# Multi-objective Optimization for Multi-UAV-assisted Mobile Edge Computing

Geng Sun, *Member, IEEE*, Yixian Wang, Zemin Sun, Qingqing Wu, *Senior Member, IEEE*, Jiawen Kang, *Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, and Victor C. M. Leung, *Life Fellow, IEEE* 

Abstract—Recent developments in unmanned aerial vehicles (UAVs) and mobile edge computing (MEC) have provided users with flexible and resilient computing services. However, meeting the computing-intensive and latency-sensitive demands of users poses a significant challenge due to the limited resources of UAVs. To address this challenge, we present a multi-objective optimization approach for multi-UAV-assisted MEC systems. First, we formulate a multi-objective optimization problem aiming at minimizing the total task completion delay, reducing the total UAV energy consumption, and maximizing the total amount of offloaded tasks by jointly optimizing task offloading, computation resource allocation, and UAV trajectory control. Since the problem is a mixed-integer non-linear programming (MINLP) and NP-hard problem which is challenging, we propose a joint task offloading, computation resource allocation, and UAV trajectory control (JTORATC) approach to solve the problem. However, since the decision variables of task offloading, computation resource allocation, and UAV trajectory control, which are solved individually to obtain the corresponding decisions. Moreover, the sub-problem of task offloading is solved by using distributed splitting and threshold rounding methods, the sub-problem of computation resource allocation is solved by adopting the Karush-Kuhn-Tucker (KKT) method, and the sub-problem of UAV trajectory control is solved by employing the successive convex approximation (SCA) method. Simulation results show that the proposed JTORATC has superior performance compared to the other benchmark methods.

Index Terms—Unmanned aerial vehicle (UAV), mobile edge computing, task offloading, computation resource allocation, UAV trajectory control

# 1 INTRODUCTION

W ITH the rapid development of 6G and the Internet of Things (IoT), the number of smart mobile devices has shown unprecedented growth, leading to the increas-

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

- Geng Sun is with the College of Computer Science and Technology, Jilin University, Changchun 130012, China, and also with the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China. He is also with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798. E-mail: sungeng@jlu.edu.cn.
- Yixian Wang and Zemin Sun are with the College of Computer Science and Technology, Jilin University, Changchun 130012, China, and also with the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China.

E-mail: yixian23@mails.jlu.edu.cn, sunzemin@jlu.edu.cn.

- Qingqing Wu is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. E-mail: qingqingwu@sjtu.edu.cn.
- Jiawen Kang is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China. E-mail:kjwx886@163.com.
- Dusit Niyato is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798. E-mail: dniyato@ntu.edu.sg.
- Victor C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada. E-mail: vleung@ieee.org.

A small part of this paper is accepted by IEEE ICC 2024.

ing proliferation of various innovative mobile applications [1]. Most of these applications, such as face recognition, automatic navigation, and image processing, require intensive computation resources and low latency [2]. However, handling the computation-hungry and real-time data generated by these applications is challenging due to the limited resources of mobile devices [3]. In this context, mobile edge computing (MEC) has been regarded as a promising solution that allows mobile devices to offload the computation-intensive and delay-sensitive tasks to proximate edge servers [4], thereby reducing the computation burden, execution delay, and energy consumption of devices [5], [6], [7]. However, conventional MEC network relies on terrestrial infrastructures which are inflexible to deploy due to the installation cost and environmental limitations.

To overcome the physical restrictions of traditional terrestrial MEC systems, unmanned aerial vehicles (UAVs)assisted MEC is emerging to offer flexible and low-cost offloading services due to the high maneuverability, flexibility, fast deployment, and line-of-sight (LoS) links of UAVs [8], [9], [10], [11]. By offloading the computation tasks to the proximate UAVs, mobile users can flexibly enjoy cloud computing services anywhere and anytime [12], [13], [14].

However, designing an efficient task offloading approach in the multi-UAV-assisted MEC systems still faces several challenges. The design of efficient strategy could face the following challenges. *First*, due to the heterogeneous and strict requirements of users, it is complex to determine efficient task offloading decisions that can satisfy the diverse requirements of different users under the resource constraints of UAVs. *Second*, unlike the powerful cloud

computing, the UAV-assisted MEC servers are equipped with restricted computing capabilities due to the weight limitation of the UAVs. Therefore, it is challenging to efficiently allocate the limited computing resources of UAVs to guarantee the delay sensitivity and computation intensity of the tasks. Finally, UAVs are intrinsically limited by the finite onboard energy, which further limits the endurance of flight and service. However, performing the computationintensive tasks could be time-consuming. Consequently, energy-efficient trajectory control is crucial but challenging for UAVs to provide satisfactory offloading service within the limited flight time. From the perspective of problem formulation, most current research tends to concentrate on optimizing a single performance metric of the system, such as time delay or energy consumption while overlooking the multi-objective optimization. In terms of algorithm design, the methods of machine learning such as reinforcement learning generally face the challenge of difficult convergence and long training periods, especially in large-scale scenarios.

To address the abovementioned challenges, this work proposes a multi-objective optimization problem of task offloading, computation resource allocation, and UAV trajectory control for multi-UAV assisted MEC systems. The main contributions are summarized as follows:

- We consider a multi-UAV-assisted MEC system. Based on this system, we formulate a multi-objective optimization problem aimed at minimizing the total task completion delay, reducing the total UAV energy consumption, and maximizing the total amount of offloaded tasks by jointly optimizing task offloading, computation resource allocation, and UAV trajectory control. Moreover, the optimization problem is proven to be a non-convex and mixed-integer non-linear programming (MINLP) problem.
- We propose a joint task offloading, computation resource allocation and UAV trajectory control (JTORATC) approach to solve the problem. However, the decision variables of task offloading, computation resource allocation, and UAV trajectory control are coupled with each other. Thus, we first split the problem into three sub-problems, i.e., task offloading, computation resource allocation, and UAV trajectory control. Then, the sub-problem of task offloading is solved by using distributed splitting and threshold rounding methods, and the sub-problem of computation resource allocation is solved by adopting the Karush-Kuhn-Tucker (KKT) method. Besides, the sub-problem of UAV trajectory control is solved by employing the successive convex approximation (SCA) method. Ultimately, the proposed JTORATC approach effectively optimizes the performance of the entire system, reduces the complexity of the problem, and improves the solution efficiency.
- Simulations are conducted and the results show that the proposed JTORATC achieves superior performance in terms of objective function value, total task completion delay, and total UAV energy consumption compared to several benchmark schemes. Moreover, we find that the proposed algorithm has better scalability in the considered scenarios.

The rest of this study is organized as follows. Section 2 reviews the related work. Section 3 introduces the proposed

system model and problem formulation. Section 4 presents the JTORATC approach. Moreover, the simulation results are given in section 5. Finally, section 6 summarizes the paper.

# 2 RELATED WORK

In this section, we review the research work that are related to MEC, UAV-assisted MEC, task offloading, resource allocation, and UAV trajectory control, as well as optimization approaches. Moreover, we summarize the differences between the related works and this work in Table 1.

In recent years, MEC has become an area of intense interest. By offloading the latency-sensitive, compute-intensive and energy-intensive computing tasks to MEC servers, the quality of service (QoS) of mobile users can be significantly improved. For example, Zhang et al. [15] studied energyefficient computation offloading (EECO) mechanisms for MEC in 5G heterogeneous networks. The optimization problem is formulated to minimize the energy consumption of the offloading system while considering the energy cost of both task computing and file transmission. Furthermore, Mao et al. [16] considered MEC systems with energy harvesting devices and proposed an online task offloading algorithm which jointly determines the offload decision, CPU cycle frequency, and transmitted power. Besides, Kuang et al. [17] studied the joint optimization problem of partial offload scheduling and resource allocation in MEC systems with multiple independent tasks, with the aim of minimizing the weighted sum of execution delay and energy consumption while ensuring the transmission power constraints of the tasks. However, the abovementioned works studied task offloading and resource allocation in MEC with fixed edge infrastructures, which may not be applicable to the remote or disaster scenarios with inadequate infrastructures.

To be more flexible in meeting the mobile needs of UAVs in specific environments and to handle dynamic tasks efficiently, there has been a growing research interest in task offloading, resource allocation, and trajectory optimization for UAV-assisted MEC systems. For example, Li et al. [18] addressed the problem of maximizing UAV energy efficiency through the joint optimization of UAV trajectory, user transmit power, and computational load distribution. Moreover, the authors in [19] explored the problem of minimizing the total energy consumption by optimizing bit allocation, time slot scheduling, power allocation, and UAV trajectory. However, these studies mainly focus on single-server MEC systems, which may not be applicable to real-world scenarios, especially for large-scale scenarios with many users.

To overcome the challenges above, more research has focused on multi-UAV-assisted MEC environments. For example, Tun et al. [20] proposed a multi-UAVs assisted cooperative MEC system integrated with an MEC-enabled terrestrial base station (BS) to minimize the total delay experienced by mobile users, studying the joint offloading decision as well as the allocation of communication and computing resources under the energy constraints of mobile users and UAVs. Guo et al. [21] proposed a software defined network enhanced cooperative multiple UAV-enabled aerial computing (MUEAC) system to minimize the processing

TABLE 1 Comparison between Related Works and This Work

	Edge computing architecture		Optimization objective			Optimization variables			Method
Reference	UAV-assisted MEC	Multi-UAV- assisted MEC	Service delay	Energy consumption	Offloaded tasks	Task offloading	Resource allocation	UAV Trajectory Control	Distributed splitting, Threshold rounding, SCA, KKT
[15]	×	×	X	$\checkmark$	×	$\checkmark$	×	×	X
[16]	×	×	$\checkmark$	×	×	$\checkmark$	×	×	X
[17]	×	×	$\checkmark$	$\checkmark$	×	$\checkmark$	$\checkmark$	×	X
[18]	$\checkmark$	×	Х	$\checkmark$	×	$\checkmark$	×	×	X
[19]	$\checkmark$	X	Х	$\checkmark$	×	√	×	$\checkmark$	X
[20]	$\checkmark$	$\checkmark$	$\checkmark$	×	×	<ul> <li>✓</li> </ul>	$\checkmark$	×	X
[21]	$\checkmark$	$\checkmark$	$\checkmark$	×	×	<ul> <li>✓</li> </ul>	$\checkmark$	×	X
[10]	$\checkmark$	<ul> <li>✓</li> </ul>	Х	$\checkmark$	×	$\checkmark$	$\checkmark$	×	X
[22]	$\checkmark$	×	$\checkmark$	$\checkmark$	×	<ul> <li>✓</li> </ul>	<ul> <li>✓</li> </ul>	×	X
[23]	$\checkmark$	×	X	$\checkmark$	×	<ul> <li>✓</li> </ul>	$\checkmark$	×	X
[24]	$\checkmark$	$\checkmark$	Х	×	$\checkmark$	<ul> <li>✓</li> </ul>	×	$\checkmark$	X
[25]	$\checkmark$	$\checkmark$	Х	$\checkmark$	×	$\checkmark$	×	$\checkmark$	×
[26]	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	×	×	$\checkmark$	$\checkmark$	X
[27]	$\checkmark$	$\checkmark$	Х	X	×	$\checkmark$	×	×	X
[28]	$\checkmark$	$\checkmark$	Х	X	×	<ul> <li>✓</li> </ul>	$\checkmark$	$\checkmark$	X
This work	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	<ul> <li>✓</li> </ul>	$\checkmark$	$\checkmark$	$\checkmark$

delay of separable tasks, and studied the joint task scheduling and computing resource allocation problem under task data dependence and UAV energy consumption constraints. Moreover, Ei et al. [10] studied a multi-UAVs assisted twostage MEC system, where UAVs provide computing and relaying services to mobile devices to minimize the energy consumption of mobile devices (MDs) and UAVs by jointly optimizing task offloading, communication as well as computing resource allocation. However, all the studies above only optimize task offloading and resource allocation, and the UAVs are either stationary or follow a certain trajectory, which may not be suitable for dynamic or cooperative UAV flight scenarios.

