (a), (c), (e) and (g) hold, if problems in (14), (15), (25), (26) and (27) are sequentially solved. We can find that (b) and (d) hold, because we obtain $\mathbf{Q}^{i+1}$ and $\mathbf{H}^{i+1}$ by taking $\mathbf{Q}^i$ and $\mathbf{H}^i$ as the local points of the first-order Taylor expansions in (25) and (26). Further, (f) holds because we obtain $\mathbf{T}^{i+1}$ by taking $\mathbf{T}^i$ as the benchmark local point in (27). Thus, it is validated that the object value of $\mathcal{P}$ is non-increasing in Algorithm 1, and the Algorithm can converge to a locally optimal solution of problem $\mathcal{P}$.

IV. SIMULATION

In this section, the proposed solution is validated through simulation using Matlab with CVX tool. The simulation considers GT distributions in densified, medium, and sparse scenarios, which is to validate the adaptation of the proposed solution. As discussed in previous sections, we assume the rotary-wing/fixed-wing UAV will be initialized to have circular trajectory and TSP trajectory, and the stages of UAV taking off and landing will not be considered during the trajectory design. We set the system parameters as listed in table 1.
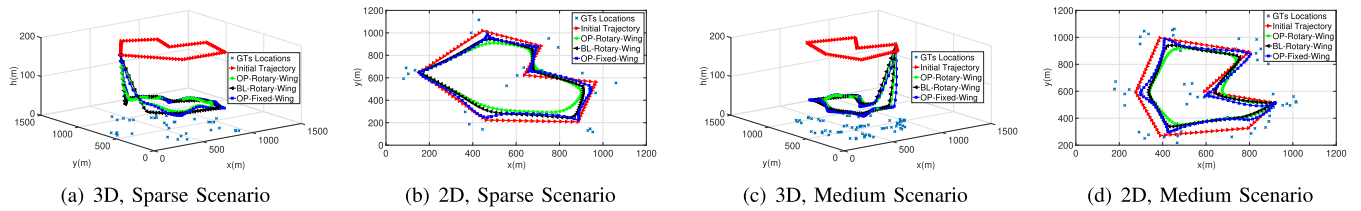
| (a) 3D, Sparse Scenario | (b) 2D, Sparse Scenario | (c) 3D, Medium Scenario | (d) 2D, Medium Scenario |

FIGURE 2. TSP trajectory of fixed-wing/rotary-wing UAV led by different solutions.

| (a) 3D, Sparse Scenario | (b) 3D, Medium Scenario | (c) 3D, Densified Scenario |

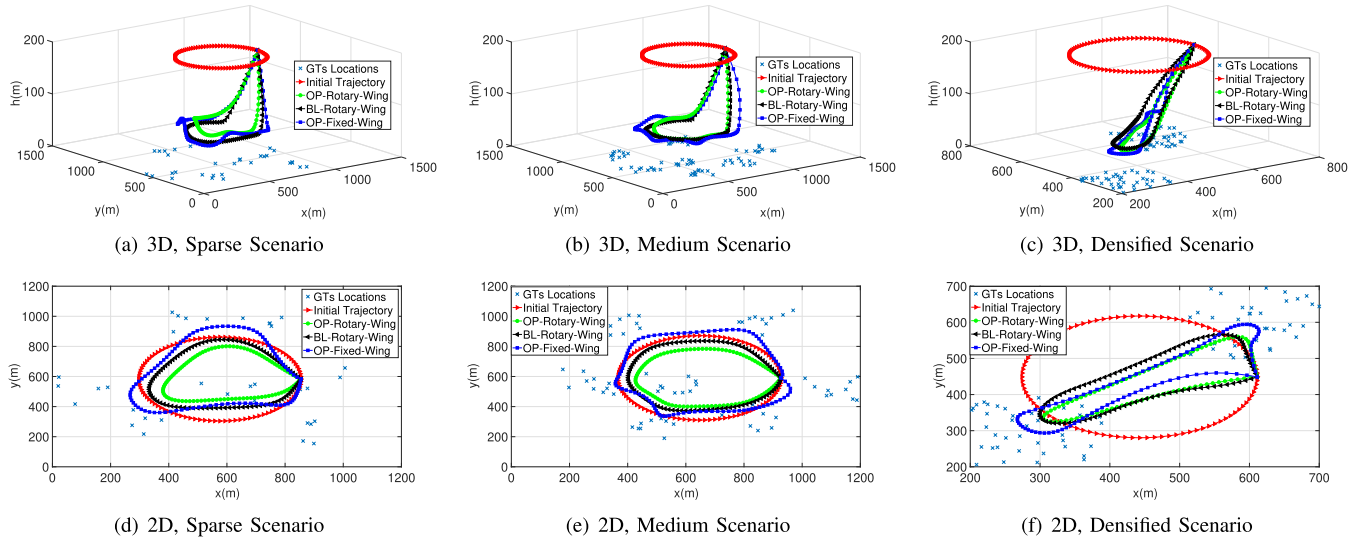| (d) 2D, Sparse Scenario | (e) 2D, Medium Scenario | (f) 2D, Densified Scenario |

