IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621306

An Online Joint Optimization Approach for QoE Maximization in UAV-Enabled Mobile Edge Computing

Long He[†], Geng Sun[†*], Zemin Sun[†], Pengfei Wang[‡], Jiahui Li[†], Shuang Liang[§], Dusit Niyato[¶]
[†]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[‡]School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[§]School of Information Science and Technology, Northeast Normal University, Changchun 130024, China
[¶]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore
E-mails:{helong0517, lijiahui0803}@foxmail.com, {sungeng, sunzemin}@jlu.edu.cn,
wangpf@dlut.edu.cn, liangshuang@nenu.edu.cn, dniyato@ntu.edu.sg
[*]Corresponding author: Geng Sun

***Abstract*—Given flexible mobility, rapid deployment, and low cost, unmanned aerial vehicle (UAV)-enabled mobile edge computing (MEC) shows great potential to compensate for the lack of terrestrial edge computing coverage. However, limited battery capacity, computing and spectrum resources also pose serious challenges for UAV-enabled MEC, which shorten the service time of UAVs and degrade the quality of experience (QoE) of user devices (UDs) without effective control approach. In this work, we consider a UAV-enabled MEC scenario where a UAV serves as an aerial edge server to provide computing services for multiple ground UDs. Then, a joint task offloading, resource allocation, and UAV trajectory planning optimization problem (JTRTOP) is formulated to maximize the QoE of UDs under the UAV energy consumption constraint. To solve the JTRTOP that is proved to be a future-dependent and NP-hard problem, an online joint optimization approach (OJOA) is proposed. Specifically, the JTRTOP is first transformed into a per-slot real-time optimization problem (PROP) by using the Lyapunov optimization framework. Then, a two-stage optimization method based on game theory and convex optimization is proposed to solve the PROP. Simulation results validate that the proposed approach can achieve superior system performance compared to the other benchmark schemes.**

I. Introduction

WITH artificial intelligence and wireless communications development, many intelligent applications with strict requirements on computing resources and latency have emerged explosively [1], such as real-time video analysis [2], virtual reality/augmented reality [3], and interactive online games [4]. However, the limited battery capacity and computing capability of user devices (UDs) make it difficult to maintain a high-level quality of experience (QoE) for these intelligent applications [5]. To overcome this challenge, mobile edge computing (MEC) has emerged as a promising paradigm to offer cloud computing resources in close proximity to UDs [6], [7]. Specifically, UDs can offload latency-sensitive and computation-hungry tasks to edge servers to improve the QoE. Equipped with cloud computing capabilities, the edge servers can concurrently provide real-time and energy-efficient computing services for multiple UDs. However, conventional terrestrial MEC still faces the challenges of limited network coverage and high deployment cost due to the dependence on ground infrastructures, especially in remote areas [8].

The limitations of conventional terrestrial MEC have prompted a paradigm shift toward UAV-enabled MEC due to the line-of-sight (LoS) communication, high maneuverability, and flexible deployment of UAVs [9]–[11]. First, the high probability LoS links of UAVs boost the communication coverage, network capacity, and reliable connectivity [12], [13]. Furthermore, their flexible mobility enables rapid and on-demand deployment, especially in distant areas where terrestrial infrastructures are unavailable. Besides, the integration of UAV and MEC offers flexible computing capabilities to improve the QoE of UDs.

However, several fundamental challenges should be overcome to fully exploit the benefits of UAV-enabled MEC. ***i) Resource Allocation.*** Various tasks of UDs are generally heterogeneous and time-varying, and they have stringent requirements for the offloading service. However, the limited computing resources and scarce spectrum resources of UAV-enabled MEC and the stringent demands of UDs could lead to the competition for resources inside the MEC server, especially during peak times. Thus, under resource constraints, it is challenging for the MEC server to determine an efficient resource allocation strategy to meet the demands of various tasks. ***ii) Task Offloading.*** The offloading decision of each UD depends not only on its own offloading demand but also on the offloading decisions of the other UDs, which makes the offloading decisions among UDs coupling and complex. ***iii) Trajectory Planning.*** Although the mobility of UAVs increases the flexibility and elasticity of MEC, it also brings significant difficulties in UAV trajectory planning. ***iv) Energy Constraint.*** The limited onboard battery capacity of UAVs leads to finite service time, which makes it challenging to balance the service time of UAVs and the QoE of UDs. In addition, under the constraints of UAV's resources and energy, the resource allocation strategy of UAVs, the task offloading decisions of UDs, and the trajectory planning of UAVs have mutual effects on each other, leading to the complexity of the decision-making process.

To overcome the aforementioned challenges, we propose an online approach for joint optimization of task offloading, resource allocation, and UAV trajectory planning to maximize the QoE of UDs under the UAV energy consumption constraint. The main contributions are summarized as follows:

- ***System Architecture.*** We consider a stochastic UAV-enabled MEC system with energy and resource constraints consisting of a UAV and multiple ground UDs. Specifically, the UAV is employed as an aerial edge server relying on limited battery capacity, computing and communication resources to provide computing services to UDs with time-varying computation requirements and dynamic mobility.
- ***Problem Formulation.*** We formulate a novel joint task offloading, resource allocation, and UAV trajectory planning optimization problem (JTRTOP) with the aim of maximizing the QoE of UDs under the UAV energy consumption constraint. Specifically, the QoE of UDs is theoretically measured by synthesizing the completion delay of the tasks and energy consumption of UDs.
- ***Algorithm Design.*** Since the JTRTOP not only requires future information but is also non-convex and NP-hard, we propose an online joint optimization approach (OJOA) to solve the problem. Specifically, we first transform the JTRTOP into a per-slot real-time optimization problem (PROP) by using the Lyapunov optimization framework. Then, we propose a two-stage method to optimize the task offloading, resource allocation, and UAV position of PROP by using convex optimization and game theory.
- ***Validation.*** Both theoretical analysis and simulation experiments are performed to verify the effectiveness and performance of the proposed OJOA. Specifically, theoretical analysis demonstrates that the OJOA not only satisfies the UAV energy consumption constraint but also converges to a sub-optimal solution in polynomial time. Moreover, simulation results indicate that the proposed OJOA outperforms other benchmark schemes.

The remainder of the work is organized as follows. Section II summarizes the related work. Section III details the relevant system models and problem formulation. Section IV describes the Lyapunov-based problem transformation. Section V presents the two-stage optimization algorithm and theoretical analysis. In Section VI, simulation results are displayed and analyzed. Finally, Section VII concludes the overall paper.

II. Related Work

Most existing studies on UAV-enabled MEC are devoted to the design of offline algorithms to plan the entire task offload, resource allocation, and UAV trajectory, which assume that the locations of UDs are invariant and the computing requirements of UDs are fixed or known in advance [14]–[16]. However, many edge computing scenarios change dynamically over time, such as real-time video analysis and interactive online games, which means that the computing tasks arrive stochastically, the computing requirements of UDs are time-varying and the UDs are dynamically mobile. Therefore, it is necessary to design real-time decision-making algorithms without future information.

There are also some works studying real-time decision-making. For example, Yang et al. [17] studied the UAV-enabled MEC system with random task arrival and user mobility. Specifically, the UAV trajectory and resource allocation were decided in real time to minimize the average energy consumption of all users through online algorithms based on Lyapunov optimization. Considering the time-varying computing requirements of user equipment, Wang et al. [18] jointly optimized the user association, resource allocation and trajectory of UAVs with the aim of minimizing energy consumption of all user equipment. To minimize the average power consumption of the system with randomly arriving user tasks, Hoang et al. [19] developed a Lyapunov-guided deep reinforcement learning framework. Zhou et al. [20] proposed an alternating optimization-based algorithm by leveraging the Lyapunov optimization approach and dependent rounding technique to minimize the service delay.

In practice, due to the limited energy and computing resources of UDs, task completion delay and energy consumption are important indicators to measure the QoE of UDs. However, the abovementioned works mainly focus on minimizing the task completion delay and energy consumption of users (or the whole system) separately, which could not provide a high-level QoE for users. Furthermore, these works consider resource allocation from either the communication or the computation aspects, which may lead to severe performance degradation in practical UAV-enabled MEC systems where both communication and computing resources are insufficient. Motivated by these issues, in this work, we consider a stochastic UAV-enabled MEC system with time-varying computation requirements and dynamic mobility of UDs to minimize the user energy consumption and task completion latency simultaneously. Furthermore, the computing and communication resource allocation are jointly optimized.