To make full use of the flexible mobility of UAVs, some researches are devoted to integrating UAV trajectory control into UAVs-assisted MEC systems for optimizing the path planning of UAVs in dynamic environments and achieving more efficient task execution and resource allocation. For example, Yu et al. [22] focused on minimizing the weighted sum of service latency and UAV energy consumption of all IoT devices by jointly optimizing UAV location, communication and computing resource allocation, and task splitting decisions. Moreover, Chen et al. [23] aimed to minimize UAV energy consumption by jointly optimizing user offloading decisions, UAV positioning, and computing resource allocation. However, these studies mainly concentrated on UAV deployment while overlooking the trajectory optimization of UAVs, which could lead to inaccurate decision-making due to the lack of real-time control, especially in dynamic and complex environments. To address these issues, researchers have turned their focus to trajectory optimization in multi-UAV-assisted MEC systems. For example, Ji et al. [24] proposed a new scheme that maximizes minimum throughput among users of UAV services by jointly optimizing cache placement, UAV trajectory, and transmission power within a finite time period. Furthermore, Lee et al. [25] investigated a multi-UAV-mobile edge computing (UAV-MEC) network. By controlling the UAV-MEC trajectory and offloading ratio, the energy consumption, queue stability, and energy consumption of mobile devices (MDs) are jointly optimized. Besides, Park et al. [26] considered the deployment and optimization of MEC-UAVs for subterahertz communication in remote areas so that minimizing the energy consumption and latency of MEC-UAVs and mobile users.

In addition, researchers have adopted various methods to solve the maximum or minimum optimization problem in UAV-assisted MEC systems. For example, Wu et al. [27] studied joint computational offloading, UAV role and location selection problems in hierarchical multi-coalition UAVassisted MEC networks, and formulated discrete Stackelberg games with multiple leaders and followers to capture hierarchical features and discrete optimization. Moreover, Zhang et al. [28] proposed the problem of maximizing the computational efficiency in the multi-UAV assisted MEC system, and the joint optimization of user association, CPU cycle frequency, power and spectrum resource allocation and UAV trajectory scheduling was carried out and the iterative optimization algorithm of double-cycle structure was used to find the optimal solution. However, the abovementioned methods may have high computational complexity, especially when involving large-scale systems or complex dynamic environments.

The limitations of these previous studies are summarized as follows. First, these works primarily concentrated on optimizing one single performance of the system, such as delay and energy. Second, most of these studies only consider partial decision variables, e.g., task offloading and resource allocation or UAV trajectory and task offloading. Finally, most of the algorithms in these studies have high complexity. To address the limitations of these studies, we formulate a joint task offloading, computing resource allocation, and UAV trajectory control optimization problem to jointly minimize the total task completion delay and the total UAV energy consumption, and maximize the total amount of offloaded tasks. Moreover, we adopt the methods of distributed splitting, threshold rounding, KKT, and SCA to reduce the computational complexity.

TABLE 2 Summary of Notations

Symbol	Description	Symbol	Description
U	The number of users	M	The number of UAVs
U	The set of users	$\mathcal{M}$	The set of UAVs
Н	UAV flight altitude	В	Channel bandwidth
$d_{u,m}[n]$	The distance between UAV $m$ and user $u$ in	$g_{u,m}[n]$	Channel power gain of UAV $m$ and user
	time slot n		u in time slot $n$
$R_{u,m}[n]$	The data transmission rate between user <i>u</i> and	8	The rotor stiffness
	UAV $m$ in time slot $n$		
$p_u$	Transmit power of user <i>u</i>	$\sigma^2$	Noise power
$D_u[n]$	The size of the task	$C_u$	The computation intensity of the task
$f_u$	The computing capability of user <i>u</i>	$f_{m,u}[n]$	The computing capability allocated by
			UAV $m$ to user $u$
$T_{\rm local}[n]$	The service delay of user <i>u</i> to offload its task	$T_{\rm off}[n]$	The service delay of user <i>u</i> to process
	to UAV $m$ in time slot $n$		task locally in time slot $n$
$T_{m,u}^{\mathrm{comp}}[n]$	Task processing delay	$T_{u,m}^{\text{trans}}[n]$	Task transmission delay
$P_0$	The blade profile power of UAV in hover state	Pind	The induced power of UAV in hover
			state
$U_{tip}$	The blade tip speed of the rotor blade	$v_0$	The average rotor induced speed
	The fuselage resistance ratio	A	The area swept by UAV blades
$E_m^{\mathrm{fly}}[n]$	UAV $m$ flight energy consumption	$E_{m,u}^{\text{comp}}[n]$	UAV $m$ calculate energy consumption
V <sub>min</sub>	The minimum speed of UAV $m$	Vmax	The maximum speed of UAV $m$
$\kappa_m$	The effective switching capacitance of UAV $m$	$\kappa_u$	Effective switching capacitance of user <i>u</i>

# **3** System Model and Problem Formulation

In this section, we first present the system model, followed by the mobility model, communication model and computation model. Then, we formulate the multi-objective optimization problem. In addition, the notations associated with the system model are listed in Table 2 for the sake of readability.

#### 3.1 System Model



Fig. 1. Multi-UAV-assisted MEC for task offloading.

As shown in Fig. 1, a multi-UAV assisted MEC system is considered, which consists of M rotary-wing UAVs and U users, denoted as  $\mathcal{M} = \{1, 2, \ldots, M\}$  and  $\mathcal{U} = \{1, 2, \ldots, U\}$ . For ease of exposition, the continuous system timeline T is discretized into N time slots with equal time duration  $\delta_t = T/N$ , which is consistent with the coherence block of the wireless channel. Furthermore, each UAV is equipped with one or multiple MEC servers to provide computing services to the users. Besides, in each time slot, each user could generate a task that can be executed locally or offloaded to a UAV. The task of user u generated in time slot n is characterized as  $< D_u[n], C_u, \tau_u >$ , wherein  $D_u[n]$ 

is the task size (in bits),  $C_u$  is computation intensity of the task (cycles/bit), and  $\tau_u$  denotes the deadline of the task.

# 3.2 Mobility Model

Without loss of generality, we consider a three-dimensional (3D) Cartesian coordinate system in which the horizontal position of each user  $u \in \mathcal{U}$  is denoted as  $w_u = [x_u, y_u]^{\mathrm{T}}$ . Moreover, we consider that each UAV flies at a constant altitude H with the instantaneous horizontal position  $\mathbf{q}_m = [x_m[n], y_m[n]]^{\mathrm{T}}$ , acceleration  $a_m[n]$ , and velocity  $v_m[n]$  [29]. Let  $V_{\max}$  be the maximum speed of the UAV, then the maximum distance that each UAV can travel between the two stone troughs is expressed as  $D_{\max} = V_{\max}\delta_t$ . In addition, it is critical to ensure a safe distance among UAVs to avoid collisions, which means that the distance between any two UAVs should be greater than the minimum safe distance in any time slot. Therefore, the mobility constraints of UAVs can be given as follows:

$$\mathbf{q}_{m}[n+1] = \mathbf{q}_{m}[n] + v_{m}[n]\delta_{t} + \frac{1}{2}a_{m}[n]\delta_{t}^{2}, \ \forall m, n, \quad (1a)$$

$$v_m[n+1] = v_m[n] + a_m[n]\delta_t, \ \forall m, n, \tag{1b}$$

$$||v_m[n]|| \le V_{\max}, ||v_m[n]|| \ge V_{\min}, \ \forall m, n,$$
 (1c)

$$\|\mathbf{q}_m[n] - \mathbf{q}_i[n]\| \ge D_{\min}, \ \forall m \neq i, n, \tag{1d}$$

$$\mathbf{q}_m[1] = \mathbf{q}_m[N], \ \forall m, \tag{1e}$$

$$\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\| \le D_{\max} = V_{\max}\delta_t, \ \forall m, n, \qquad (1f_{j})$$

where (1a) and (1b) denote the position update and velocity update of UAVs, respectively, (1c) represents the speed constraints, (1d) ensures the safe distance between any two UAVs to avoid collisions, (1e) indicates that the initial and final positions of each UAV are fixed at the same location, and (1f) constrains the maximum flight distance during each time slot.

#### 3.3 Communication Model

The channel power gain between a user and a UAV is calculated by incorporating the probabilistic LoS transmissions

into both small-scale and large-scale fading [30]. For the uplink communication, the channel power gain between user u and UAV m at time slot n can be given as:

$$g_{u,m}[n] = p_{u,m}^{\text{LoS}} g_{u,m}^{\text{LoS}}[n] + (1 - p_{u,m}^{\text{LoS}}) g_{u,m}^{\text{NLoS}}[n], \qquad (2)$$

where  $p^{\text{LoS}}$  represents the probability of LoS link,  $g_{u,m}^{\text{LoS}}[n]$  and  $g_{u,m}^{\text{NLoS}}[n]$  represent the channel power gain between user u and UAV m for LoS and NLoS links, respectively, which is calculated as:

$$g_{u,m}^{\text{LoS}}[n] = |h_{u,m}^{\text{LoS}}[n]|^2 \left( \mathcal{I}_{u,m}^{\text{LoS}}[n] \right)^{-1} 10^{\frac{-\mathcal{I}_{\sigma}^{\text{LoS}}}{10}},$$
(3a)

$$g_{u,m}^{\rm NLoS}[n] = |h_{u,m}^{\rm NLoS}[n]|^2 \left(\mathcal{I}_{u,m}^{\rm NLoS}[n]\right)^{-1} 10^{\frac{-\mathcal{J}_{\sigma}^{\rm NLoS}}{10}}, \quad (3b)$$

where  $h_{u,m}^{\text{LoS}}[n]$  and  $h_{u,m}^{\text{NLoS}}[n]$ ,  $\mathcal{I}_{u,m}^{\text{LoS}}[n]$  and  $\mathcal{I}_{u,m}^{\text{NLoS}}[n]$ ,  $\mathcal{J}_{\sigma}^{\text{LoS}}$  and  $\mathcal{J}_{\sigma}^{\text{NLoS}}$  are the components of small-scale fading, path loss, and shadowing for LoS and NLoS links, respectively. For LoS link, these components are detailed as follows.