FIGURE 3. Circular trajectory of fixed-wing/rotary-wing UAV led by different solutions.

TABLE 1. System configuration on simulations.

Parameter	Value
Bandwidth B, Uplink power: p_k	2GHz, 5mW
Pathloss: a, b, η_{Los}, η_{NLos}	9.61, 0.16, 1, 20 [40]
Fixed-wing: c_1, c_2, c_3	$9.26 * 10^{-4}$, 2250, 3.33 [32]
Rotary-wing:$U_{tip}, v_0, d_0, s, \rho, G$	120, 4.3, 0.6, 0.05, 1.225, 0.503 [31]
P_0, P_1, P_2	$\frac{12*30^3*0.4^3}{8}\rho s G$, $\frac{1.1*20^{3/2}}{\sqrt{2\rho G}}$,11.46
Δ_{max}^h, Δ_{max}^v	35m, 20m
fixed-wing: V_{max}^h, V_{min}^h, V_{max}^v	35m/s, 5m/s, 20m/s
Rotary-wing: V_{max}^h, V_{min}^h, V_{max}^v	20m/s, 0m/s, 15m/s
Cache: C^c, offloading: C^o	2GHz, 1Gb
Noisy density: N_0	-169dBm/Hz
Energy on computation: φ, ϑ	10^{-9}, 3
GT task:T_k, D_k	200~300s, 1~5 Mb
GT task: F_k	$6 \times 10^6 \sim 8 \times 10^6$ CPU cycles
Computation capacity: f_k^o	$2 \times 10^5 \sim 4 \times 10^5$ CPU cycles
Computation capacity: f_k^l	$10^5 \sim 1.6 \times 10^5$ CPU cycles
T_{max}, $\widehat{D}$	5 minutes, 3Km
UAV height: h_m^0, h_{min}, h_{max}	200m, 50m, 200m
Number of GTs: K	35(sparse),70(medium),80(densified)

We compare the proposed solution to a baseline solution with respect to the energy consumptions, task latency, and data transmission, to validate the benefit of our proposed

solution. The baseline solution in practice employs a greedy approach to find the trajectory-task-cache result, which does not consider energy-efficiency of the UAV. Instead, the baseline solution aims to control GTs to maximally offload and cache tasks to the UAV, and has UAV trajectory closer to GTs to facilitate the task offloading. As a result, the baseline solution intends to benefit the GTs in a maximum extend while sacrifice the UAV endurance. Through this comparison, we thus can validate the benefit of our proposed solution on energy-efficiency.

In rest part of this section, we will demonstrate different UAV trajectories led by the two compared solutions in Fig.2 and Fig.3. and the task latencies, data transmissions, energy consumptions led by the solutions in Fig.4 and Fig.5. Afterwards, the task-cache strategies led by the two solutions will be denoted in Fig.6 and Fig.7, which are correlated to each designed UAV trajectories. Then we will give a conclusion according to the numerical results in the end.