III. System Model and Problem Formulation

As illustrated in Fig. 1, we considered a UAV-enabled MEC system that consists of a rotary-wing UAV u and M UDs with the set $\mathcal{M} = \{1, 2, \ldots, M\}$. Equipped with MEC capability, the UAV is employed as an aerial edge server relying on limited battery capacity to provide computing offloading services to the UDs within a finite system timeline. Moreover, we discretize the system timeline into equal T time slots [21], i.e., $t \in \mathcal{T} = \{1, 2, \ldots, T\}$, wherein each slot duration is denoted as τ.

A. Basic Model

UD Model. We assume that each UD generates one computing task per time slot [18], [22]. For UD $m \in \mathcal{M}$, the UD's attributes at time slot t can be characterized as $\mathbf{St}_m^{\mathrm{UD}}(t) = \big(f_m^{\mathrm{UD}}, \mathbf{\Phi}_m(t), \mathbf{P}_m(t)\big)$, where f_m^{UD} denotes the local computing

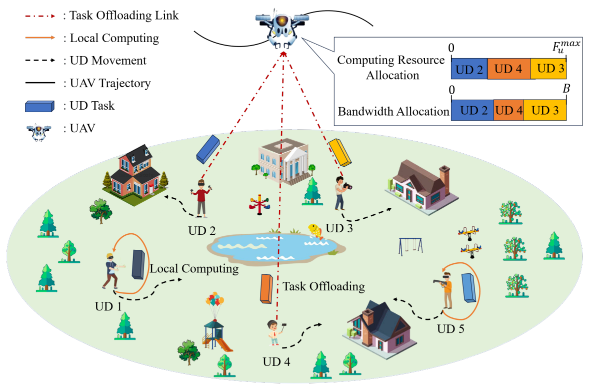


Fig. 1. The UAV-enabled MEC consists a UAV and multiple ground UDs. The UAV provides computing services to UDs by allocating communication and computing resources. Each UD independently decides to compute its task locally or offload the task to the UAV.

capability of UD m. The computing task generated by UD m is characterized as $\mathbf{\Phi}_m(t) = \{D_m(t), \eta_m(t), T_m^{\max}(t)\}$ at time slot t, wherein $D_m(t)$ represents the input data size (in bits), $\eta_m(t)$ denotes the computation intensity (in cycles/bit), and $T_m^{\max}(t)$ is the maximum tolerable delay. $\mathbf{P}_m(t) = [x_m(t), y_m(t)]$ represents the location coordinates of UD m at time slot t. Similar to [23], [24], the mobility of UDs is modeled as a Gauss-Markov mobility model, which is widely employed in cellular communication networks [25]. Specifically, the velocity of UD m at time slot $t+1$ are updated as follows:

$$\mathbf{v}_m(t+1) = \alpha\mathbf{v}_m(t) + (1-\alpha)\overline{\mathbf{v}}_m + \sqrt{1-\alpha^2}\mathbf{w}_m(t), \quad (1)$$

where $\mathbf{v}_m(t) = (v_m^x(t), v_m^y(t))$ denotes the velocity vector at time slot t. α represents the memory level, which reflects the temporal-dependent degree and $\overline{\mathbf{v}}_m$ is the asymptotic means of velocity. $\mathbf{w}_m(t)$ is the uncorrelated random Gaussian process $N(0, \sigma_m^2)$, where σ_m denotes the asymptotic standard deviation of velocity. Therefore, the mobility of UD m can be updated as follows:

$$\mathbf{P}_m(t+1) = \mathbf{P}_m(t) + \mathbf{v}_m(t)\tau. \quad (2)$$

UAV Model. UAV u is characterized by $\mathbf{St}^u(t) = (\mathbf{P}_u(t), H, F_u^{\max}, B)$, wherein $\mathbf{P}_u(t) = [x_u(t), y_u(t)]$ and H represent the horizontal coordinate and flight height of the UAV at time slot t, respectively. $F_u^{\max}$ represents the total computing resources and B denotes the total bandwidth resources.

Decision Variables. The following decisions need to be made jointly. *i) Task Offloading Decision.* For task $\mathbf{\Phi}_m(t)$, we define a binary variable $a_m(t)$ to represent the offloading decision of UD m at time slot t, where $a_m(t) = 0$ indicates that the task is processed locally, and $a_m(t) = 1$ indicates that the task is offloaded to the UAV for processing. *ii) Resource Allocation Decision.* For the UAV, the resources allocated to task $\mathbf{\Phi}_m(t)$ are denoted as $\{F_m(t), w_m(t)\}$ at time slot t, where $F_m(t)$ is the amount of allocated computing resources and $w_m(t)$ is the proportion of allocated bandwidth resources. *iii) UAV Trajectory Planning.* For the UAV, trajectory planning can be expressed as a sequence of optimal positions for each time slot, i.e., $\mathbf{P}_u = \{\mathbf{P}_u(t)\}_{t\in\mathcal{T}}$.

B. Communication Model

The probabilistic line-of-sight (LoS) channel model is employed to model the communication between the UAV and UDs [26]. First, the LoS probability $P_{m,u}^{\text{LoS}}(t)$ between UD m and the UAV at time slot t can be defined as [27]

$$P_{m,u}^{\text{LoS}}(t) = \frac{1}{1+\xi_1\exp(-\xi_2(\theta_{m,u}(t)-\xi_1))}, \quad (3)$$

where ξ_1 and ξ_2 are constants depending on the propagation environment, $\theta_{m,u}(t) = \frac{180}{\pi}\arcsin\frac{H}{d_{m,u}(t)}$ denotes the elevation angle and $d_{m,u}(t)$ represents the straight-line distance between UD m and the UAV. Similar to [17], [28], the channel power gain can be calculated as

$$\begin{aligned} g_{m,u}(t) &= P_{m,u}^{\text{LoS}}(t)\beta_0 d_{m,u}^{-\tilde{\mu}}(t) + (1-P_{m,u}^{\text{LoS}}(t))\kappa\beta_0 d_{m,u}^{-\tilde{\mu}}(t) \\ &= \tilde{P}_{m,u}^{\text{LoS}}(t)\beta_0 d_{m,u}^{-\tilde{\mu}}(t), \end{aligned} \quad (4)$$

where $\tilde{P}_{m,u}^{\text{LoS}}(t) \triangleq P_{m,u}^{\text{LoS}}(t) + (1-P_{m,u}^{\text{LoS}}(t))\kappa$, κ is the additional attenuation factor, β_0 denotes the channel gain at the reference distance 1 m, and $\tilde{\mu}$ is the path loss exponent. Therefore, the spectral efficiency of UD m can be expressed as

$$r_{m,u}(t) = \log_2\left(1 + \frac{\phi_m(t)}{(||\mathbf{P}_u(t)-\mathbf{P}_m(t)||^2+H^2)^{\mu}}\right), \quad (5)$$

where $\phi_m(t) = \frac{P_m\beta_0\tilde{P}_{m,u}^{\text{LoS}}(t)}{N_0}$, $\mu = \frac{\tilde{\mu}}{2}$, P_m is the transmission power of UD m, and N_0 represents the noise power.

Moreover, the widely used orthogonal frequency-division multiple access (OFDMA) is employed in the communication models. Therefore, the communication rate of UD m at time slot t can be presented as [29]

$$R_{m,u}(t) = w_m(t)Br_{m,u}(t), \quad (6)$$

C. Computation Model

For task $\mathbf{\Phi}_m(t)$ generated by UD m, the task can be processed either locally on the UD or remotely on the UAV, which is determined by the UD's offloading decision $a_m(t)$.

Local Computing. UD m processes task $\mathbf{\Phi}_m(t)$ locally (i.e., $a_m(t) = 0$). The local completion latency of the task at time slot t can be calculated as

$$T_m^{\text{loc}}(t) = \frac{\eta_m(t)D_m(t)}{f_m^{\text{UD}}}, \quad (7)$$

Accordingly, the energy consumption of UD m to execute task $\mathbf{\Phi}_m(t)$ locally at time slot t is calculated as [15]

$$E_m^{\text{loc}}(t) = k(f_m^{\text{UD}})^3 T_m^{\text{loc}}(t), \quad (8)$$

where k denotes the effective switched capacitance cofficient that depends on the hardware architecture of the UD.