First, the small-scale fading characteristic of the channel is captured by using a parametric-scalable and good-fitting generalized fading model, i.e., Nakagami-*m* fading [31], and to distinguish it from the identification of UAV, we use *w* to represent the shape parameters in the Nakagami model, i.e., Nakagami-*w*. Specifically,  $h_{u,m}^{\text{LoS}}[n]$  and  $h_{u,m}^{\text{NLoS}}[n]$  follow the Nakagami distribution with fading parameter  $w^{\text{LoS}}$  and  $w^{\text{NLoS}}$ , which can be given as:

$$h_{u,m}^{\mathrm{LoS}}[n] = \frac{2\left(w^{\mathrm{LoS}}\right)^{w^{\mathrm{LoS}}}h_{u,m}^{\mathrm{LoS}}[n]^{2w^{\mathrm{LoS}}-1}e^{\left(-\frac{w^{\mathrm{LoS}}h^{2}}{\overline{p}}\right)}}{\Gamma\left(w^{\mathrm{LoS}}\right)\overline{p}^{w^{\mathrm{LoS}}}}, \quad (4a)$$
$$h_{u,m}^{\mathrm{NLoS}}[n] = \frac{2\left(w^{\mathrm{NLoS}}\right)^{w^{\mathrm{NLoS}}}h_{u,m}^{\mathrm{NLoS}}[n]^{2w^{\mathrm{NLoS}}-1}e^{\left(-\frac{w^{\mathrm{NLoS}}h^{2}}{\overline{p}}\right)}}{\overline{p}}, \quad (5a)$$

$$\Gamma\left(w^{\mathrm{NLoS}}\right)\overline{p}^{w^{\mathrm{NLoS}}}$$
, (4b)

where  $\overline{p}$  is the average received power in the fading envelope, and  $\Gamma(w)$  is the Gamma function.

Second, the path loss between user u and UAV m for LoS/NLoS link can be given as:

$$\mathcal{I}_{u,m}^{\text{LoS}}[n] = \frac{\left(4\pi d_0 f_c\right)^2}{c^2} \left(\frac{d_{u,m}[n]}{d_0}\right)^{\beta^{\text{LoS}}},$$
 (5a)

$$\mathcal{I}_{u,m}^{\rm NLoS}[n] = \frac{(4\pi d_0 f_c)^2}{c^2} \left(\frac{d_{u,m}[n]}{d_0}\right)^{\beta^{\rm NLoS}},$$
 (5b)

where  $f_c$  is the carrier frequency, c is the speed of light,  $d_0$  is the reference distance,  $d_{u,m}[n]$  is the distance between user u and UAV m, and  $\beta^{\text{LoS}}$  and  $\beta^{\text{NLoS}}$  are the path loss exponents for LoS and NLoS links, respectively.

Finally, the shadowing captures the signal attenuation caused by shadowing in transmission, and it can be modeled as a zero-mean Gaussian distributed random variable, which is as follows:

$$\mathcal{J}_{\sigma}^{\mathrm{LoS}}[n] \sim \mathcal{O}\left(0, \left(\sigma^{\mathrm{LoS}}\right)^{2}\right), \tag{6a}$$

$$\mathcal{J}_{\sigma}^{\mathrm{NLoS}}[n] \sim \mathcal{O}\left(0, \left(\sigma^{\mathrm{NLoS}}\right)^{2}\right), \tag{6b}$$

where  $\sigma^{\text{LoS}}$  and  $\sigma^{\text{NLoS}}$  are the standard deviation of shadowing for LoS and NLoS links, respectively [30].

Accordingly, in time slot n, the data transmission rate between user u and UAV m can be given as:

$$R_{u,m}[n] = B \log_2 \left( 1 + \frac{p_u g_{u,m}[n]}{\sigma^2} \right),$$
(7)

where *B* is the channel bandwidth,  $p_u$  is the transmit power

of user u, and  $\sigma^2$  is the noise power.

#### 3.4 Computation Model

This section presents the service delay, energy consumption, and total amount of offloaded tasks.

#### 3.4.1 Service Delay

The service delay depends on the task offloading decision  $a_{u,m}[n]$ , which indicates whether the task of user u is offloaded to UAV m (i.e.,  $a_{u,m}[n] = 1$ ) or processed locally (i.e.,  $a_{u,m}[n] = 0$ ).

**Local Computing.** The service delay of user u to process task  $K_u[n]$  locally in time slot n can be given as:

$$T_{\text{local}}[n] = \frac{D_u[n]C_u}{f_u},\tag{8}$$

where  $f_u$  denotes the computing capability of user u.

**UAV-Assisted Computing.** The service delay of user u to offload task  $K_u[n]$  to UAV m in time slot n mainly consists of the transmission delay and processing delay. Specifically, the transmission delay can be given as:

$$T_{u,m}^{\text{trans}}[n] = \frac{D_u[n]}{R_{u,m}[n]}.$$
(9)

Moreover, the processing delay can be expressed as:

$$T_{m,u}^{\rm comp}[n] = \frac{D_u[n]C_u}{f_{m,u}[n]},$$
(10)

where  $f_{m,u}$  indicates the computing capability allocated by UAV m to the user u.

Therefore, by combining Eqs. (9) and (10), the service delay for UAV-assisted computing can be calculated as:

$$T_{\text{off}}[n] = \frac{D_u[n]C_u}{f_{m,u}[n]} + \frac{D_u[n]}{R_{u,m}[n]}.$$
 (11)

According to Eqs. (8) and (11), the total task completion delay for all users across N time slots can be calculated as:

$$T_{\text{total}} = \sum_{n=1}^{N} \sum_{u=1}^{U} \sum_{m=1}^{M} a_{u,m}[n] T_{\text{off}}[n] + (1 - a_{u,m}[n]) T_{\text{local}}[n]$$
$$= \sum_{n=1}^{N} \sum_{u=1}^{U} \sum_{m=1}^{M} a_{u,m}[n] \left(T_{\text{off}}[n] - T_{\text{local}}[n]\right) + T_{local}[n].$$
(12)

Given that the results are typically much smaller in comparison to the input data for most applications, the result download delay is ignored [32].

#### 3.4.2 Energy Consumption

Completing tasks could impose additional costs of energy consumption on both users and UAVs.

*Local Computing.* The energy consumption of user u to process task  $K_u[n]$  locally in time slot n can be given as:

$$E_{\text{local}}^{\text{comp}}[n] = \kappa_u (f_u)^2 D_u[n] C_u, \tag{13}$$

where  $\kappa_u$  denotes the effective capacitance coefficient of user *u*, which depends on the chip structure of the CPU [33].

**UAV-Assisted Edge Computing.** The energy consumed by UAV m to process task  $K_u[n]$  in time slot n can be given as [34]:

$$E_{m,u}^{\text{comp}}[n] = \kappa_m (f_{m,u}[n])^2 D_u[n] C_u,$$
(14)

where  $\kappa_m$  represents the effective switching capacitance that depends on the CPU architecture.

*UAV Fight.* The propulsion power consumption of UAV m flying at speed  $v_m[n]$  can be given as [35], [36]:

$$E_m^{\text{fly}}[n] = \left(P_0\left(1 + \frac{3\|v_m[n]\|^2}{U_{\text{tip}}^2}\right) + P_{\text{ind}}\left(\sqrt{1 + \frac{\|v_m[n]\|^4}{4v_0^4}} - \frac{\|v_m[n]^2\|}{2v_0^2}\right)^{1/2} + \frac{1}{2}d_0\rho_0|sA\|v_m[n]^3\|)\delta_t,$$
(15)

where  $P_0$  is the blade profile power,  $P_{ind}$  is the induced power,  $U_{tip}$  is the blade tip speed of the rotor blade, and  $v_0$  is the average rotor induced speed. Moreover,  $d_0$  indicates the fuselage resistance ratio,  $\rho_0$  is the air density, s is the rotor stiffness, and A indicates the area swept by UAV blades.

According to Eqs. (14) and (15), the total energy consumption of the UAVs across N time slots is mainly incurred by task execution and flight, which can be given as:

$$E_{\text{total}} = \sum_{n=1}^{N} \left( \sum_{u=1}^{U} \sum_{m=1}^{M} \left( E_m^{\text{fly}}[n] + a_{u,m}[n] E_{m,u}^{\text{comp}}[n] \right) \right).$$
(16)

# 3.4.3 Total Amount of Offloaded Tasks

The total amount of offloaded tasks can reflect the service quality provided by UAVs, which can be calculated as:

$$K_{\text{total}} = \sum_{n=1}^{N} \left( \sum_{u=1}^{U} \sum_{m=1}^{M} a_{u,m}[n] D_u[n] \right).$$
(17)

# 3.5 Problem Formulation

This work aims to simultaneously minimize the total task completion delay, minimize the total UAV energy consumption, and maximize the total amount of offloaded tasks by optimizing the decisions of task offloading  $\mathbf{A} = \{a_{u,m}[n], \forall u, m, n\}$ , computation resource allocation  $\mathbf{F} = \{f_{m,u}[n], \forall u \in \mathbf{U}_0, m, n\}$ , and UAV trajectory control  $\mathbf{Q} = \{\mathbf{q}_m[n], \forall m, n\}$ . Accordingly, the multi-objective optimization problem can be formulated as:

$$\mathbf{P}: \min_{\mathbf{A}, \mathbf{F}, \mathbf{Q}} \{T_{\text{total}}, E_{\text{total}}, -K_{\text{total}}\}$$
(18a)

s.t. 
$$a_{u,m}[n] \in \{0,1\}, \forall u, m, n,$$
 (18b)

$$\sum_{\substack{m=1\\ U}} a_{u,m}[n] \le 1, \ \forall u, m, n, \tag{18c}$$

$$\sum_{u=1}^{c} a_{u,m}[n] \le 1, \ \forall u, m, n,$$
(18d)

$$a_{u,m}[n] \cdot T_{\text{local}}[n] \le \tau_u, \ \forall u, m, n,$$
 (18e)

$$a_{u,m}[n] \cdot T_{\text{off}}[n] \le \tau_u, \ \forall u, m, n, \tag{18f}$$

$$0 \le f_{m,u}[n] \le f_m^{\max}, \ \forall m, n, \tag{18g}$$

$$\sum_{u=1}^{3} f_{m,u}[n] \le f_m^{\max}, \ \forall u, m, n,$$
(18h)

$$(1a) \sim (1f).$$
 (18i)

Constraints (18b)-(18d) represent the task offloading constraints of users. Constraints (18e) and (18f) indicate the deadline of the tasks. Constraints (18g) and (18h) limit the computation resource of UAV. Moreover, constraint (18i) represents the mobility constraint of UAV. **Theorem 1.** Problem **P** is a non-convex and NP-hard MINLP.