In Fig.2 and Fig.3, the UAV trajectory in 3D and 2D spaces led by different solutions are demonstrated, where the UAV in fixed-wing or rotary-wing is initialized to fly a circular trajectory and a TSP trajectory respectively. As shown in Fig.2 and Fig.3, the baseline solution (BL) controls the rotary-wing UAV flying geographically close to GTs to get better links to GTs. In contrast, the proposed optimization solution (OP)
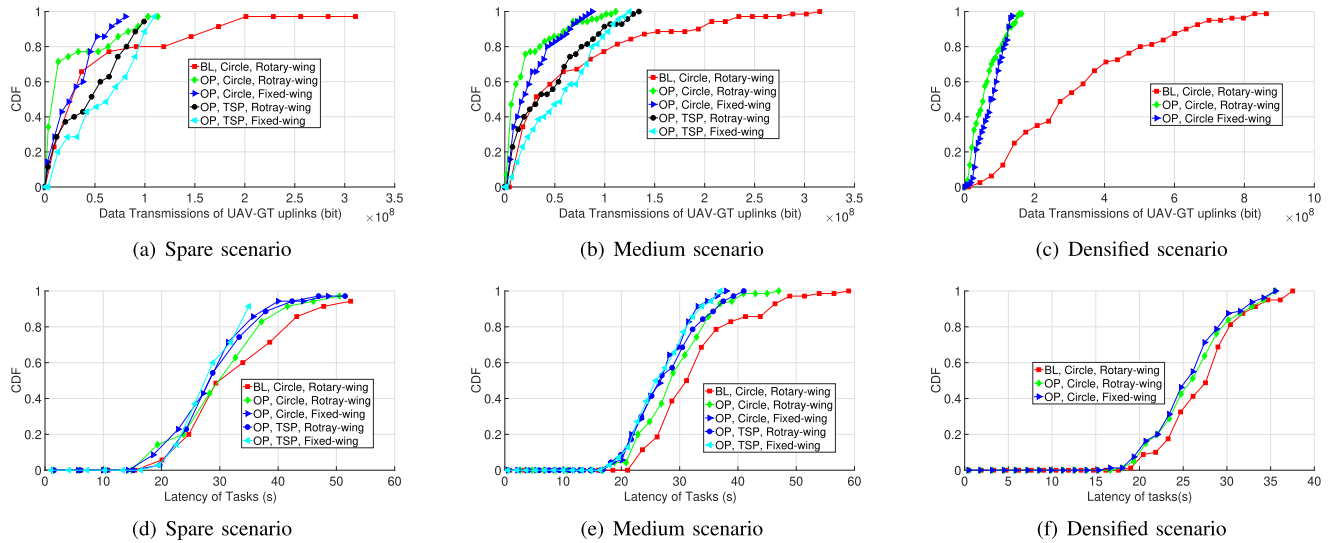
FIGURE 4. Data transmissions and latencies of GT tasks led by different solutions.

controls the rotary-wing UAV to adopt trajectories flying neither too close to GTs nor too away from GTs. Therefore, with the proposed solution, the UAV can better balance its energy consumption and services to work with energy efficiency. In Fig.2 and Fig.3, the trajectories of the fixed-wing UAV led by the baseline solution are not demonstrated, due to the limited space of this paper. However, it will be the same result as the case of rotary-wing UAV. In addition, Fig.2 and Fig.3 show that the fixed-wing UAV adopts a trajectory with longer distance compared to the rotary-wing UAV. This is because the fixed-wing UAV cannot hover statically and requests a minimum velocity to stay aloft. Also this is because the upper bound of the horizontal velocity (34 m/s) of fixed-wing UAV is higher than the rotary-wing UAV (20 m/s), leading to the fixed-wing UAV possibly adopting higher flying speed and longer distances compared to the rotary-wing UAV. On the other hand, in vertical dimension, Fig.2 and Fig.3 show that the UAV normally descends from its original height to approach to GTs for better links. Compared to the baseline solution, the optimization solution normally controls the UAV to descend less distance, thus helps save the energy consumption of the UAV in the vertical dimension. Also the proposed solution controls the UAV to descend/ascend in a smoother way than the baseline solution that causes UAV to descend/ascend dramatically.