Edge Computing. Task $\mathbf{\Phi}_m(t)$ is offloaded to the UAV for processing (i.e., $a_m(t) = 1$). In this case, the UAV allocates computing and communication resources to perform the task. The edge processing delay includes transmission delay and edge execution delay, which can be calculated as

$$T_m^{\text{ec}}(t) = \frac{D_m(t)}{R_{m,u}(t)} + \frac{\eta_m(t)D_m(t)}{F_m(t)}. \quad (9)$$

The energy consumption generated by processing the task at time slot t consists of the transmission energy consumption of UD m and the computation energy consumption of the UAV. The transmission energy consumption of UD m at time slot t can be calculated as

$$E_m^{\text{ec}}(t) = P_m\frac{D_m(t)}{R_{m,u}(t)}. \quad (10)$$

Then, the computation energy consumption of the UAV to execute task $\mathbf{\Phi}_m(t)$ can be given as [22]

$$E_{m,u}^{\mathrm{c}}(t) = \varpi \eta_m(t) D_m(t). \tag{11}$$

where ϖ represents the UAV energy consumption per unit CPU cycle. Therefore, the total computation energy consumption of the UAV at time slot t can be given as

$$E_u^{\mathrm{c}}(t) = \sum_{m \in \mathcal{M}} a_m(t) E_{m,u}^{\mathrm{c}}(t). \tag{12}$$

D. Cost Model

UD Cost. In this work, we consider that each UD's cost at time slot t consists of the task completion delay and the UD's energy consumption, which reflects the UD's QoE. The completion delay of task $\mathbf{\Phi}_m(t)$ can be presented as

$$T_m(t) = (1 - a_m(t)) T_m^{\mathrm{loc}}(t) + a_m(t) T_m^{\mathrm{ec}}(t). \tag{13}$$

Then, the energy consumption of UD m can be given as

$$E_m(t) = (1 - a_m(t)) E_m^{\mathrm{loc}}(t) + a_m(t) E_m^{\mathrm{ec}}(t). \tag{14}$$

Similar to [30], [31], the cost of UD m at time slot t can be formulated as

$$C_m(t) = \gamma_m T_m(t) + (1 - \gamma_m) E_m(t), \tag{15}$$

where γ_m and $1 - \gamma_m$ represent the weighted parameters of delay and energy consumption of UD m respectively, which can be flexibly set based on the UD's preference for delay and energy consumption. Obviously, minimizing the cost of UDs is equivalent to maximizing the QoE of UDs.

UAV Energy Cost. Here, the cost of the UAV at time slot t is expressed as the energy consumption, which includes the computing energy consumption and propulsion energy consumption. Similar to [28], [32], the propulsion power consumption for a rotary-wing UAV with speed v_u can be expressed as

$$P_u(v_u) = \underbrace{C_1\left(1 + \frac{3v_u^2}{U_{\mathrm{p}}^2}\right)}_{\text{blade profile}} + \underbrace{C_2\sqrt{\sqrt{C_3 + \frac{v_u^4}{4}} - \frac{v_u^2}{2}}}_{\text{induced}} + \underbrace{C_4 v_u^3}_{\text{parasite}}, \tag{16}$$

where U_{p} refers to the rotor's tip speed, and $C1$, $C2$, $C3$, and $C4$ are constants described in [17]. Therefore, the energy consumption of the UAV at time slot t can be given as

$$E_u(t) = E_u^{\mathrm{c}}(t) + E_u^{\mathrm{p}}(t). \tag{17}$$

where $E_u^{\mathrm{p}}(t) = P_u(v_u(t))\tau$ denotes the propulsion energy consumption at time slot t. To guarantee service time, we define the UAV energy consumption constraint as follows:

$$\lim_{T \to +\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left\{E_u(t)\right\} \le \bar{E}_u, \tag{18}$$

where $\bar{E}_u$ is the energy budget of the UAV per time slot.

E. Problem Formulation

The objective of this work is to minimize the average costs of all UDs over time (i.e., time-average UD cost), by jointly optimizing the task offloading strategy $\mathbf{A} = \{\mathcal{A}^t | \mathcal{A}^t = \{a_m(t)\}_{m \in \mathcal{M}}\}_{t \in \mathcal{T}}$, computing resource allocation $\mathbf{F} = \{\mathcal{F}^t | \mathcal{F}^t = \{F_m(t)\}_{m \in \mathcal{M}}\}_{t \in \mathcal{T}}$, communication resource allocation $\mathbf{W} = \{\mathcal{W}^t | \mathcal{W}^t = \{w_m(t)\}_{m \in \mathcal{M}}\}_{t \in \mathcal{T}}$, and trajectory planning $\mathbf{P}_u = \{\mathbf{P}_u(t)\}_{t \in \mathcal{T}}$. Therefore, the problem can be formulated as follows:

$$\mathbf{P}: \min_{\mathbf{A},\mathbf{F},\mathbf{W},\mathbf{P}_u} \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} C_m(t) \tag{19}$$

$$\text{s.t.} \lim_{T \to +\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left\{E_u(t)\right\} \le \bar{E}_u, \tag{19a}$$

$$a_m(t) \in \{0,1\}, \forall m \in \mathcal{M}, t \in \mathcal{T}, \tag{19b}$$

$$a_m(t) T_m^{\mathrm{ec}}(t) \le T_m^{\max}, \forall m \in \mathcal{M}, t \in \mathcal{T}, \tag{19c}$$

$$0 \le F_m(t) \le F_u^{\max}, \forall m \in \mathcal{M}, t \in \mathcal{T}, \tag{19d}$$

$$\sum_{m=1}^{M} a_m(t) F_m \le F_u^{\max}, \forall t \in \mathcal{T}, \tag{19e}$$

$$0 \le w_m(t) \le 1, \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{19f}$$

$$\sum_{m=1}^{M} a_m(t) w_m(t) \le 1, \forall t \in \mathcal{T}, \tag{19g}$$

$$\mathbf{P}_u(1) = \mathbf{P}_I, \tag{19h}$$

$$\|\mathbf{p}_u(t+1) - \mathbf{p}_u(t)\| \le v_u^{\max}\tau, \forall t \in \mathcal{T}, \tag{19i}$$

Constraint (19a) is the long-term energy consumption constraint of the UAV. Constraint (19b) indicates that each UD can only select one strategy as its offloading decision. Constraint (19c) means that the completion delay of edge computing should not exceed the maximum tolerance delay. Constraints (19d) and (19e) imply that the allocated computing resources should be a positive value and not exceed the total amount of computing resources owned by the UAV. Constraints (19f) and (19g) limit the allocation of communication resources. Constraints (19h)-(19i) are the constraints on trajectory planning.

Challenges. There are two main challenges to obtain the optimal solution of problem $\mathbf{P}$. *i) Future-dependent.* Optimally solving problem $\mathbf{P}$ requires complete future information, e.g., task computing demands and locations of all UDs across all time slots. However, obtaining the future information is very challenging in the considered time-varying scenario. *ii) Non-convex and NP-hard.* Problem $\mathbf{P}$ contains both binary variables (i.e., task offloading decision $\mathbf{A}$) and continuous variables (i.e., resource allocation $\{\mathbf{F}, \mathbf{W}\}$ and UAV's trajectory $\mathbf{P}_u$) is an mixed-integer non-linear programming (MINLP) problem, which is non-convex and NP-hard [33], [34]. Therefore, solving the problem directly remains challenging even with knowledge of the future information.

IV. Lyapunov-Based Problem Transformation

Since problem $\mathbf{P}$ is future-dependent, an online approach is necessary to make real-time decisions without foreseeing the future. Lyapunov-based optimization framework is a commonly adopted method for designing online algorithms [17], [35], which has the advantage of being simple and effective. To this end, we first transform problem $\mathbf{P}$ into a per-slot real-time optimization problem based on the Lyapunov optimization framework.

Firstly, to satisfy the UAV energy constraint (19a), we define two virtual energy queues $Q_u^{\mathrm{c}}(t)$ and $Q_u^{\mathrm{p}}(t)$ to represent the computing energy queue and the propulsion energy queue at time slot t based on Lyapunov optimization technique, respectively. We assume that the queues are set as zero at the initial time slot, i.e., $Q_u^{\mathrm{c}}(1)=0$ and $Q_u^{\mathrm{p}}(1)=0$. Therefore, the virtual energy queues can be updated as

$$\begin{cases} Q_u^{\mathrm{c}}(t+1)=\max\left\{Q_u^{\mathrm{c}}(t)+E_u^{\mathrm{c}}(t)-\bar{E_u^{\mathrm{c}}},0\right\}, \forall t\in\mathcal{T}, \\ Q_u^{\mathrm{p}}(t+1)=\max\left\{Q_u^{\mathrm{p}}(t)+E_u^{\mathrm{p}}(t)-\bar{E_u^{\mathrm{p}}},0\right\}, \forall t\in\mathcal{T}, \end{cases} \tag{20}$$

where $\bar{E_u^{\mathrm{c}}}$ and $\bar{E_u^{\mathrm{p}}}$ represent the computation and propulsion energy budgets per slot, respectively and $\bar{E_u^{\mathrm{c}}}+\bar{E_u^{\mathrm{p}}}=\bar{E_u}$. Secondly, we define the *Lyapunov function* $L(\mathbf{Q}_u(t))$, which represents a scalar measure of the queue backlogs, i.e.,

$$L(\mathbf{Q}_u(t))=\frac{(Q_u^{\mathrm{c}}(t))^2+(Q_u^{\mathrm{p}}(t))^2}{2}. \tag{21}$$

where $\mathbf{Q}_u(t)=\{Q_u^{\mathrm{c}}(t),Q_u^{\mathrm{p}}(t)\}$ is the vector of current queue backlogs. Thirdly, we define the *conditional Lyapunov drift* for time slot t as:

$$\Delta L(\mathbf{Q}_u(t))\triangleq\mathbb{E}\{L(\mathbf{Q}_u(t+1))-L(\mathbf{Q}_u(t))\mid\mathbf{Q}_u(t)\}. \tag{22}$$

Finally, similar to [17], [22], [36], the *drift-plus-penalty* can be given as

$$D(\mathbf{Q}_u(t))=\Delta L(\mathbf{Q}_u(t))+V\mathbb{E}\left\{C_s(t)\mid\mathbf{Q}_u(t)\right\}, \tag{23}$$

where $C_s(t)=\sum_{m=1}^{M}C_m(t)$ is the total cost of all UDs at time slot t, and V is a parameter that trades off the total cost and queue stability.