*Proof.* Problem **P** involves binary variables (i.e., task offloading **A**) and continuous variables (i.e., computation resource allocation **F** and UAV trajectory control **Q**), while the inequalities in (1c) and (1d) are non-convex constraints. Consequently, problem **P** is an MINLP problem, which is also non-convex and NP-hard.

It can be deduced from Theorem 1 that it is difficult to find an optimal solution to problem **P** due to the NP hardness, which motivates us to propose the JTORATC.

# 4 THE PROPOSED JTORATC

In this section, the JTORATC is proposed to solve the formulated optimization problem. First, we transform the original multi-objective optimization problem into a single-objective optimization problem to simplify complexity, unify evaluation criteria, and facilitate trade-offs of different objectives. Then, we split the single-objective optimization problem into three sub-problems of task offloading, computation resource allocation, and UAV trajectory control based on the block alternate descent method, which alternates the solution of each sub-problem iteratively. Specifically, we employ distributed splitting and threshold rounding methods to solve the sub-problem of task offloading. Moreover, the sub-problem of computation resource allocation is solved by adopting KKT method, and the sub-problem of UAV trajectory control is solved by employing SCA method.

# 4.1 Problem Reformulation

Similar to [37], we convert the multi-objective optimization problem into a single-objective optimization problem by using the right-of-use coefficient method. Consequently, the objective function can be rewritten as:

$$\rho(\mathbf{A}, \mathbf{F}, \mathbf{Q}) = w_1 T_{\text{total}} + w_2 E_{\text{total}} - w_3 K_{\text{total}}$$

$$= \sum_{n=1}^{N} \sum_{u=1}^{U} \sum_{m=1}^{M} w_1 \left( a_{u,m}[n] \left( T_{\text{off}}[n] - T_{\text{local}}[n] \right) + T_{local}[n] \right)$$

$$+ w_2 \left( E_m^{\text{fly}}[n] + a_{u,m}[n] E_{m,u}^{\text{comp}}[n] \right) - w_3 \left( a_{u,m}[n] D_u[n] \right),$$
(19)

where the non-negative parameters  $w_1$ ,  $w_2$ , and  $w_3$  denote the weight factors of the delay, energy consumption, and amount of offloaded tasks, respectively. Note that the weight coefficients can be adjusted according to the user preferences on the objectives. Then, the original multi-objective optimization problem **P** is transformed into a single-objective optimization problem **P**<sub>1</sub>, which is expressed as follows:

$$\mathbf{P_{1}}: \min_{\mathbf{A}, \mathbf{F}, \mathbf{Q}} \quad \rho(\mathbf{A}, \mathbf{F}, \mathbf{Q})$$
(20a)

s.t. 
$$a_{u,m}[n] \in \{0,1\}, \forall u, m, n,$$
 (20b)

$$\sum_{m=1} a_{u,m}[n] \le 1, \ \forall u, m, n,$$
 (20c)

$$\sum_{u=1}^{U} a_{u,m}[n] \le 1, \ \forall u, m, n,$$
(20d)

$$a_{u,m}[n] \cdot T_{\text{local}}[n] \le \tau_u, \ \forall u, m, n,$$
 (20e)

$$a_{u,m}[n] \cdot T_{\text{off}}[n] \le \tau_u, \ \forall u, m, n, \tag{20f}$$

$$0 \le f_{m,u}[n] \le f_m^{\max}, \ \forall m, n, \tag{20g}$$

$$\sum_{u=1}^{U_0} f_{m,u}[n] \le f_m^{\max}, \ \forall u, m, n,$$
(20h)

$$(1a) \sim (1f).$$
 (20i)

This transformation reduces the difficulty of the original formulated optimization problem, avoids the metric differences between different objectives, and permits flexible handling of objectives through weight adjustments.

However, the transformed optimization problem is still difficult to solve due to the convex objective function and non-convex constraints. Considering that the decision variables of task offloading, computation resource allocation, and UAV trajectory control are coupled with each other, we split the single-objective optimization problem into three sub-problems of task offloading, computation resource allocation, and UAV trajectory control based on the block alternate descent method.

**Remark 1.** If problem  $P_1$  is split into three sub-problems, each sub-problem can be transformed into a convex optimization problem and can be solved using the existing methods. However, splitting problem  $P_1$  into two sub-problems may compromise flexibility, solving efficiency, and complexity. For example, if the original problem is split into the sub-problems of  $P_{1,1}$ :  $\min_{\mathbf{A},\mathbf{F}} \rho(\mathbf{A},\mathbf{F})$  and  $\mathbf{P}_{1,2}$  :  $\min_{\mathbf{Q}} \rho(\mathbf{Q})$ , or into the subproblems of  $P_{1,1}$  :  $\min_{A,Q} \rho(A, Q)$  and  $P_{1,2}$  :  $\min_{F} \rho(Q)$ , combining binary variable A and continuous variable F or **Q** into a single sub-problem might limit the ability to apply different methods tailored to each variable type. Furthermore, if the problem is split into sub-problems of  $\mathbf{P}_{1,1}$ : min<sub>A</sub>  $\rho(\mathbf{A})$  and  $\mathbf{P_{1,2}}$ :  $\min_{\mathbf{F},\mathbf{Q}} \rho(\mathbf{F},\mathbf{Q})$ , the mixed continuous variables could further add the complexity on solving the problem. Thus, it is reasonable to split the original optimization problem into three sub-problems.

#### 4.2 Task Offloading

Given the computation resource allocation  $\mathbf{F} = \{ \mathbf{\hat{F}}_{m,u}[n], \forall u \in \mathbf{U}_0, m, n \}$  and UAV trajectory control  $\mathbf{\hat{Q}} = \{ \mathbf{\hat{q}}_m[n], \forall m, n \}$ , the task offloading optimization problem can be expressed as:

$$\mathbf{P_{1.1}}: \min_{\mathbf{A}} \quad \rho\left(\mathbf{A}, \widehat{\mathbf{F}}, \widehat{\mathbf{Q}}\right)$$
(21a)

s.t. 
$$a_{u,m}[n] \in \{0,1\}, \ \forall u,m,n,$$
 (21b)

$$\sum_{m=1} a_{u,m}[n] \le 1, \ \forall u, m, n, \tag{21c}$$

$$\sum_{u=1}^{U} a_{u,m}[n] \le 1, \ \forall u, m, n,$$
(21d)

$$a_{u,m}[n] \cdot T_{\text{local}}[n] \le \tau_u, \ \forall u, m, n,$$
 (21e)

$$a_{u,m}[n] \cdot T_{\text{off}}[n] \le \tau_u, \ \forall u, m, n.$$
 (21f)

Problem  $\mathbf{P}_{1,1}$  is still non-convex due to the constraint (21b), which involves the binary variable  $a_{u,m}[n]$ . Therefore, we first transform problem  $\mathbf{P}_{1,1}$  into a convex problem by using the distributed splitting method. Then, the threshold rounding method is adopted to convert the relaxed continuous variables back into binary variables.

For the non-convex constraint (21b), we first employ the distributed splitting method to transform the binary variables into continuous variables. In specific, the variable of task offloading  $\mathbf{A}$  is split into two parts, i.e.,  $\mathbf{X}$  and  $\mathbf{Y}$  as follows:

$$\mathbf{X} = \{x_a^u[n] \mid x_a^u[n] = 1 - a^u[n], \ \forall u, n\},$$
(22a)

$$\mathbf{Y} = \{y_a^m[n] \mid y_a^m[n] = a^m[n], \ \forall m, n\},$$
(22b)

where  $a^u[n] = 1$  indicates that the task is processed locally, and  $a^m[n] = 1$  means being offloaded to the UAV. Then, We slack the variables into continuous variables as follows:

$$\overline{\mathbf{X}} \triangleq \left\{ \sum_{u=1}^{U} x_a^u[n] = 1, x_a^u[n] \in [0,1] \right\},$$
(23a)

$$\overline{\mathbf{Y}} \triangleq \left\{ \sum_{m=1}^{M} y_a^m[n] = 1, y_a^m[n] \in [0, 1] \right\}.$$
 (23b)

Based on the above steps, problem  $P_{1,1}$  is transformed into problem  $P_{1,1'}$  as follows:

$$\mathbf{P}_{\mathbf{1.1'}}: \quad \min_{\overline{\mathbf{X}}, \overline{\mathbf{Y}}} \quad \rho\left(\overline{\mathbf{X}}, \overline{\mathbf{Y}}, \widehat{\mathbf{F}}, \widehat{\mathbf{Q}}\right)$$
(24a)

s.t. 
$$\sum_{\substack{m=1\\ U}}^{M} y_a^m[n] \le 1, \ \forall u, m, n,$$
 (24b)

$$\sum_{u=1}^{6} x_{a}^{u}[n] \le 1, \ \forall u, m, n,$$
(24c)

$$x_a^u[n] \cdot T_{\text{local}}[n] \le \tau_u, \ \forall u, m, n,$$
 (24d)

$$y_a^m[n] \cdot T_{\text{off}}[n] \le \tau_u, \ \forall u, m, n, \tag{24e}$$

(23a), (23b). (24f)

**Theorem 2.** Problem  $P_{1,1'}$  is a convex non-linear programming (NLP) problem.

*Proof.* Since the computation resource allocation and UAV trajectory control have been fixed,  $E_m^{\text{fly}}[n]$  and  $E_{m,u}^{\text{comp}}[n]$  are fixed values that do not affect the convexity of problem  $\mathbf{P}'_{1,1}$ . Consequently, the objective function  $\rho$  is the sum of three linear expressions about  $x_a^u[n]$  and  $y_a^m[n]$ , rendering it a linear function. Besides, two variables (i.e.,  $\overline{\mathbf{X}}$  and  $\overline{\mathbf{Y}}$ ) are involved in the sub-problem. Therefore, problem  $\mathbf{P}_{1,1'}$  is a convex NLP problem [38].