To validate the effect of different solutions, the data transmissions and latencies of each GT task are compared in Fig.4. Also, the service and propulsion related energy consumptions of different solutions are shown in Fig.5. These numerical results correspond to the UAV trajectories in Fig.2 and Fig.3. According to the results shown in Cumulative Distribution Function (CDF) form in Fig.4 and Fig.5, the baseline solution causes significant energy consumption, and leads to higher amount of data transmissions via uplinks. However, the baseline solution leads to higher latencies of GT tasks in contrast

to the proposed solution. This is because the baseline solution greedily offloads tasks to the UAV, which causes high latency during data transmissions and the heavy load of the UAV. The proposed solution instead consumes much less energy on service and UAV propulsion, and supports each task with lower latency. Therefore, we validate that the proposed solution works energy efficiently.

In addition, it can be found in Fig.4, the UAV flying a TSP based trajectory can support higher amount of data transmissions than the circle trajectory, as the UAV in TSP based trajectory will fly closer to the GTs compared to the circular trajectory. Therefore, the UAV in TSP based trajectory supports lower task latency as compared to the circular trajectory. Moreover, in Fig.5, the UAV almost consumes the same level of energy on services while in fixed-wing or rotary-wing style, and it is quite obvious that the TSP trajectory will cause less service related energy consumption in medium GT scenario, due to the TSP trajectory can better supports the UAV-GT links. Considering propulsion energy, the fixed-wing UAV will consume more propulsion energy compared to the rotary-wing UAV. Also, the UAV in TSP based trajectory will consume more propulsion energy compared to the circular trajectory, as the TSP based trajectory involves longer flight distance.

To validate the task offloading and cache strategies, Fig.6 shows the results. As demonstrated, the baseline solution adopts a greedy way, which causes more task offloaded to the UAV in contrast to the proposed solution. Thus, because the baseline solution forces the UAV to take more responsibility for the GTs and involves more data transmissions, the task latencies and energy consumptions led by the baseline solution therefore will be both higher than the proposed solution. Also, it can be found that the GTs will offload very similar amount of tasks to the fixed-wing and rotary-wing UAVs by the proposed solution. This is because the offloading
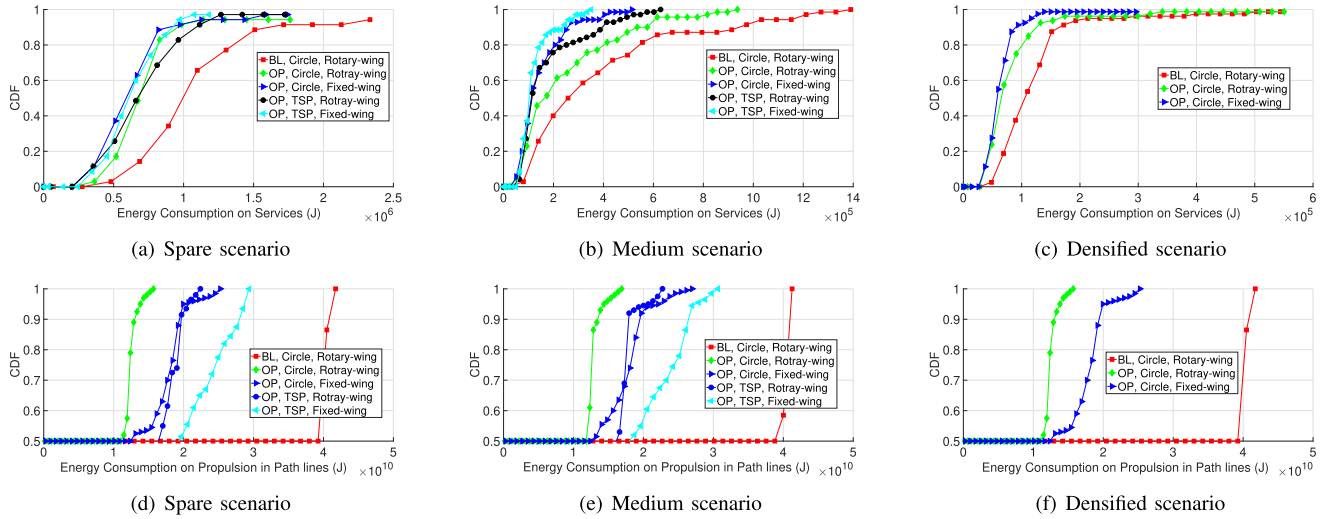
(a) Spare scenario   (b) Medium scenario   (c) Densified scenario