Theorem 1. *For all t and all possible queue backlogs $\mathbf{Q}_u(t)$, the drift-plus-penalty is upper bounded as*

$$\begin{aligned} D(\mathbf{Q}_u(t))\leq & W+Q_u^{\mathrm{c}}(t)(E_u^{\mathrm{c}}(t)-\bar{E_u^{\mathrm{c}}}) \\ & +Q_u^{\mathrm{p}}(t)(E_u^{\mathrm{p}}(t)-\bar{E_u^{\mathrm{p}}})+V\times C_s(t), \end{aligned} \tag{24}$$

where $W=\frac{1}{2}\max\left\{\left(\bar{E_u^{\mathrm{c}}}\right)^2,\left(E_{\max}^{\mathrm{c}}-\bar{E_u^{\mathrm{c}}}\right)^2\right\}+\frac{1}{2}\max\left\{\left(\bar{E_u^{\mathrm{p}}}\right)^2,\left(E_{\max}^{\mathrm{p}}-\bar{E_u^{\mathrm{p}}}\right)^2\right\}$ *is a finite constant.*

Proof. The proof can refer to Theorem 1 in [17]. Due to the space limit, we omit the details. ■

According to the Lyapunov optimization framework, we minimize the right-hand side of inequality (24). Therefore, problem $\mathbf{P}$ that relies on future information is transformed into the real-time optimization problem $\mathbf{P}^{'}$ solvable with only current information, which is given as follows:

$$\mathbf{P}': \min_{\mathcal{A}^t,\mathcal{F}^t,\mathcal{W}^t,\mathbf{P}_{u'}} Q_u^{\mathrm{c}}(t)E_u^{\mathrm{c}}(t)+Q_u^{\mathrm{p}}(t)E_u^{\mathrm{p}}(t)+V\sum_{m=1}^{M}C_m(t) \tag{25}$$
$$\text{s.t. } (19\text{b})-(19\text{i})$$

where $\mathbf{P}_{u'}=\mathbf{P}_u(t+1)$ represents the UAV position at time slot $t+1$. However, problem $\mathbf{P}'$ is still an MINLP problem and the decision variables are coupled to each other. Therefore, a large amount of computational overhead caused by seeking the optimal solution for problem $\mathbf{P}'$ may not be suitable for real-time decision making. To this end, we design a two-stage optimization method that obtains a sub-optimal solution in polynomial time complexity. Furthermore, similar to [37], we drop the time index for variables for the convenience of the following description.

V. Two-Stage Optimization Algorithm

In the section, a two-stage optimization method is proposed to solve the transformed problem $\mathbf{P}'$. In the first stage, assuming a feasible $\mathbf{P}_{u'}$, we optimize the task offloading decision $\mathcal{A}$ and resource allocation $\{\mathcal{F},\mathcal{W}\}$. In the second stage, based on the obtained task offloading decision $\mathcal{A}^*$ and resource allocation $\{\mathcal{F}^*,\mathcal{W}^*\}$, we optimize the UAV position $\mathbf{P}_{u'}$.

A. Stage 1: Task Offloading and Resource Allocation

Assuming a feasible $\mathbf{P}_{u'}$ and removing irrelevant constant terms, $\mathbf{P}^{'}$ can be transformed into a subproblem **P1** to decide task offloading and resource allocation, which is given as

$$\mathbf{P1}: \quad V\cdot\min_{\mathcal{A},\mathcal{F},\mathcal{W}}\left(\frac{Q_u^{\mathrm{c}}}{V}E_u^{\mathrm{c}}+\sum_{m=1}^{M}C_m\right) \tag{26}$$
$$\text{s.t. } (19\text{b})-(19\text{g})$$

Problem **P1** is still an MINLP problem, and the decisions of task offloading and resource allocation are coupled with each other. Considering that the UAV is dominant in the considered UAV-enabled MEC system, we prioritize resource allocation strategies for the UAV. Then, based on the resource allocation strategy, we optimize the UDs' offloading decisions.

1) Resource Allocation: Given an arbitrary task offloading decision profile $\mathcal{A}$ of the UDs, the UAV decides resource allocation strategies to minimize problem **P1**. Define $s_m=\frac{F_m}{F_u^{\max}}$, the resource allocation problem can be formulated as

$$\begin{aligned} \mathbf{P1.1}: \min_{\mathcal{S},\mathcal{W}}\sum_{m\in\mathbf{M}_1} & \left[\gamma_m\left(\frac{D_m}{w_mBr_{m,u}}+\frac{\eta_mD_m}{s_mF_u^{\max}}\right)\right. \\ & \left.+(1-\gamma_m)\frac{P_mD_m}{w_mBr_{m,u}}\right] \end{aligned} \tag{27}$$

$$\text{s.t. } s_m\geq 0,\forall m\in\mathbf{M}_1, \tag{27a}$$
$$\sum_{m\in\mathbf{M}_1}s_m\leq 1, \tag{27b}$$
$$w_m\geq 0,\forall m\in\mathbf{M}_1, \tag{27c}$$
$$\sum_{m\in\mathbf{M}_1}w_m\leq 1, \tag{27d}$$

where $\mathcal{S}=\{s_m\}_{m\in\mathbf{M}_1}$, and $\mathbf{M}_1$ represents the set of UDs who offload tasks to the UAV, which is determined by the offloading decisions $\mathcal{A}$.

Lemma 1. *Problem **P1.1** is convex.*

Proof. Since the constraints are linear, Lemma 1 can be proved by showing that the Hessian matrix of the objective function (27) is positive semi-definite. ■

Theorem 2. *The optimal resource allocation coefficient, i.e., the solution of problem **P1.1**, can be given as follows:*

$$\begin{cases} s_m^*=\dfrac{\sqrt{\frac{\gamma_m\eta_mD_m}{F_u^{\max}}}}{\sum_{i\in\mathbf{M}_1}\sqrt{\frac{\gamma_i\eta_iD_i}{F_u^{\max}}}}, \\ w_m^*=\dfrac{\sqrt{\frac{\gamma_mD_m+(1-\gamma_m)P_mD_m}{Br_{m,u}}}}{\sum_{i\in\mathbf{M}_1}\sqrt{\frac{\gamma_iD_i+(1-\gamma_i)P_iD_i}{Br_{i,u}}}}. \end{cases} \tag{28}$$

Proof. Since problem **P1.1** is convex, the above conclusion can be obtained by KKT conditions [33]. ■

2) Task Offloading: For UD m, let us define U_m^{loc} as the utility of local computing and U_m^{ec} as the utility of edge computing, which can be given as follows:

$$U_m^{\mathrm{loc}} = \gamma_m T_m^{\mathrm{loc}} + (1-\gamma_m)E_m^{\mathrm{loc}}, \tag{29}$$

$$U_m^{\mathrm{ec}} = \frac{Q_u^{\mathrm{c}}}{V} E_{m,u}^{\mathrm{c}} + \gamma_m T_m^{\mathrm{ec}} + (1-\gamma_m)E_m^{\mathrm{ec}}. \tag{30}$$

Therefore, we can design the utility function of UD m as follows:

$$U_m(\mathcal{A}) = \begin{cases} U_m^{\mathrm{loc}}, a_m = 0, \\ U_m^{\mathrm{ec}}, a_m = 1. \end{cases} \tag{31}$$

According to the optimal resource allocation policy $\{\mathcal{F}^*, \mathcal{W}^*\}$ and removing irrelevant constant terms, problem **P1** can be transformed into a task offloading problem as follows:

$$\mathbf{P1.2}: \quad \min_{\mathcal{A}} \sum_{m \in \mathcal{M}} U_m(\mathcal{A}) \tag{32}$$
$$\text{s.t.} \quad \text{(19b) and (19c)}.$$

The offloading decision of UD m depends not only on its own demand but also on the offloading decisions of the other UDs. Considering the competitive nature of task offloading among UDs, game theory is employed to solve the task offloading decision problem.