According to Theorem 2, solving problem  $\mathbf{P}_{1.1'}$  directly remains challenging since it is a convex NLP. Therefore, we iteratively optimize  $\overline{\mathbf{X}}$  and  $\overline{\mathbf{Y}}$  by splitting  $\mathbf{P}_{1.1'}$ . Specifically, it can be expressed as:

$$\mathbf{P}_{\mathbf{1},\mathbf{1}'}^{\overline{\mathbf{X}}}: \min_{\overline{\mathbf{X}}} \rho\left(\overline{\mathbf{X}},\widehat{\mathbf{F}},\widehat{\mathbf{Q}}\right)$$
(25a)

s.t. 
$$(24c), (24d), (23a).$$
 (25b)

$$\mathbf{P}_{\mathbf{1},\mathbf{1}'}^{\overline{\mathbf{Y}}}: \quad \min_{\overline{\mathbf{Y}}} \quad \rho\left(\overline{\mathbf{Y}},\widehat{\mathbf{F}},\widehat{\mathbf{Q}}\right)$$
(26a)

s.t. 
$$(24b), (24e), (23b).$$
 (26b)

Then, the optimal solution of  $\mathbf{X}^r$  can be obtained by applying CVX. However,  $\mathbf{X}^r$  is a continuous variable within the closed interval of [0, 1] while the decision of task offloading is a binary variable. Hence, the threshold rounding method [39] is applied to transform the relaxed  $\mathbf{X}^r$  into binary variables. Specifically, each element  $x^* \in \mathbf{X}^r$  is transformed as:

$$x^* = \begin{cases} 1, \text{ if } x^* \ge \delta, \\ 0, \text{ otherwise,} \end{cases}$$
(27)

where  $\delta \in (0, 1)$  is a positive rounding threshold.

Therefore, the optimal decisions of task offloading for

users can be obtained as  $X^*$ , and problem  $P_{1.1'}^{\overline{X}}$  is solved. Similarly, problem  $P_{1.1'}^{\overline{Y}}$  can be solved using the analogous method. As a result, the optimal solutions of task offloading can be obtained as  $X^*$  and  $Y^*$ , and thus  $A^r$  can be obtained.

However, the integrity gap is significant because the rounding process from continuous variables to binary variables may violate the constraint. Therefore, to overcome this issue after rounding, we denote  $\Delta_1$  and  $\Delta_2$  as the maximum violation of constraints (24c) and (24d), respectively. As described in [40], the integrity gap  $\zeta$  as follows:

$$\zeta = \min_{\overline{\mathbf{X}}} \frac{\rho}{\rho + \xi \Delta},\tag{28}$$

where  $\Delta = \Delta_1 + \Delta_2$ , and  $\zeta$  is the weight of  $\Delta$ . Furthermore, the optimal solutions are accepted when  $\zeta = 1$ , which is proved by Theorem 3.

**Theorem 3.** There is no violation of the constraints when  $\zeta = 1$ .

*Proof.* Taking  $X^*$  as an example, constraints (24c) and (24d) are modified as follows:

$$\sum_{u=1}^{U} x_a^u[n] \le 1 + \Delta_1,$$
(29a)

$$x_a^u[n]T_{\text{local}}[n] \le \tau_u + \Delta_2, \tag{29b}$$

where  $\Delta_1$  and  $\Delta_2$  are obtained as follows:

$$\Delta_1 = \max\left\{\sum_{u=1}^{U} x_a^u[n] - 1, 0\right\},$$
(30a)

$$\Delta_2 = \max\left\{x_a^u[n]T_{\text{local}}[n] - \tau_u, 0\right\},\tag{30b}$$

where the solution of  $\rho$  is obtained through relaxing the variables  $\overline{\mathbf{X}}$ , while the solution of  $\rho + \xi \Delta$  is obtained after rounding the relaxed variables. We consider that the best rounding is achieved when  $\zeta$  ( $\zeta \leq 1$ ) is closer to 1. In other words,  $\zeta = 1$ , when  $\Delta_1 = 0$  and  $\Delta_2 = 0$ . Similarly, the same proof also applies to  $\mathbf{Y}^*$ .

#### 4.3 Computation Resource Allocation

Given the optimized task offloading  $\mathbf{A}^r = \{a_{u,m}^r[n], \forall u, m, n\}$  for users and the UAV trajectory control  $\widehat{\mathbf{Q}} = \{\widehat{\mathbf{q}}_m[n], \forall m, n\}$ , the computation resource allocation problem can be expressed as:

$$\mathbf{P_{1.2}}: \min_{\mathbf{F}} \rho\left(\mathbf{A^r}, \mathbf{F}, \widehat{\mathbf{Q}}\right)$$
(31a)

s.t. 
$$0 \le f_{m,u}[n] \le f_m^{\max}, \ \forall m, n,$$
 (31b

$$\sum_{u=1}^{\infty} f_{m,u}[n] \le f_m^{\max}, \ \forall u, m, n.$$

$$(31c)$$

It can be deduced from Theorem 4 that (31a) is convex. The sub-problem of computation resource allocation is solved by adopting the KKT method. This is because KKT method can effectively deal with complex problems by transforming original problems into optimization problems with equality constraints, and improve the solving efficiency when considering multiple constraints.

# **Theorem 4.** Problem $P_{1,2}$ is a convex optimization problem.

*Proof.* The first-order derivative of  $\rho$  with respect to  $f_{m,u}[n]$  is calculated as:

$$\frac{\partial \rho}{\partial f_{m,u}} = -\frac{a'_{u,m}[n]D_u[n]C_u}{f_{m,u}[n]^2} + 2\kappa_m a^r_{u,m}[n]D_u[n]C_u f_{m,u}[n].$$
(32)

The second-order derivative of  $\rho$  with respect to  $f_{m,u}[n]$  is calculated as:

$$\frac{\partial^2 \rho}{\partial f_{m,u}^2} = -\frac{2a_{u,m}^r[n]D_u[n]C_u}{f_{m,u}[n]^3} + 2\kappa_m a_{u,m}^r[n]D_u[n]C_u.$$
(33)

It is clear that the second derivative  $\frac{\partial^2 \rho}{\partial f_{m,u}^2} \geq 0$ , and objective function  $\rho$  is a convex function with regard to  $f_{m,u}[n]$ . Therefore, problem  $\mathbf{P_{1.2}}$  is convex and has an optimal solution.

Hence, the slater condition is satisfied and the problem can be solved by using partial Lagrange function, which is formulated as:

$$L(F,\lambda) = \rho(\mathbf{A^r}, \mathbf{F}, \widehat{\mathbf{Q}}) + \lambda \left(\sum_{u=1}^{\mathbf{U_0}} f_{m,u}[n] - f_m^{\max}\right), \quad (34)$$

where  $\lambda \ge 0$  is the Lagrange multiplier related to the computation resource constraint of UAV.

Furthermore, the KKT conditions are used to obtain the optimal computation resource allocation  $\mathbf{F}^{\mathbf{r}}$ . Specifically, by differentiating  $L(F, \lambda)$  with respect to  $f_{m,u}^{r}[n]$  and setting the result as 0, which can be expressed as:

$$2\kappa_m a_{u,m}^r[n] D_u[n] C_u f_{m,u}^r[n]^3 + \lambda^* f_{m,u}^r[n]^2 - a_{u,m}^r[n] D_u[n] C_u = 0.$$
(35)

Then, the optimal computation resource allocation solution  $\mathbf{F}^{\mathbf{r}}$  can be achieved by applying the bisection algorithm [41], as shown in Algorithm 1. Specifically, the search accuracy threshold  $\varepsilon$ , lower bound  $\lambda^{\min}$  and upper bound  $\lambda^{\max}$  are set firstly (Line 1). Then, enter the loop of iterations until condition  $\lambda^{\max} - \lambda^{\min} \ge \varepsilon$  is satisfied (Line 2). Furthermore, in each iteration, the algorithm determines the value of  $\lambda$  by bisection and performs the calculation of the computation resource allocation  $f_{m,u}^r[n]$ , then compares it with the maximum resource owned by the UAV  $f_m^{\max}$  and updates  $\lambda$  if the condition is met (Lines 3 to 12).

Algorithm 1: Bisection Algorithm-based Computa-				
tion Resource Allocation				
Input: The maximum resource owned by the UAV				
$f_m^{\max}$ .				
<b>Output:</b> The optimal computation resource				
allocation $\mathbf{F}^r = \{f_{m,u}^r[n], m \in \mathcal{M}\}.$				
1 <b>Initialization:</b> Search accuracy threshold: $\varepsilon$ , the				
lower bound $\lambda^{\min} = 0$ and the upper bound				
$\lambda^{\max} = \lambda^{\text{bound}};$				
2 while $\lambda^{max} - \lambda^{min} \geq \varepsilon$ do				
3 Define $\lambda = \frac{\lambda^{\min} + \lambda^{\max}}{2}$ ;				
4 for $u \in \mathbf{U}_0$ do				
5 Compute $f_{m,n}^{T}[n]$ by substituting $\lambda$ into Eq.				
(35);				
6 end				
7 <b>if</b> $f_{m,n}^r[n] \ge f_m^{max}$ then				
$\mathbf{s}     \lambda^{\min} = \lambda;$				
9 else				
10 $\lambda^{\max} = \lambda;$				
11 end				
12 end				
13 return $\mathbf{F}^r = \{f^r_{m,u}[n], m \in \mathcal{M}\}.$				

# 4.4 UAV Trajectory Control

Given the optimized decisions of task offloading  $\mathbf{A}^r = \{a_{u,m}^r[n], \forall u, m, n\}$  and computation resource allocation  $\mathbf{F}^r = \{f_{m,u}^r[n], \forall u \in \mathbf{U}_0, m, n\}$ , the UAV trajectory control problem can be expressed as:

$$\mathbf{P_{1.3}}: \min_{\mathbf{Q}} \rho\left(\mathbf{A^r, F^r, Q}\right)$$
(36a)

s.t. 
$$\mathbf{q}_m[n+1] = \mathbf{q}_m[n] + v_m[n]\delta_t + \frac{1}{2}a_m[n]\delta_t^2, \ \forall m, n,$$
(36b)

$$v_m[n+1] = v_m[n] + a_m[n]\delta_t, \ \forall m, n,$$
 (36c)

$$||v_m[n]|| \le V_{\max}, ||v_m[n]|| \ge V_{\min}, \forall m, n,$$
(36d)

$$\|\mathbf{q}_m[n] - \mathbf{q}_i[n]\| \ge D_{\min}, \ \forall m \neq i, n, \tag{36e}$$

$$\mathbf{q}_m[1] = \mathbf{q}_m[N], \ \forall m, \tag{36f}$$

$$\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\| \le D_{\max} = V_{\max}\delta_t, \ \forall m, n.$$
(36g)

Since the constraints (36d) and (36e) are non-convex, the sub-problem  $P_{1.3}$  is non-convex. In view of the fact that the SCA method shows reliable convergence when dealing with non-convex optimization problems, and its property of effectively handling complex constraints, we choose to use SCA method to solve the sub-problem of UAV trajectory control, so as to improve the efficiency and speed of problem solving.