(d) Spare scenario   (e) Medium scenario   (f) Densified scenario

FIGURE 5. Energy consumption on service and UAV propulsion led by different solutions.

(a) Spare scenario   (b) Medium scenario   (c) Densified scenario

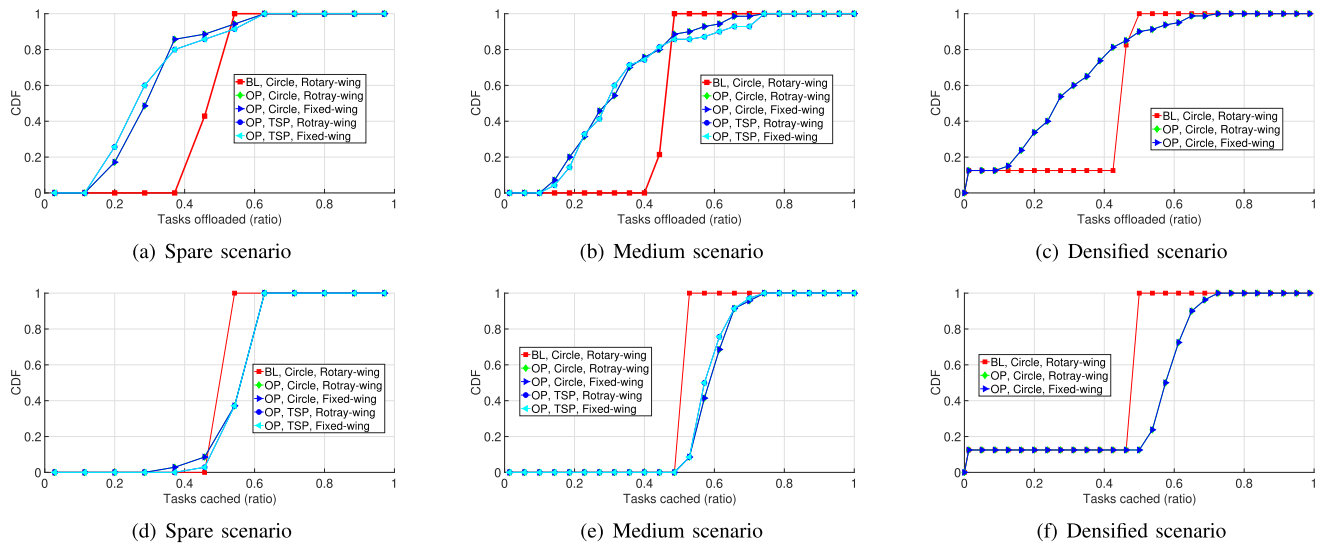(d) Spare scenario   (e) Medium scenario   (f) Densified scenario

FIGURE 6. Task offloading and cache led by different solutions.

strategies adopted by the proposed solution are not sensitive to the trajectories and styles of the UAV. Instead, the strategies are to minimize the computing and data transmission related energy, thus they are more sensitive to GTs' energy status. Moreover, it is shown that the proposed solution more efficiently caches GT tasks in the UAV to save redundant data transmission in contrast to the baseline solution. This is because the baseline solution controls more tasks offloaded to the UAV and causes congestion. Also the proposed solution controls the GT tasks cached to the UAV in similar level of amount with respect to different UAV trajectories and styles. In conclusion, we validate that the proposed solution outperforms the baseline solution, and can support energy-efficient performance of the UAV-enabled mobile edge networks.