(1) Game Formulation. We first model the task offloading decision problem as a multi-UDs task offloading game (MU-TOG). Specifically, the MU-TOG can be defined as a triplet $\Gamma = \{\mathcal{M}, \mathbb{A}, (U_m)_{m \in \mathcal{M}}\}$, which is detailed as follows:

- $\mathcal{M} = \{1, 2, \ldots, M\}$ denotes the set of players, i.e., all UDs.
- $\mathbb{A} = \mathbf{A}_1 \times \cdots \times \mathbf{A}_M$ denotes the strategy space, wherein $\mathbf{A}_m = \{0, 1\}$ is the set of offloading strategies for player m ($m \in \mathcal{M}$), $a_m \in \mathbf{A}_m$ denotes the offloading decision of player m, and $\mathcal{A} = (a_1, \ldots, a_M) \in \mathbb{A}$ is the strategy profile.
- $(U_m)_{m \in \mathcal{M}}$ is the utility function of player m that maps each strategy profile $\mathcal{A}$ to a real number.

Each player aims to minimize its utility by choosing a proper offloading strategy. Mathematically, the MU-TOG can be described by the following distributed optimization problem:

$$\min_{a_m} U_m(a_m, a_{-m}), \ \forall m \in \mathcal{M}, \tag{33}$$

where $a_{-m} = (a_1, \ldots, a_{m-1}, a_{m+1}, \ldots, a_M)$ denotes the offloading decisions of the other players except player m.

(2) The solution to MU-TOG. To determine the solution of MU-TOG, we first introduce the concept of Nash equilibrium, which describes a situation where no player has any incentive to unilaterally deviate from the current strategy.

Definition 1. *The strategy profile $\mathcal{A}^* = (a_1^*, \ldots, a_M^*)$ is a pure-strategy Nash equilibrium of game Γ if and only if*

$$U_m(a_m^*, a_{-m}^*) \le U_m(a_m', a_{-m}^*) \quad \forall a_m' \in \mathbf{A}_m, m \in \mathcal{M}. \tag{34}$$

Next, we introduce a powerful tool, known as exact potential game [38], to help us study the existence of Nash equilibrium and how to obtain a Nash equilibrium solution for the MU-TOG.

Definition 2. *A game is called an exact potential game if and only if a potential function $F(\mathcal{A}): \mathbb{A} \mapsto \mathbb{R}$ exists such that*

$$U_m(a_m, a_{-m}) - U_m(b_m, a_{-m}) \\ = F(a_m, a_{-m}) - F(b_m, a_{-m}), \forall (a_m, a_{-m}), (b_m, a_{-m}) \in \mathbb{A}. \tag{35}$$

Definition 3. *The exact potential game with finite strategy sets always has a Nash equilibrium and the finite improvement property (FIP) [38], [39].*

The FIP implies that a Nash equilibrium can be obtained in a finite number of iterations by any asynchronous better response update process.

Theorem 3. *The MU-TOG is an exact potential game where the potential function $F(\mathcal{A})$ can be given as*

$$F(\mathcal{A}) = \sum_{i \in \mathcal{M}} a_i \left(\frac{Q_u^{\mathrm{c}}}{V} E_{i,u}^{\mathrm{c}} + \beta_i \sum_{j \le i} a_j \beta_j + \phi_i \sum_{j \le i} a_j \phi_j \right) \\ + \sum_{i \in \mathcal{M}} (1 - a_i) U_i^{\mathrm{loc}}, \ \forall j \in \mathcal{M}, \tag{36}$$

where $\beta_i = \sqrt{\frac{\gamma_i \eta_i D_i}{F_u^{\max}}}$ and $\phi_i = \sqrt{\frac{\gamma_i D_i + (1-\gamma_i) P_i D_i}{B r_{i,u}}}$.

Proof. The proof can refer to Theorem 3 in [40]. ■

Then, let us consider the effect of constraint (19c) on the game. We can infer that imposing the constraint may render some strategy profiles infeasible. Suppose $\mathbb{A}'$ is the feasible strategy space, this leads to a new game $\Gamma' = \{\mathcal{M}, \mathbb{A}', (U_m)_{m \in \mathcal{M}}\}$.

Theorem 4. *Γ' is also an exact potential game and has the same potential function as Γ.*

Proof. The proof can refer to Theorem 2.23 in [39]. ■

The key idea of the MU-TOG is to utilize the FIP to update the offloading strategies of the players iteratively until the Nash equilibrium is reached, which is shown in Algorithm 1. The main steps of implementing the MU-TOG are described as follows. **i)** All UDs choose local computing for the initial setting (Line 1). **ii)** Each iteration is divided into N decision slots (Lines 4-15). At each decision slot, one UD is selected to update its offloading decision while the offloading decisions of the other UDs remain unchanged (Line 5). **iii)** If lower utility is achieved and constraint (19c) is satisfied, the UD's offloading decision is updated; otherwise, the original offloading decision is maintained (Lines 6-14). **iv)** When no UD changes its offloading decision, the MU-TOG reaches the Nash equilibrium.

B. Stage 2: UAV Trajectory Planning

Given the optimal task offloading decisions $\mathcal{A}^*$ and resource allocation $\{\mathcal{F}^*, \mathcal{W}^*\}$, while removing irrelevant constant terms, problem $\mathbf{P}'$ can be converted into the subproblem **P2** to decide the UAV trajectory planning, which is expressed as follows:

$$\mathbf{P2}: \min_{\mathbf{P}_{u'}} V \sum_{m \in \mathbf{M}_1} \frac{\gamma_m D_m + (1-\gamma_m) P_m D_m}{w_m^* B \log_2(1 + \frac{\phi_m}{(\|\mathbf{P}_{u'} - \mathbf{P}_m\|^2 + H^2)^{\mu}})} + \\ Q_u^{\mathrm{p}} \left(C_1 \left(1 + \frac{3v_u^2}{U_{\mathrm{p}}^2}\right) + C_2 \sqrt{\sqrt{C_3 + \frac{v_u^4}{4}} - \frac{v_u^2}{2}} + C_4 v_u^3 \right) \tau \tag{37}$$
$$\text{s.t. (19h)} - \text{(19i)}$$

Algorithm 1: The First Stage Algorithm

```
Input: The UD information $\{\mathbf{St}_m^{\mathrm{UD}}(t)\}_{m\in\mathcal{M}}$ and the
       current UAV location $\mathbf{P}_u$.
Output: The optimal task offloading and resource
        allocation decisions $\{\mathcal{A}^*, \mathcal{F}^*, \mathcal{W}^*\}$.
1  Initialization: The iteration number $l = 1$, $\mathcal{A}^0 = \emptyset$
   and $\mathcal{A}^1 = \{0, \dots, 0\}$;
2  repeat
3      $\mathcal{A}^{l-1} = \mathcal{A}^l$;
4      for UD $m \in \mathcal{M}$ do
5          $\mathbf{A}^l(m) = a_m^{\mathrm{ec}} = 1$;
6          Obtain $F_m^*$ and $w_m^*$ based on Eq. (28);
7          Calculate $T_m^{\mathrm{ec}}$ based on Eq. (9);
8          Calculate $U_m^{\mathrm{ec}}$ based on Eq. (30);
9          if $T_m^{\mathrm{ec}} \geq T_m^{\max}$ then
10             $\mathbf{A}^l(m) = a_m^{\mathrm{loc}} = 0$;
11         end
12         if $U_m^{\mathrm{ec}} \leq U_m^{\mathrm{loc}}$ then
13             $\mathbf{A}^l(m) = a_m^{\mathrm{loc}} = 0$;
14         end
15     end
16     Update $l = l + 1$;
17 until $\mathcal{A}^{l-1} = \mathcal{A}^l$;
18 $\mathcal{A}^* = \mathcal{A}^l$;
19 Obtain $\{\mathcal{F}^*, \mathcal{W}^*\}$ based on Eq. (28);
20 return $\{\mathcal{A}^*, \mathcal{F}^*, \mathcal{W}^*\}$.
```

where $v_u = \frac{\|\mathbf{P}_{u'} - \mathbf{P}_u\|}{\tau}$. Obviously, the objective function (37) is non-convex with respect to $\mathbf{P}_{u'}$ due to the non-convex terms $TM_0 = C_2\sqrt{\sqrt{C_3 + \frac{v_u^4}{4}} - \frac{v_u^2}{2}}$ and $\{TM_m = \frac{1}{\log_2\left(1+\frac{\phi_m}{(\|\mathbf{P}_{u'}-\mathbf{P}_m\|^2+H^2)^\mu}\right)}\}_{m\in\mathbf{M}_1}$. Therefore, it is difficult to directly solve problem $\mathbf{P}'$. We next transform the objective function into a convex function by introducing slack variables.