According to Lemma 1, we can transform the nonconvex constraint into a lower bound on a convex function.

**Lemma 1.** For a given local point  $x^r$ , we have the following inequality:

$$x^{2} \ge 2x^{r} \left(x - x^{r}\right) + \left(x^{r}\right)^{2}.$$
(37)

*Proof.* we first define a quadratic function as follows:  $f(x) = x^2$ . It can be easily observed that f(x) is a convex function. Since any convex function can obtain its lower bound by adopting its first-order Taylor expansion at a local point, the inequality holds in (37) [24].

For the non-convex constraint (36d), we first rewrite the non-convex constraints in (36d) as:

$$||v_m[n]||^2 \ge V_{\min}^2, ||v_m[n]||^2 \le V_{\max}^2.$$
 (38)

Furthermore, according to Lemma 1, given the local point  $v_m^r[n]$ , we have:

$$\|v_m[n]\|^2 \ge \|v_m^r[n]\|^2 + 2(v_m^r[n])^T (v_m[n] - v_m^r[n]) \triangleq \vartheta_m^r[n].$$
(39)

Therefore, the non-convex constraint (36d) can be bounded as:

$$\vartheta_m^r[n] \ge V_{\min}^2, \ \vartheta_m^r[n] \le V_{\max}^2.$$
(40)

For the non-convex constraint (36e), we introduce an auxiliary variable  $D_{m,i}[n]$  as:

$$D_{m,i}[n] = \|\mathbf{q}_m[n] - \mathbf{q}_i[n]\|, \ \forall n, m \neq i.$$
(41)

Furthermore, since the distance between any two UAVs in the system should be greater than or equal to the defined safe distance, Eq. (41) can be further rewritten as  $||D_{m,i}||^2 \ge D_{\min}^2$ . Thus, given the fixed point  $D_{m,i}^r[n]$ , we have:

$$||D_{m,i}[n]||^{2} \geq 2 \left( D_{m,i}^{r}[n] \right)^{T} \left( D_{m,i}[n] - D_{m,i}^{r}[n] \right) + \left\| D_{m,i}^{r}[n] \right\|^{2}.$$
(42)

Therefore, for the given points of UAV  $\mathbf{q}_i^r[n]$  and  $\mathbf{q}_m^r[n]$ , the non-convex constraint (36e) can be converted to:

$$D_{\min}^{2} \leq \|\mathbf{q}_{m}^{r}[n] + \mathbf{q}_{i}^{r}[n]\|^{2} + 2\left(\mathbf{q}_{m}^{r}[n] - \mathbf{q}_{i}^{r}[n]\right)^{T} \\ \times \left(\mathbf{q}_{m}[n] - \mathbf{q}_{i}[n] - \mathbf{q}_{m}^{r}[n] + \mathbf{q}_{i}^{r}[n]\right) \\ \triangleq \zeta_{m,i}^{r}[n], \quad \forall n, m \neq i.$$

$$(43)$$

In conclusion, according to (40) and (43), sub-problem **P**<sub>1.3</sub> can be transformed into a convex problem **P**<sub>1.3'</sub> by introducing the auxiliary variables  $\vartheta_m^r[n]$  and  $\zeta_{m,i}^r[n]$ , which is as follows:

$$\mathbf{P_{1.3'}:} \quad \min_{\mathbf{Q}} \quad \rho\left(\mathbf{A^r, F^r, Q}\right) \tag{44a}$$

s.t. 
$$\vartheta_m^r[n] \ge V_{\min}^2, \ \vartheta_m^r[n] \le V_{\max}^2, \ \forall m, n,$$
 (44b)

$$\zeta_{m,i}^r[n] \ge D_{\min}^2, \ \forall n, m \neq i, \tag{44c}$$

$$(36b), (36c), (36f), (36g).$$
 (44d)

It can be directly solved by existing optimization software such as CVX [42].

**Theorem 5.** Problem  $\mathbf{P}_{1.3'}$  does not change the optimality of problem  $\mathbf{P}_{1.3}$ , i.e.,  $\rho_{1.3'}$  ( $\mathbf{A}^{\mathbf{r}}, \mathbf{F}^{\mathbf{r}}, \mathbf{Q}$ )  $\leq \rho_{1.3}$  ( $\mathbf{A}^{\mathbf{r}}, \mathbf{F}^{\mathbf{r}}, \mathbf{Q}$ ).

*Proof.* By applying the lower bound results in (40) and (43), it can be deduced that the set of feasible solutions for subproblem  $P_{1.3'}$  is included within that for  $P_{1.3}$ . Accordingly, the objective function value obtained from  $P_{1.3'}$  is always less than or equal to that from  $P_{1.3}$ , which indicates that there exists at least one local optimal solution [24].

#### 4.5 Main Steps of JTORATC and Analysis

In this section, the main steps of the JTORATC are shown in Algorithm 2. First, the decision parameters of task offloading, computation resource allocation, UAV trajectory, and iteration numbers are initialized (Line 1). Then, the optimal decisions of task offloading, computation resource allocation and UAV trajectory control are obtained by solving the sub-problems  $P_{1.1}$ ,  $P_{1.2}$  and  $P_{1.3}$  alternately (Lines 2 to 8). In addition, the convergence and computational complexity of JTORATC are proved respectively.

# 4.5.1 Convergence and Computation Complexity

The convergence of JTORATC is analyzed as follows:

$$\rho\left(\mathbf{A}^{r}, \mathbf{F}^{r}, \mathbf{Q}^{r}\right) \stackrel{(a)}{\geq} \rho\left(\mathbf{A}^{r+1}, \mathbf{F}^{r}, \mathbf{Q}^{r}\right), \tag{45a}$$

$$\stackrel{(b)}{\geq} \rho\left(\mathbf{A}^{r+1}, \mathbf{F}^{r+1}, \mathbf{Q}^{r}\right), \tag{45b}$$

$$\stackrel{(c)}{\geq} \rho\left(\mathbf{A}^{r+1}, \mathbf{F}^{r+1}, \mathbf{Q}^{r+1}\right). \tag{45c}$$

Since task offloading is solved for a given resource allocation and UAV trajectory, the inequality (45a) holds. Furthermore, inequality (45b) is satisfied because the KKT condition is introduced to solve resource allocation sub-problem. Moreover, the inequality (45c) is due to the sub-optimality of UAV trajectory. Besides, the objective function  $\rho$  (**A**, **F**, **Q**) is always positive owing to its non-negative expression. Therefore, the objective function is always non-increasing after every iteration, which is also finitely lower-bounded.

The computational complexity of problem  $\mathbf{P_{1.1}}$  is  $\mathcal{O}\left(K\left(\log_2(\frac{1}{\epsilon}+1)\right)$  [43] based on the analysis of Theorem 2 and Theorem 3, where K = U + M, U is the number of users, M is the number of UAVs, and  $\epsilon$  is the tolerance of accuracy. Furthermore, problem  $\mathbf{P_{1.2}}$  is solved

- **Input:** Task offloading  $\mathbf{A}^0$ , computation resource allocation  $\mathbf{F}^0$ , UAV trajectory  $\mathbf{Q}^0$ , and iteration index  $r^0$ .
- **Output:** Optimal task offloading  $\mathbf{A}^{opt}$ , computation resource allocation  $\mathbf{F}^{opt}$  and UAV trajectory  $\mathbf{Q}^{opt}$ .
- 1 **Initialization:** The iteration index r = 1, the tolerance of accuracy  $\epsilon = 10^{-4}$ ,  $\mathbf{A}^0 = \emptyset$ ;

2 repeat

- 3 Fix { $\mathbf{F}^r$ ,  $\mathbf{Q}^r$ }, obtain the optimal solution of the problem  $\mathbf{P}_{1,1}$  is  $\mathbf{A}^{r+1}$ ;
- 4 Fix { $\mathbf{A}^{r+1}, \mathbf{F}^r, \mathbf{Q}^r$ }, obtain the optimal solution of the problem  $\mathbf{P}_{1,2}$  based on Algorithm 1 is  $\mathbf{F}^{r+1}$ ;
- 5 Fix { $\mathbf{A}^{r+1}$ ,  $\mathbf{F}^{r+1}$ ,  $\mathbf{Q}^{r}$ }, obtain the optimal solution of the problem  $\mathbf{P}_{1,3}$  is  $\mathbf{Q}^{r+1}$ ;
- 6 Update { $\mathbf{A}, \mathbf{F}, \mathbf{Q}$ }<sup>*r*</sup>  $\leftarrow$  { $\mathbf{A}^{r+1}, \mathbf{F}^{r+1}, \mathbf{Q}^{r+1}$ } and *r* = *r* + 1; 7 until  $\frac{\rho_{r+1} - \rho_r}{r} < \epsilon$ ;
- s return { $\stackrel{\rho_r}{\mathbf{A}^{opt}}, \stackrel{-}{\mathbf{F}^{opt}}, \mathbf{Q}^{opt}$ }.

based on the bisection algorithm presented in Algorithm 1, and its computational complexity can be calculated as  $\mathcal{O}\left(\log_2\left(\frac{\lambda^{\max}-\lambda^{\min}}{\varepsilon}\right)\right)$  [41], where  $\lambda^{\min}$  and  $\lambda^{\max}$  are the lower and upper bounds of  $\lambda$ , respectively, and  $\varepsilon$  denotes the search accuracy. Moreover, problem  $\mathbf{P_{1.3}}$  involves solving non-convex problem, and the computational complexity is  $\mathcal{O}(M^{3.5}\log_2(\frac{1}{\epsilon}))$  [44]. Therefore, the proposed JTORATC approach converges in polynomial, and the computational complexity is  $\mathcal{O}(I_c(K(\log_2(\frac{1}{\epsilon}+1) + \log_2(\frac{\lambda^{\max}-\lambda^{\min}}{\varepsilon}) + M^{3.5}\log_2(\frac{1}{\epsilon})))$ , where  $I_c$  is the number of outer iterations of Algorithm 2.

# **5** SIMULATION RESULTS

In this section, simulation results are presented to validate the effectiveness of the proposed approach.

# 5.1 Simulation Setup

We perform simulations to verify the validity of our proposed JTORATC method. Specifically, all the simulations are performed in MATLAB R2022b on a desktop computer equipped with an Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21GHz and 8 GB RAM.