V. CONCLUSION

This paper tries to optimize UAV trajectory jointly with the task offloading and cache to realize energy-efficient

performance of the UAV-enabled mobile edge network. The joint problem considers the energy model and flight constraints of the different fixed-wing and rotary-wing UAVs in 3D spaces. The problem is multiple constrained and non-convex, and difficult to be solved optimally. In this paper, we obtain the sub-optimal solution of the joint problem through a block coordinate descend algorithm by leveraging successive convex optimization technique. We validate the proposed solution through simulation and compare it to the baseline solution, considering fixed-wing/rotary-wing UAV working in sparse, medium and densified GT scenarios. According to the numerical results, the proposed solution can intensively save the energy consumptions of the UAV on propulsion and services, which ensuring the GT tasks finished before the completion deadline. With such energy-efficient performance achieved, the UAV-enabled MEC thus can have its power constraint released, thus can better support CPS applications. In future, we intend to extend the energy model

of fixed-wing/rotary-wing UAV to denote how the velocity affecting the propulsion energy of UAV in vertical dimension. Then we try to explore high quality solution to lead to energy-efficient performance of the UAV-enabled MEC, based on the extended energy model. In addition, we intend to consider the system with multiple UAVs, and solve its energy consumption related problems using deep reinforcement learning technique.

APPENDIX
PROVE THEOREM IN (13)

Given a UAV trajectory, assume the data rate of the uplink from k-th GT to the UAV is R_k. Then according to (9) and (11.4), one has $r_k = \frac{(a_k - a_k x_k)D_k}{(a_k - a_k x_k)\frac{F_k}{f_k^l} - \left(a_k \frac{F_k}{f_k^o} - x_k \frac{F_k}{f_k^l}\right) + T_k - \frac{F_k}{f_k^l}}$ as the requested data rate of the UAV-GT k link. Then r_k is constrained to $r_k \leq R_k$ so as to guarantee GT task U_k executed before its completion deadline. Then one has

$$R_k \geq \frac{(a_k - a_k x_k)D_k}{(a_k - a_k x_k)\frac{F_k}{f_k^l} - \left(a_k \frac{F_k}{f_k^o} - x_k \frac{F_k}{f_k^l}\right) + T_k - \frac{F_k}{f_k^l}}$$

$$\geq \frac{(a_k - a_k x_k)D_k}{(a_k - a_k x_k)\frac{F_k}{f_k^l} - (a_k - a_k x_k)\frac{F_k}{f_k^o} + T_k - \frac{F_k}{f_k^l}} \quad (29)$$

where the inequality is due to the fact that the UAV will execute each offloaded task quicker than the task being conducted by GT locally, i.e. $\frac{a_k F_k}{f_k^o} \leq \frac{F_k}{f_k^l}$.

Then, considering the possible caused latency ($\frac{D_k}{R_k} + \frac{F_k}{f_k^o}$) during task offloading, we only need to consider three possible scenarios of (29): 1) if ($\frac{D_k}{R_k} + \frac{F_k}{f_k^o}$) $\geq T_k > \frac{F_k}{f_k^l}$, (29) can be converted as $a_k(1 - x_k) \leq \frac{T_k - \frac{F_k}{f_k^l}}{\frac{D_k}{R_k} + \frac{F_k}{f_k^o} - \frac{F_k}{f_k^l}}$, where

$0 \leq \frac{T_k - \frac{F_k}{f_k^l}}{\frac{D_k}{R_k} + \frac{F_k}{f_k^o} - \frac{F_k}{f_k^l}} \leq 1$; 2) if $T_k \geq (\frac{D_k}{R_k} + \frac{F_k}{f_k^o}) > \frac{F_k}{f_k^l}$, (29)

can be converted as $a_k(1 - x_k) \leq 1 \leq \frac{T_k - \frac{F_k}{f_k^l}}{\frac{D_k}{R_k} + \frac{F_k}{f_k^o} - \frac{F_k}{f_k^l}}$; 3) if

$T_k > \frac{F_k}{f_k^l} \geq (\frac{D_k}{R_k} + \frac{F_k}{f_k^o})$, (29) can be converted as $1 \geq$