For the non-convex term TM_0, we introduce the slack variable y such that $y = TM_0$ and add the following constraint:

$$y \geq \sqrt{\sqrt{C_3 + \frac{v_u^4}{4}} - \frac{v_u^2}{2}} \Longrightarrow \frac{C_3}{y^2} \leq y^2 + v_u^2. \tag{38}$$

For the non-convex term TM_m, we introduce the slack variable z_m such that $z_m = TM_m$ and add the following constraint:

$$z_m \leq \log_2\left(1 + \frac{\phi_m}{\left(H^2 + \|\mathbf{P}_{u'} - \mathbf{P}_m\|^2\right)^\mu}\right). \tag{39}$$

According to the above-mentioned relaxation transformation, problem **P2** can be equivalently transformed as follows:

$$\begin{aligned}\mathbf{P2}' : \min_{\mathbf{P}_{u'}, y, z_m} \; & Q_u\left(P_0\left(1 + \frac{3v_u^2}{U_{\text{tip}}^2}\right) + C_2 y + C_3 v_u^3\right)\tau \\ & + V\sum_{m\in\mathbf{M}_1} \frac{\gamma_m D_m + (1-\gamma_m)P_m D_m}{w_m^* B z_m} \\ \text{s.t. } & (19\text{i}), (38) \text{ and } (39)\end{aligned} \tag{40}$$

Theorem 5. *Problem* $\mathbf{P2}'$ *is equivalent to problem* $\mathbf{P2}$.

Proof. Suppose $\{\mathbf{P}_{u'}^*, y^*, z_m^*\}$ is the optimal solution of problem $\mathbf{P2}'$. The following equation holds:

$$\begin{aligned} y^* &= \sqrt{\sqrt{C_3 + \frac{(v_u^*)^4}{4}} - \frac{(v_u^*)^2}{2}}, \\ z_m^* &= \log_2\left(1 + \frac{\phi_m}{\left(H^2 + \|\mathbf{P}_{u'}^* - \mathbf{P}_m\|^2\right)^\mu}\right), \end{aligned} \tag{41}$$

where $v_u^* = \frac{\|\mathbf{P}_{u'}^* - \mathbf{P}_u\|}{\tau}$. Otherwise, we can further reduce the objective function by choosing a smaller y or a larger z_m without violating the constraints (38) and (39). Therefore, $\mathbf{P}_{u'}^*$ is also the optimal solution to problem $\mathbf{P2}$. ■

For problem $\mathbf{P2}'$, the optimization objective (40) is convex but the additional constraints (38) and (39) are still non-convex. Similar to [15], [17], [41], the successive convex approximation (SCA) method is adopted to solve the non-convexity of (38) and (39).

Theorem 6. *Let* $f(\mathbf{P}_{u'}, y) = y^2 + v_u^2$ *and given a local point* $\mathbf{P}_{u'}^{(l)}$ *at the l-th iteration, we can obtain a global concave lower bound for* $f(\mathbf{P}_{u'}, y)$ *as*

$$\begin{aligned} f^{(l)}(\mathbf{P}_{u'}, y) \triangleq & \left(y^{(l)}\right)^2 + 2y^{(l)}\left(y - y^{(l)}\right) + \frac{\|\mathbf{p}_{u'}^{(l)} - \mathbf{p}_u\|^2}{\tau^2} \\ & + \frac{2}{\tau^2}(\mathbf{p}_{u'}^{(l)} - \mathbf{p}_u)^T\left(\mathbf{p}_{u'} - \mathbf{p}_u\right), \end{aligned} \tag{42}$$

where $y^{(l)}$ *is defined as*

$$y^{(l)} = \sqrt{\sqrt{C_3 + \frac{\|\mathbf{p}_{u'}^{(l)} - \mathbf{p}_u\|^4}{4\tau^4}} - \frac{\|\mathbf{p}_{u'}^{(l)} - \mathbf{p}_u\|^2}{2\tau^2}}. \tag{43}$$

Proof. Since $f(\mathbf{P}_{u'}, y)$ is a convex quadratic form, the first-order Taylor expansion of $f(\mathbf{P}_{u'}, y)$ at local point $\mathbf{P}_{u'}^{(l)}$ is a global concave lower bound. ■

Theorem 7. *Let* $g_m(\mathbf{P}_{u'}) = \log_2\left(1 + \frac{\phi_m}{\left(H^2 + \|\mathbf{P}_{u'} - \mathbf{P}_m\|^2\right)^\mu}\right)$, *we can obtain a global concave lower bound for* $g_m(\mathbf{P}_{u'})$ *as*

$$\begin{aligned} g_m^{(l)}(\mathbf{P}_{u'}) \triangleq & \log_2\left(1 + \frac{\phi_m}{\left(H^2 + \|\mathbf{p}_{u'}^{(l)} - \mathbf{p}_m\|^2\right)^\mu}\right) \\ & - \frac{\mu\phi_m(\log_2 e)(\|\mathbf{p}_{u'} - \mathbf{p}_m\|^2 - \|\mathbf{p}_{u'}^{(l)} - \mathbf{p}_m\|^2)}{[\phi_m + (H^2 + \|\mathbf{p}_{u'}^{(l)} - \mathbf{p}_m\|^2)^\mu](H^2 + \|\mathbf{p}_{u'}^{(l)} - \mathbf{p}_m\|^2)}. \end{aligned} \tag{44}$$

Proof. The proof can refer to Proposition 1 in [17]. ■

According to Theorems 6 and 7, at the l-th iteration, constraints (38) and (39) can be approximated as:

$$\frac{C_3}{y^2} \leq f^{(l)}(\mathbf{P}_{u'}, y), \tag{45}$$

$$z_m \leq g_m^{(l)}(\mathbf{P}_{u'}), \tag{46}$$

which are convex. Therefore, problem $\mathbf{P2}'$ is converted into a convex optimization problem, which can be efficiently resolved by off-the-shelf optimization tools such as CVX [42]. We summarize the second stage algorithm in Algorithm 2.

C. Main Steps of OJOA and Performance Analysis

In this section, the main steps of OJOA are described in Algorithm 3, and the corresponding analysis is given.

Algorithm 2: The Second Stage Algorithm

```
Input: The optimal task offloading and resource allocation decisions $\{\mathcal{A}^*, \mathcal{F}^*, \mathcal{W}^*\}$.
Output: The next location $\mathbf{P}_{u'}$.
1 Initialization: The accuracy threshold $\varepsilon = 0.01$, the local point $\mathbf{P}_{u'}^{(0)} = \mathbf{P}_u$, the iterative number $l = 1$ and the objective function value $G^{(0)} = 0$;
2 repeat
3     Calculate $y^{(l)}$ based on Eq. (43);
4     Obtain the optimal position $\mathbf{P}_{u'}^*$ and the objective value $G^{(l)}$ by solving problem $\mathbf{P2}'$;
5     Update the local point $\mathbf{P}_{u'}^{(l)} = \mathbf{P}_{u'}^*$;
6     Update $l = l + 1$;
7 until $|G^{(l)} - G^{(l-1)}| < \varepsilon$;
8 return $\mathbf{P}_{u'}^*$.
```

Theorem 8. *Assume that the proposed algorithm produces an optimality gap $C \geq 0$ in solving $\mathbf{P}'$ and C_s^{opt} denotes the optimal time-average UD cost that problem $\mathbf{P}$ can achieve over all policies given full knowledge of the future computing demands and locations for all UDs, the time-average UD cost achieved by the proposed algorithm is bounded by*

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M} C_m(t) \leq C_s^{\text{opt}} + \frac{WT + C}{V}, \tag{47}$$

where W is defined in Theorem 1.