**Scenarios.** We consider a multi-UAV-assisted MEC system where 2 UAVs, 4 UAVs and 6 UAVs are deployed to offer offloading services to 8 users in a  $2500 \times 3000 \text{ m}^2$  rectangular area. The timeline is set as T = 100 s which is divided into 50 time slots.

**Parameters.** The default values of the simulation parameters are listed in Table 3.

**Benchmarks.** This work evaluates the proposed JTORATC in comparison with the following schemes.

 Random offloading and JTORATC-based resource allocation and trajectory control (ROJRATC): the task offloading strategies of users are decided randomly, while the computation resource allocation and the trajectories of UAVs are decided based on the proposed JTORATC.

TABLE 3 Simulation parameters

Symbol	Meaning	Default value
$f_u$	Computing capability of user	340 MHz
	u	
$p_u$	Transmission power of user $u$	30 dBm
$ au_u$	Deadline of the task	[0.1, 75]s
H	Fixed flight altitude of UAV $m$	100 m
$\mathbf{q}_1^I$	Initial position of UAV1	[800, 1200]
$\mathbf{q}_2^I$	Initial position of UAV2	[2000, 1000]
$V_{min}$	Minimum speed of UAV m	20 m/s
$V_{max}$	Maximum speed of UAV $m$	60 m/s
$f_m^{\max}$	Computing capability of UAV	1200 MHz
	m	
$\kappa_m$	Effective switching	$10^{-27}$
	capacitance of UAV m	
$D_{min}$	Safe distance between UAVs	10 m
В	Channel bandwidth	20 MHz
$\sigma^2$	Noise power	-100 dBm

- Nearby offloading and JTORATC-based resource allocation and trajectory control (NOJRATC): the tasks of each user are offloaded to the nearest UAV, while the computation resource allocation and the trajectory control are decided based on the proposed JTORATC.
- Many-to-many matching-based offloading and JTORATCbased resource allocation and trajectory control (MOJRATC): the task offloading strategies are decided by using a matching-to-matching mechanism [45], while the computation resource allocation and the trajectories of UAVs are decided based on the proposed JTORATC.
- Equalizing resources, JTORATC-based offloading and trajectory control (ERJOTC): the computation resource allocation of UAVs are decided averagely while the strategies of task offloading and trajectory control are determined based on the proposed JTORATC.
- JTORATC-based offloading and resource allocation, circular trajectory (JORACT): the task offloading strategies and the computation resource allocation are decided based on the proposed JTORATC, while the UAVs fly following the circular trajectories.
- JTORATC-based offloading and resource allocation, predefined trajectory (JORAPT): the task offloading strategies and the computation resource allocation are decided based on the proposed JTORATC, while the UAVs follow predefined trajectories.

# 5.2 Evaluation Results

In this section, we first present simulation results to evaluate the performance of our proposed JTORATC. Then, we compare the effects of different scenario setups on the performance of the proposed JTORATC and benchmark schemes.

#### 5.2.1 System Performance

Figs. 2(a), 2(b), 2(c), and 2(d) show the comparison results of the objective function value, total task completion delay, total UAV energy consumption, and total amount of offloaded tasks of seven approaches with the increasing of the number of UAVs. It can be observed from Fig. 2 that as the number of UAVs increases, the objective function value and total UAV energy consumption of all approaches



Fig. 2. System performance with UAV computation capacity. (a) Objective function value. (b) Total task completion delay. (c) Total UAV energy consumption. (d) Total amount of offloaded tasks.

except the NOJRATC approach increase. This is because that the increase in the number of UAVs may lead to complex task scheduling, fierce competition for computing resources and cross-conflict of trajectories. However, considering the spatio-temporal relationship of tasks, the NOJRATC simplifies task scheduling, effectively uses resources and reduces conflicts on flight trajectories.

Moreover, the proposed JTORATC maintains superior performance compared to other approaches as the number of UAVs increases, and this can be attributed to several reasons as follows. First, ROJRATC, NOJRATC and MOJRATC mainly focus on optimizing the task offloading. The random offloading of ROJRATC, the offloading of the nearest MEC server of NOJRATC, and the matching offloading of MOJRATC may lead to inflexibly resource allocation of the MEC servers, triggering congestion and excessive resource usage. Second, ERJOTC mainly focuses on optimizing the computation resource allocation. It can be observed from Fig. 2(c) that as the number of UAVs increases, ERJOTC is slightly better than the proposed JTORATC in terms of total UAV energy consumption, but far worse than the proposed JTORATC in terms of the objective function value and total task completion delay. This is because the proposed JTORATC allows for more flexible task collaborative optimization, adaptation to different task requirements and dynamic changes, as well as latency reduction through parallel processing, thus improving the overall system performance, while ERJOTC can avoid excessive concentration of resources on a few UAVs. Furthermore, JORACT and JORAPT mainly focuses on the UAV trajectory control. The circular trajectory of JORACT and predefined trajectory of JORAPT have poor performance compared with the proposed JTORATC. This is because the proposed JTORATC allows dynamic adjustment of paths to accommodate realtime changes, resulting in more efficient and flexible task execution.

Accordingly, it can be concluded that the proposed JTORATC achieves optimal performance in multi-UAV-assisted MEC systems compared to other approaches.

#### 5.2.2 Performance Evaluation

In this section, we evaluate the impact of different parameters on the system performance based on two UAVs. In this process, we keep the rest of parameters with default values and only vary the UAV computation capacity, task computation intensity, task size, and number of users. Through such exploration, we further verify that our proposed JTORATC approach has significant robustness and scalability in the system.

(1) Impact of UAV Computation Capacity. Figs. 3(a), 3(b), and 3(c) show the effects of UAV computation capacity on the objective function value, total task completion delay, total UAV energy consumption, and total amount of offloaded tasks. It can be observed from Fig. 3(a) that when the UAV computation capacity is less than 400 MHz, the ERJOTC approach achieves a lower objective function value than the proposed JTORATC. However, it is essential to recognize that while ERJOTC may find a local optimal solution in certain cases, ERJOTC cannot guarantee a global optimal solution. Therefore, this discrepancy is expected. In addition, as the computation capacity of UAVs increases, the proposed JTORATC is superior to other schemes in terms of objective function. The reason is that JTORATC jointly optimizes task offloading, computing resource allocation, and UAV trajectory control to adapt the computing capabilities of different UAV, resulting in satisfactory performance.

From Figs. 3(b) and 3(c), it is worth noting that when the UAV computation capacity is less than 600 MHz, the ROJRATC approach obtains a lower total task completion delay than the proposed JTORATC. Similarly, this outcome is expected since the proposed JTORATC is constrained by insufficient computing resources, leading to longer task completion times. Besides, with an increase in the UAV computation capacity, ROJRATC achieves a similar total completion delay to the proposed JTORATC by using the random offloading approach. However, the excessive focus on load balancing in ROJRATC causes higher energy consumption, leading to the relatively elevated energy consumption of UAVs.

From Fig. 3(d), we can see that for the total amount of offloaded tasks, the seven approaches all show a constant trend regardless of the varying UAV computation capacity, since the total amount of tasks completed by the UAV depends on the amount of tasks generated by the users, and has nothing to do with the computation capacity of UAV.

In summary, the results in Fig. 3 illustrate the adaptability of the proposed JTORATC with varying computing capacities of UAVs.

(2) Impact of Task Computation Intensity. Figs. 4(a), 4(b), 4(c), and 4(d) reveal the effects of task computation



Fig. 3. System performance with numbers of UAVs. (a) Objective function value. (b) Total task completion delay. (c) Total UAV energy consumption. (d) Total amount of offloaded tasks.



Fig. 4. System performance with task computation intensity. (a) Objective function value. (b) Total task completion delay. (c) Total UAV energy consumption. (d) Total amount of offloaded tasks.

intensity on the objective function value, total task completion delay, total UAV energy consumption, and total amount of offloaded tasks. As can be seen, as the computational intensity increases, the seven approaches exhibit similar upward trends in terms of objective function value, total task completion delay and total UAV energy consumption. This is because higher computational intensity indicates heavier system workloads, which further leads to higher computational requirements. Consequently, more frequent task offloading, computation resource allocation and adjustments of UAV trajectory are necessary, which potentially results in additional overhead on computation and energy consumption.

Moreover, it can be seen from Fig. 4(a) that the proposed JTORATC outperforms other approaches in terms of objective function value. This is because JTORATC iteratively optimizes the decisions of task offloading, computation resource allocation, and UAV trajectory control to adapt to scenarios with varying workloads. From Fig. 4(b), we can observe that with the increasing of task computation intensity, ROJRATC achieves a similar total completion delay to the proposed JTORATC by using the random offloading approach. However, the performance of ROJRATC in minimizing the objective function value and total UAV energy consumption is not as good as our proposed ITORATC approach. This is because the ROJRATC mainly relies on a random offloading strategy, which could not provide sufficient performance benefits in terms of the objective function, completion delay, and energy consumption, as the computational intensity of the tasks increases. In contrast,

the proposed JTORATC takes a more comprehensive approach for system optimization.

From Fig. 4(c), it can be observed that ERJOTC shows optimal performance in terms of total UAV energy consumption. This is because ERJOTC helps to ensure the load balance of each UAV during task execution, reduce the excessive use of computing resources, and achieve an even distribution of overall computing energy consumption. However, the performance of ERJOTC is slightly inferior to our proposed JTORATC approach in objective function value and total task completion delay aspects. While from Fig. 4(d), we can see that for the total amount of offloaded tasks, the seven approaches all show a constant trend regardless of the varying task computation intensity. Similarly, because the total amount of tasks generated by the UAV depends on the amount of tasks computation intensity.

In conclusion, the simulation results in Fig. 4 indicate the superiority of the proposed JTORATC in both light and heavy computational scenarios.

(3) Impact of Task Size. Figs. 5(a), 5(b), 5(c), and 5(d) depict the effects of task size on the objective function value, the total task completion delay, the total UAV energy consumption, and the total amount of offloaded tasks. It can be seen from Fig. 5 that the seven approaches exhibit upward trends as the task size increases. This is because the increasing task size indicates heavier workloads, which could further lead to higher costs of processing delay and energy consumption.

From Figs. 5(a), 5(b), and 5(c), it can be observed that when the task size is small (less than 1.5MB), the perfor-



Fig. 5. System performance with task size. (a) Objective function value. (b) Total task completion delay. (c) Total UAV energy consumption. (d) Total amount of offloaded tasks.