$a_k(1 - x_k) \geq 0 \geq \frac{T_k - \frac{F_k}{f_k^l}}{\frac{D_k}{R_k} + \frac{F_k}{f_k^o} - \frac{F_k}{f_k^l}}$. To this end, we can conclude

that one always has $a_k(1 - x_k) \leq \begin{cases} H_k & \text{if } H_k > 0 \\ 1 & \text{if } H_k = 0 \end{cases}$ in all the

three cases, where $H_k = \min\left\{\max\left\{\frac{T_k - \frac{F_k}{f_k^l}}{\frac{D_k}{R_k} + \frac{F_k}{f_k^o} - \frac{F_k}{f_k^l}}, 0\right\}, 1\right\}$.
Then the theorem is proofed.

REFERENCES

[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. Design Autom. Conf.*, Anaheim, CA, USA, Jun. 2010, pp. 731–736.

[2] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. 11th IEEE Int. Symp. Object Compon.-Oriented Real-Time Distrib. Comput. (ISORC)*, Orlando, FL, USA, May 2008, pp. 363–369.

[3] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019. doi: 10.1109/JIOT.2018.2887086.

[4] Q. Wu, L. Liu, and R. Zhang, "Fundamental trade-offs in communication and trajectory design for UAV-enabled wireless network," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 36–44, Feb. 2019. [Online]. Available: https://arxiv.org/abs/1805.07038

[5] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[6] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[7] S. Jeong, O. Simeone, and J. Kang, "Mobile cloud computing with a UAV-mounted cloudlet: Optimal bit allocation for communication and computation," *IET Commun.*, vol. 11, no. 7, pp. 969–974, May 2017.

[8] 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.

[9] *Mobile Edge Computing Introductory Technical White Paper(PDF)*, ETSI, Sophia Antipolis, France, Oct. 2015. [Online]. Available: http://etsi.org

[10] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[11] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[12] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[13] Y. Du, K. Wang, K. Yang, and G. Zhang, "Energy-efficient resource allocation in UAV based MEC system for IoT devices," in *Proc. IEEE Global Commun. Conf. (GLOBALCOM)*, Dec. 2018, pp. 1–6.

[14] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Apr. 2019.

[15] J. Xiong, H. Guo, and J. Liu, "Task offloading in UAV-aided edge computing: Bit allocation and trajectory optimization," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 538–541, Mar. 2019.

[16] M. Hua, Y. Wang, C. Li, Y. Huang, and L. Yang, "UAV-aided mobile edge computing systems with one by one access scheme," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 3, pp. 664–678, Sep. 2019. doi: 10.1109/TGCN.2019.2910590.

[17] Y. Qian, F. Wang, J. Li, L. Shi, K. Cai, and F. Shu, "User association and path planning for UAV-aided mobile edge computing with energy restriction," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1312–1315, Oct. 2019. doi: 10.1109/LWC.2019.2913843.

[18] P. Liu, G. Xu, K. Yang, K. Wang, and X. Meng, "Jointly optimized energy-minimal resource allocation in cache-enhanced mobile edge computing systems," *IEEE Access*, vol. 7, pp. 3336–3347, 2019.

[19] Y. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, "Energy efficient task caching and offloading for mobile edge computing," *IEEE Access*, vol. 6, pp. 11365–11373, 2018.

[20] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.

[21] X. Cao, J. Xu, and R. Zhang, "Mobile edge computing for cellular-connected UAV: Computation offloading and trajectory optimization," Mar. 2018, *arXiv:1803.03733*. [Online]. Available: https://arxiv.org/pdf/1803.03733v1.pdf

[22] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Empirical power consumption model for UAVs," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Chicago, IL, USA, Aug. 2018, pp. 1–5.

[23] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[24] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[25] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," 2018, *arXiv:1808.01977*. [Online]. Available: https://arxiv.org/abs/1808.01977

[26] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 6, no. 3, pp. 760–770, Jul./Sep. 2018.