Proof. According to Lemma 4.11 in [36], the *T-slot drift-plus-penalty* achieved by the proposed algorithm ensures that

$$L(\mathbf{Q}_u(T)) - L(\mathbf{Q}_u(1)) + V\sum_{t=1}^{T} C_s(t) \leq WT^2 + CT + VTC_s^{\text{opt}}. \tag{48}$$

Using the fact that $L(\mathbf{Q}_u(T)) \geq 0$ and $L(\mathbf{Q}_u(1)) = 0$, and dividing by VT for the above inequality, we can prove the theorem. ■

Theorem 9. *The proposed algorithm can satisfy the UAV energy consumption constraint defined in (18).*

Proof. The proof can refer to Theorem 2 in [22]. ■

Theorem 10. *The proposed OJOA has a polynomial worst-case complexity in each time slot, i.e., $\mathcal{O}(I_c M + M^{3.5}\log_2(\frac{1}{\varepsilon}))$, where I_c represents the number of iterations required for Algorithm 1 to converge to the Nash equilibrium, M denotes the number of UDs and ε is the accuracy of SCA for solving problem $\mathbf{P2}'$.*

Proof. OJOA contains two phases in each time slot, i.e., Algorithm 1 and Algorithm 2. In Algorithm 1, assuming that the outer iteration (i.e., Lines $2-17$) converges after I_c iterations, the computational complexity of the algorithm can be calculated as $\mathcal{O}(I_c M)$. In Algorithm 2, according to the analysis in [18], the computational complexity is $\mathcal{O}(M^{3.5}\log_2(\frac{1}{\varepsilon}))$. Therefore, the computational complexity of OJOA is $\mathcal{O}(I_c M + M^{3.5}\log_2(\frac{1}{\varepsilon}))$ in the worst case. ■

Accordingly, it is proven that the proposed algorithm can effectively guarantee the performance of the system, meet the UAV energy consumption constraint and have low computational complexity.

Algorithm 3: OJOA

```
Input: The energy queue $Q_u^{\text{c}}(1) = 0$, $Q_u^{\text{p}}(1) = 0$ and the control parameter $V$.
Output: time-average UD cost $TSC$.
1 Initialization: Initialize $TSC = 0$ and the initial position of the UAV $\mathbf{P}_u(1) = \mathbf{P}_I$;
2 for $t = 1$ to $t = T$ do
3     Acquire the UD information $\{\mathbf{St}_m^{\text{UD}}(t)\}_{m\in\mathcal{M}}$;
4     With fixed $\mathbf{P}_u(t)$, call Algorithm 1 to obtain $\{\mathcal{A}^*, \mathcal{F}^*, \mathcal{W}^*\}$;
5     With fixed $\{\mathcal{A}^*, \mathcal{F}^*, \mathcal{W}^*\}$, call Algorithm 2 to obtain $\mathbf{P}_{u'}^*$;
6     All UDs perform their tasks based on $\mathcal{A}^*$ and obtain corresponding cost $C_m^*(t)$;
7     The UAV provides MEC service to the UDs and flies towards position $\mathbf{P}_{u'}^*$;
8     System cost $C_s(t) = \sum_{m=1}^{M} C_m^*(t)$;
9     $TSC = TSC + C_s(t)$;
10    Update the energy queue $\mathbf{Q}_u(t+1)$ according to Eq. (20);
11    Update $t = t + 1$;
12 end
13 $TSC = TSC/T$;
14 return $TSC$.
```

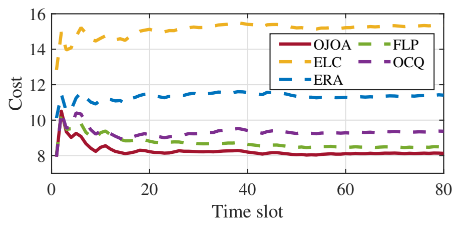
VI. Simulation Results

In this section, we perform simulations to validate the effectiveness of our proposed OJOA.

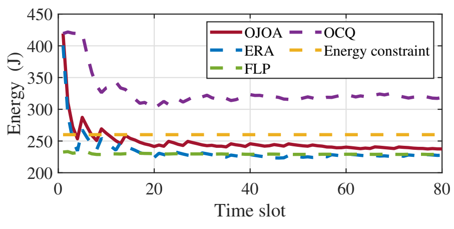
A. Simulation Setup

We consider a UAV-enabled MEC system consisting of a UAV and 20 UDs, where the initial horizontal position of the UAV is set as $\mathbf{P}_I = [200, 200]$, the fixed height is $H = 100$ m, and the initial positions of UDs are distributed in the area of 400×400 m^2. The system timeline is discretized into 80 time slots and the length of each time slot is 1 s [18]. The maximum speed of the UAV is set to $v_u^{\max} = 30$ m/s [17] and the total computing resources of the UAV are defined as $F_u^{\max} = 20$ GHz. The computing capacity of UDs is randomly taken from $\{1, 1.5, 2\}$ GHz, and the transmit power is set to $P_m = 0.1$ W. Each UD generates a computing task per time slot with input data size $D_m(t) \in [0.1, 1]$ Mb, computation intensity $\eta_m(t) \in [500, 1500]$ cycles/bit [43], and maximum tolerable delay $T_m^{\max} = 1$ s [18]. The channel bandwidth is set to $B = 4$ MHz. Moreover, we compare OJOA with the following four benchmark schemes:

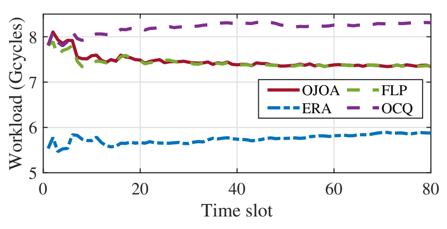
- Entire local computing (ELC): All UDs process their tasks locally.
- Equal resource allocation (ERA) [44]: The UAV allocates computing and communication resources equally.



(a) Time-average UD cost

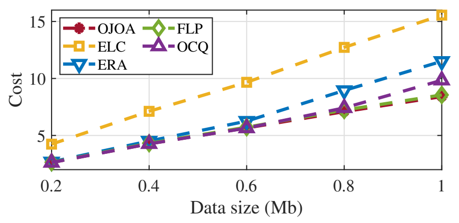


(b) Time-average UAV energy consumption

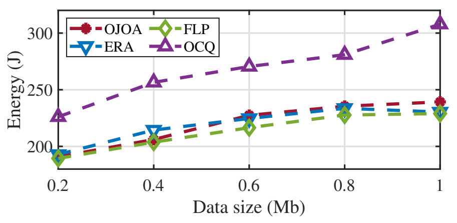


(c) Time-average UAV workload

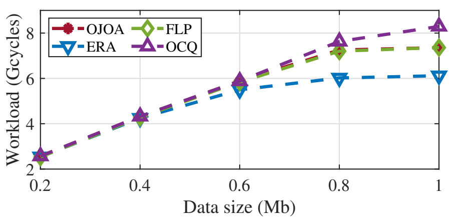
Fig. 2. System performance with respect to the time slots. (a) Time-average UD cost. (b) Time-average UAV energy consumption. (c) Time-average UAV workload.



(a) Time-average UD cost



(b) Time-average UAV energy consumption



(c) Time-average UAV workload

Fig. 3. System performance with respect to the task data size. (a) Time-average UD cost. (b) Time-average UAV energy consumption. (c) Time-average UAV workload.

- Fixed location deployment (FLP): The UAV hovers over the center of the service area to provide edge computing services.
- Only consider QoE (OCQ) [22]: Ignoring the UAV energy consumption constraint, all decisions are made only to minimize the time-average UD cost.

B. Evaluation Results

Impact of Time. Figs. 2(a), 2(b), and 2(c) show the dynamics of time-average UD cost, time-average UAV energy consumption, and time-average UAV workload among the five schemes. First, ELC exhibits the worst performance for time-average UD cost. Obviously, this is because all tasks are executed locally on UDs. Furthermore, ERA shows poorer performance in terms of time-average UD cost compared to FLP, OCQ, and OJOA. The reason is that due to UDs' heterogeneous computing requirements, the average resource allocation strategy cannot effectively utilize the limited computing and communication resources. It also explains that ERA has the lowest time-average UAV workload and time-average UAV energy consumption. In addition, it can be observed that OCQ achieves higher time-average UD cost compared to FLP and OJOA. This is mainly because of the game theory-based task offloading algorithm, which is detailed in Section V-A2. Specifically, regardless of the UAV energy consumption constraint, more UDs choose to offload tasks to the UAV, which leads to a heavier UAV workload. Finally, OJOA shows superior performance in the time-average UD cost among the five schemes and satisfies the UAV energy consumption constraint. This is because OJOA optimizes the trajectory of the UAV and adopts the optimal resource allocation strategy.

Impact of Data Size. Figs. 3(a), 3(b) and 3(c) show the impact of the task data size on time-average UD cost, time-average UAV energy consumption, and time-average UAV workload among the comparative schemes, respectively. First, it can be observed that the time-average UD cost, time-average energy consumption, and time-average UAV workload show an upward trend with the increasing task data size. This is expected as the larger task data size leads to higher overheads on computing, communication, and energy consumption for UDs and the UAV. Furthermore, we can see that ERA, OCQ, and OJOA achieve similar time-average UD cost when the task data size is relatively small (less than 0.4 Mb). The reason is the UAV has enough resources to process the tasks of UDs when the data size is small. Finally, it can be observed that the proposed OJOA is able to adapt to varying task data sizes with relatively superior performances in time-average UD cost, especially in the heavy workload scenario.

VII. Conclusion

In this work, we study task offloading, resource allocation, and UAV trajectory planning in an energy-constrained UAV-enabled MEC system. A JTRTOP is formulated to maximize the QoE of all UDs while satisfying the UAV energy consumption constraint. Since the JTRTOP is future-dependent and NP-hard, we propose the OJOA to solve the problem. Specifically, the future-dependent JTRTOP is firstly transformed into the PROP by using Lyapunov optimization methods. Furthermore, a two-stage optimization algorithm is proposed to solve the PROP. Simulation results show that OJOA outperforms the conventional approaches in terms of time-average UD cost while meeting the UAV energy consumption constraint.

Acknowledgement

This study is supported in part by the National Natural Science Foundation of China (62172186, 62002133, 61872158, 62272194), and in part by the Science and Technology Development Plan Project of Jilin Province (20230201087GX).

References

[1] C. Dong, Y. Shen, Y. Qu, K. Wang, J. Zheng, Q. Wu, and F. Wu, "UAVs as an intelligent service: Boosting edge intelligence for air-ground integrated networks," *IEEE Netw.*, vol. 35, no. 4, pp. 167–175, 2021.

[2] B. Hou, S. Yang, F. A. Kuipers, L. Jiao, and X. Fu, "EAVS: Edge-assisted adaptive video streaming with fine-grained serverless pipelines," in *Proc. of IEEE INFOCOM*, 2023.

[3] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *Proc. of ACM MobiCom*, 2019, pp. 1–16.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016.

[5] A. Hekmati, P. Teymoori, T. D. Todd, D. Zhao, and G. Karakostas, "Optimal mobile computation offloading with hard deadline constraints," *IEEE Trans. Mob. Comput.*, vol. 19, no. 9, pp. 2160–2173, 2020.

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

[7] Y. Qu, H. Dai, L. Wang, W. Wang, F. Wu, H. Tan, S. Tang, and C. Dong, "CoTask: Correlation-aware task offloading in edge computing," *World Wide Web*, vol. 25, no. 5, pp. 2185–2213, 2022.

[8] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[9] J. Li, G. Sun, L. Duan, and Q. Wu, "Multi-objective optimization for UAV swarm-assisted IoT with virtual antenna arrays," *IEEE Trans. Mob. Comput.*, 2023.

[10] J. Li, G. Sun, H. Kang, A. Wang, S. Liang, Y. Liu, and Y. Zhang, "Multi-objective optimization approaches for physical layer secure communications based on collaborative beamforming in UAV networks," *IEEE/ACM Trans. Networking*, 2023.

[11] Y. Qu, H. Sun, C. Dong, J. Kang, H. Dai, Q. Wu, and S. Guo, "Elastic collaborative edge intelligence for UAV swarm: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, 2023.

[12] P. A. Apostolopoulos, G. Fragkos, E. Tsiropoulou, and S. Papavassiliou, "Data offloading in UAV-assisted multi-access edge computing systems under resource uncertainty," *IEEE Trans. Mob. Comput.*, vol. 22, no. 1, pp. 175–190, 2023.

[13] P. Vamvakas, E. Tsiropoulou, and S. Papavassiliou, "On the prospect of UAV-assisted communications paradigm in public safety networks," in *Proc. of IEEE INFOCOM*, 2019, pp. 762–767.

[14] Y. Xu, T. Zhang, Y. Liu, D. Yang, L. Xiao, and M. Tao, "UAV-assisted MEC networks with aerial and ground cooperation," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 12, pp. 7712–7727, 2021.

[15] X. Zhang, J. Zhang, J. Xiong, L. Zhou, and J. Wei, "Energy-efficient multi-UAV-enabled multiaccess edge computing incorporating NOMA," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5613–5627, 2020.

[16] 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, 2019.

[17] Z. Yang, S. Bi, and Y. A. Zhang, "Online trajectory and resource optimization for stochastic UAV-enabled MEC systems," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 7, pp. 5629–5643, 2022.

[18] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, "Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 21, no. 10, pp. 3536–3550, 2022.

[19] L. T. Hoang, C. T. Nguyen, and A. T. Pham, "Deep reinforcement learning-based online resource management for UAV-assisted edge computing with dual connectivity," *IEEE/ACM Trans. Networking*, 2023.

[20] R. Zhou, X. Wu, H. Tan, and R. Zhang, "Two time-scale joint service caching and task offloading for UAV-assisted mobile edge computing," in *Proc. of IEEE INFOCOM*, 2022, pp. 1189–1198.

[21] Y. Qu, H. Dai, H. Wang, C. Dong, F. Wu, S. Guo, and Q. Wu, "Service provisioning for UAV-enabled mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3287–3305, 2021.

[22] H. Jiang, X. Dai, Z. Xiao, and A. Iyengar, "Joint task offloading and resource allocation for energy-constrained mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 22, no. 7, pp. 4000–4015, 2023.

[23] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for PCS networks," in *Proc. of IEEE INFOCOM*, 1999, pp. 1377–1384.

[24] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, 2020.

[25] S. Batabyal and P. Bhaumik, "Mobility models, traces and impact of mobility on opportunistic routing algorithms: A survey," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 3, pp. 1679–1707, 2015.

[26] L. Zhang, Z. Zhang, L. Min, C. Tang, H. Zhang, Y. Wang, and P. Cai, "Task offloading and trajectory control for UAV-assisted mobile edge computing using deep reinforcement learning," *IEEE Access*, vol. 9, pp. 53 708–53 719, 2021.

[27] G. Sun, X. Zheng, Z. Sun, Q. Wu, J. Li, Y. Liu, and V. C. Leung, "UAV-enabled secure communications via collaborative beamforming with imperfect eavesdropper information," *IEEE Trans. Mob. Comput.*, 2023.

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

[29] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, W. Saad, D. Niyato, and C. S. Hong, "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Trans. Mob. Comput.*, vol. 19, no. 6, pp. 1359–1374, 2020.

[30] Y. Chen, J. Zhao, Y. Wu, J. Huang, and X. S. Shen, "QoE-aware decentralized task offloading and resource allocation for end-edge-cloud systems: A game-theoretical approach," *IEEE Trans. Mob. Comput.*, 2022.

[31] Y. Ding, K. Li, C. Liu, and K. Li, "A potential game theoretic approach to computation offloading strategy optimization in end-edge-cloud computing," *IEEE Trans. Parallel Distributed Syst.*, vol. 33, no. 6, pp. 1503–1519, 2022.

[32] H. Pan, Y. Liu, G. Sun, J. Fan, S. Liang, and C. Yuen, "Joint power and 3D trajectory optimization for UAV-enabled wireless powered communication networks with obstacles," *IEEE Trans. Commun.*, vol. 71, no. 4, pp. 2364–2380, 2023.

[33] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[34] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numerica*, vol. 22, pp. 1–131, 2013.

[35] C. Ding, J. Wang, M. Cheng, M. Lin, and J. Cheng, "Dynamic transmission and computation resource optimization for dense LEO satellite assisted mobile-edge computing," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 3087–3102, 2023.

[36] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, ser. Synthesis Lect. Commun. Netw. Morgan & Claypool Publishers, 2010.

[37] G. Cui, Q. He, X. Xia, F. Chen, F. Dong, H. Jin, and Y. Yang, "OL-EUA: Online user allocation for NOMA-based mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 22, no. 4, pp. 2295–2306, 2023.

[38] D. Monderer and L. S. Shapley, "Potential Games," *Games Econ. Behav.*, vol. 14, no. 1, pp. 124–143, 1996.

[39] D. L. Quang, Y. H. Chew, and B. H. Soong, "Potential games," *Springer International Publishing*, 2016.

[40] S. Josilo and G. Dán, "Wireless and computing resource allocation for selfish computation offloading in edge computing," in *Proc. of IEEE INFOCOM*, 2019, pp. 2467–2475.

[41] J. Ji, K. Zhu, C. Yi, and D. Niyato, "Energy consumption minimization in UAV-assisted mobile-edge computing systems: Joint resource allocation and trajectory design," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8570–8584, 2021.

[42] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[43] Z. Sun, G. Sun, Y. Liu, J. Wang, and D. Cao, "BARGAIN-MATCH: A game theoretical approach for resource allocation and task offloading in vehicular edge computing networks," *IEEE Trans. Mob. Comput.*, pp. 1–18, 2023.

[44] S. Josilo and G. Dán, "Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks," *IEEE Trans. Mob. Comput.*, vol. 18, no. 1, pp. 207–220, 2019.