[27] H. Mei, K. Wang, and K. Yang, "Multi-layer cloud-RAN with cooperative resource allocations for low-latency computing and communication services," *IEEE Access*, vol. 5, pp. 19023–19032, 2017.

[28] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[29] W. Jiang, G. Feng, S. Qin, and Y. Liu, "Multi-agent reinforcement learning based cooperative content caching for mobile edge networks," *IEEE Access*, vol. 7, pp. 61856–61867, 2019.

[30] H. Mei, K. Wang, and K. Yang, "Joint cache content placement and task offloading in C-RAN enabled by multi-layer MEC," *Sensors*, vol. 18, no. 6, p. 1826, 2018.

[31] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[32] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[33] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[34] Q. Wu and R. Zhang, "Common throughput maximization in UAV-enabled OFDMA systems with delay consideration," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6614–6627, Dec. 2018.

[35] Y. Guo, S. Yin, and J. Hao, "Resource allocation and 3-D trajectory design in wireless networks assisted by rechargeable UAV," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 781–784, Jun. 2019. doi: 10.1109/LWC.2019.2892721.

[36] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[37] X. Xu, Y. Zeng, Y. L. Guan, and R. Zhang, "Overcoming endurance issue: UAV-enabled communications with proactive caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1231–1244, Jun. 2018.

[38] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: Trajectory optimization under connectivity constraint," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

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

[40] C. Zhan and H. Lai, "Energy minimization in Internet-of-Things system based on rotary-wing UAV," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1341–1344, Oct. 2019. doi: 10.1109/LWC.2019.2916549.

[41] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[42] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–5.

[43] 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. 3192–3207, Jun. 2019.

[44] M. Grant and S. Boyd. (2016). *CVX: MATLAB Software for Disciplined Convex Programming*. [Online]. Available: http://cvxr.com/cvx

[45] J. Chen, F. Ye, and Y. Li, "Travelling salesman problem for UAV path planning with two parallel optimization algorithms," in *Proc. Progr. Electromagn. Res. Symp.—Fall (PIERS-FALL)*, Singapore, Nov. 2017, pp. 832–837.

[46] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, vol. 2. Philadelphia, PA, USA: SIAM, 2001.

HAIBO MEI received the B.Sc. and M.Sc. degrees from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, in 2005 and 2008, respectively, and the Ph.D. degree from the School of Electronic Engineering and Computer Science, Queen Mary University of London (QMUL), U.K., in 2012. He was a Postdoctoral Research Assistant with QMUL and a Senior Research and Development Engineer with Securus Software Ltd., U.K. He is currently a Lecturer with the University of Electronic Science and Technology of China. His research interests include resource efficiency and self-organization of wireless communications, intelligent transportation systems, and mobile cloud computing.

KEZHI WANG received the B.E. and M.E. degrees from the College of Automation, Chongqing University, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Warwick, U.K., in 2015. He was a Senior Research Officer with the University of Essex, U.K. He is currently a Lecturer with the Department of Computer and Information Sciences, Northumbria University. His research interests include wireless communication, signal processing, and mobile cloud computing.

DONGDAI ZHOU received the B.Eng. and M.Sc. degrees from the Changchun University of Science and Technology, China, in 1992 and 1997, respectively, and the Ph.D. degree from Jilin University, China, in 2001. He is currently a Professor with the School of Information Science and Technology, Northeast Normal University, China. His research interests include software architecture and software code auto-generation, especially the architecture design and codeless development of e-learning software.

KUN YANG (SM'08) received the B.Sc. and M.Sc. degrees from the Computer Science Department, Jilin University, China, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, University College London (UCL), U.K. He is currently the Chair Professor with the School of Computer Science and Electronic Engineering, University of Essex, and leading the Network Convergence Laboratory (NCL), U.K. He is also a Professor with the University of Electronic Science and Technology of China (UESTC). He manages research projects funded by various sources, such as U.K. EPSRC, EU FP7/H2020, and industries. He has published more than 200 articles. His main research interests include wireless networks, network convergence, the future Internet technology, data and energy cooperation, and mobile computing. He serves on the editorial boards of both the IEEE and non-IEEE journals.

• • •