Fig. 6. System performance with number of users. (a) Objective function value. (b) Total task completion delay. (c) Total UAV energy consumption. (d) Total amount of offloaded tasks.

mance of ERJOTC is better than the proposed JTORATC approach in objective function and total task completion delay aspects. However, when the task size is large (more than 3MB), ERJOTC has the worst performance of all approaches. Moreover, for the total UAV energy consumption, ERJOTC shows the best performance. This is because when the task size is small, resources are allocated uniformly, then ERJOTC may be more efficient. However, when the task size is large, ERJOTC does not fully consider the complexity and different requirements of the task, which may lead to performance degradation. In addition, the average allocation strategy of ERJOTC ensures a relatively uniform distribution of tasks among UAVs and avoids excessive loads, thus reducing the energy consumption of UAVs.

Moreover, it can be seen from Fig. 5(d) that the seven approaches all show a upward trend as task size increases for the total amount of offloaded tasks, because the total amount of tasks generated by users is increasing, and the characteristics and requirements of the tasks may also change.

For the total amount of offloaded tasks, the seven approaches all show a upward trend as task size increases since the total amount of tasks generated by users is increasing, and the characteristics and requirements of the tasks may also change.

Accordingly, the simulation results in Fig. 5 demonstrate the superiority of the proposed JTORATC to adapt to heavyloaded scenarios.

(4) Impact of Number of Users. Figs. 6(a), 6(b), 6(c), and 6(d) present that with the increasing of the number of users, the objective function value, the total task completion delay,

the total UAV energy consumption, and the total amount of offloaded tasks of the seven approaches gradually increase. The reason is that as the number of users increases, the total number of tasks that need to be handled in the system also increases. This leads to a surge in resource competition and an increase in the load of communication networks.

Fig. 6(a) illustrates that our proposed JTORATC approach achieves the best performance in minimizing the value of the objective function. This can be attributed to the fact that our proposed JTORATC successfully balances the system parameters by employing iterative optimization, and thus obtains a significant advantage in the overall performance.

From Figs. 6(b), 6(c), and 6(d), it can be observed that ROJRATC performs best in terms of total task completion delay. The random nature of ROJRATC can lead to tasks being assigned and completed more quickly. However, it should be noted that ROJRATC performs relatively poorly in terms of total UAV energy consumption and total amount of offloaded tasks. This suggests that we must fully balance energy consumption and task completion efficiency while considering delay. In contrast, our proposed JTORATC approach takes this balance into account. Moreover, the ER-JOTC has a good performance in reducing the total UAV energy consumption, mainly because evenly allocating resources avoids some resources being idle or overloaded. At the same time, because of the uniform allocation of resources, there is a certain sacrifice in the task completion delay, which is far less than our proposed JTORATC approach. This may be because task offloading is not flexible enough or resources cannot be optimized for the specific

needs of the task.

Consequently, this set of simulation results highlights the importance of our proposed JTORATC approach in terms of scalability and robustness of the system.

# 6 CONCLUSION

In this work, we study the multi-objective optimization of task offloading, computation resource allocation and UAV trajectory control in multi-UAV-assisted MEC systems. We first formulate the multi-objective optimization problem with the aim of minimizing the total task completion delay, reducing the total UAV energy consumption, and maximizing the total amount of offloaded tasks. To solve the problem, we transform the multi-objective optimization problem into a single-objective optimization problem and then propose the JTORATC to solve the transformed problem at a lower cost. Simulation results demonstrate that the proposed JTORATC has superior performance in terms of the total task completion delay and total UAV energy consumption. Specifically, JTORATC demonstrates better adaptability to the scenarios with varying computing capacities of UAVs, and it shows superior scalability in both light and heavy workload scenarios.

# REFERENCES

- F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A survey on edge computing systems and tools," *Proc. IEEE*, vol. 107, no. 8, pp. 1537–1562, 2019.
- [2] M. Asim, Y. Wang, K. Wang, and P. Huang, "A review on computational intelligence techniques in cloud and edge computing," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 6, pp. 742–763, 2020.
- [3] L. M. Abualigah, A. Diabat, and M. A. E. Aziz, "Intelligent workflow scheduling for big data applications in IoT cloud computing environments," *Clust. Comput.*, vol. 24, no. 4, pp. 2957–2976, 2021.
- [4] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.
- [5] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2961– 2991, 2018.
- [6] 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.*, vol. 23, no. 2, pp. 1655–1673, 2024.
  [7] Y. Qu, H. Dai, L. Wang, W. Wang, F. Wu, H. Tan, S. Tang, G. W. Wang, F. Wu, H. Tan, S. Tang, M. Wang, F. Wu, H. Tang, S. Tang, M. Wang, M. Wang, F. Wu, H. Tang, S. Tang, M. Wang, F. Wu, H. Tang, S. Tang, M. Wang, M. Wang, F. Wu, H. Tang, S. Tang, M. Tang, M. Wang, F. Wu, H. Tang, S. Tang, M. Wa
- [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] W. Khawaja, I. Güvenç, D. W. Matolak, U. Fiebig, and N. Schneckenburger, "A survey of Air-to-Ground propagation channel modeling for unmanned aerial vehicles," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2361–2391, 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.*, vol. abs/2308.01511, 2023.
- [10] N. N. Ei, M. Alsenwi, Y. K. Tun, Z. Han, and C. S. Hong, "Energyefficient resource allocation in multi-uav-assisted two-stage edge computing for beyond 5g networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16421–16432, 2022.
- Syst., vol. 23, no. 9, pp. 16421–16432, 2022.
  [11] G. Sun, L. He, Z. Sun, Q. Wu, S. Liang, J. Li, D. Niyato, and V. C. M. Leung, "Joint task offloading and resource allocation in aerial-terrestrial UAV networks with edge and fog computing for post-disaster rescue," *IEEE Trans. Mob. Comput.*, pp. 1–18, 2024.
- [12] 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.

- [13] 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.
- [14] 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.*, vol. 62, no. 1, pp. 62–68, 2024.
- [15] 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.
- [16] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, 2016.
- [17] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, 2019.
- [18] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energyefficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3424–3438, 2020.
- [19] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Trans. Ind. Informatics*, vol. 16, no. 8, pp. 5505– 5516, 2020.
- [20] Y. K. Tun, N. D. Tri, K. Kim, M. Alsenwi, W. Saad, and C. S. Hong, "Collaboration in the sky: A distributed framework for task offloading and resource allocation in multi-access edge computing," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24221–24235, 2022.
- [21] H. Guo, Y. Wang, J. Liu, and C. Liu, "Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 1, pp. 347–359, 2024.
- [22] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, 2020.
- [23] Y. Chen, D. Pi, S. Yang, Y. Xu, J. Chen, and A. W. Mohamed, "HNIO: A hybrid nature-inspired optimization algorithm for energy minimization in UAV-assisted mobile edge computing," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 3, pp. 3264–3275, 2022.
- [24] J. Ji, K. Zhu, D. Niyato, and R. Wang, "Joint cache placement, flight trajectory, and transmission power optimization for multi-UAV assisted wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 8, pp. 5389–5403, 2020.
- [25] W. Lee and T. Kim, "Multiagent reinforcement learning in controlling offloading ratio and trajectory for multi-UAV mobile-edge computing," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 3417–3429, 2024.
- [26] Y. M. Park, S. S. Hassan, Y. K. Tun, Z. Han, and C. S. Hong, "Joint trajectory and resource optimization of MEC-assisted UAVs in Sub-THz networks: A resources-based multi-agent proximal policy optimization DRL with attention mechanism," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2003–2016, 2024.
- [27] Q. Wu, J. Chen, Y. Xu, N. Qi, T. Fang, Y. Sun, and L. Jia, "Joint computation offloading, role, and location selection in hierarchical multicoalition UAV MEC networks: A stackelberg game learning approach," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18293–18304, 2022.
- [28] J. Zhang, L. Zhou, F. Zhou, B. Seet, H. Zhang, Z. Cai, and J. Wei, "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114–2125, 2020.
- [29] Y. Wang, W. Chen, T. H. Luan, Z. Su, Q. Xu, R. Li, and N. Chen, "Task offloading for post-disaster rescue in unmanned aerial vehicles networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1525– 1539, 2022.
- [30] B. Yang, G. Mao, M. Ding, X. Ge, and X. Tao, "Dense small cell networks: From noise-limited to dense interference-limited," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4262–4277, 2018.
- [31] A. Boumaalif and O. Zytoune, "Power distribution of device-todevice communications under nakagami fading channel," *IEEE Trans. Mob. Comput.*, vol. 21, no. 6, pp. 2158–2167, 2022.
- [32] 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.

- [33] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," J. VLSI Signal Process., vol. 13, no. 2-3, pp. 203–221, 1996.
- [34] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, 2016.
- scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, 2016.
  [35] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond," *Proc. IEEE*, vol. 107, no. 12, pp. 2327–2375, 2019.
- [36] 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.
- [37] I. Alsyouf and S. Hamdan, "A multi-objective optimization of maintenance policies using weighted comprehensive criterion method (wccm)," pp. 1–4, 2017.
- [38] 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.
- [39] K. M. Elbassioni and S. Ray, "Threshold rounding for the standard LP relaxation of some geometric stabbing problems," CoRR, vol. abs/2106.12385, 2021.
- [40] U. Feige, M. Feldman, and I. Talgam-Cohen, "Oblivious rounding and the integrality gap," in *Proc. Leibniz Int. Proc. Inform.*, vol. 60, 2016, pp. 1–23.
- [41] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, 2019.
- [42] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," Online, Mar. 2014.
  [43] M. Hong, T. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z. Luo,
- [43] M. Hong, T. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z. Luo, "A block successive upper-bound minimization method of multipliers for linearly constrained convex optimization," *Math. Oper. Res.*, vol. 45, no. 3, pp. 833–861, 2020.
- [44] 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.
- [45] Z. Zhao, J. Shi, Z. Li, J. Si, P. Xiao, and R. Tafazolli, "Matchingaided-learning resource allocation for dynamic offloading in mmwave MEC system," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 11, pp. 7580–7591, 2023.



Zemin Sun received a BS degree in Software Engineering, an MS degree and a Ph.D degree in Computer Science and Technology from Jilin University, Changchun, China, in 2015, 2018, and 2022, respectively. Her research interests include vehicular networks, edge computing, and game theory.



**Qingqing Wu** (S'13-M'16-SM'21) received the B.Eng. and the Ph.D. degrees in Electronic Engineering from South China University of Technology and Shanghai Jiao Tong University (SJTU) in 2012 and 2016, respectively. From 2016 to 2020, he was a Research Fellow in the Department of Electrical and Computer Engineering at National University of Singapore. He is currently an Associate Professor with Shanghai Jiao Tong University. His current research interest includes intelligent reflecting surface (IRS), unmanned

aerial vehicle (UAV) communications, and MIMO transceiver design. He has coauthored more than 100 IEEE journal papers with 26 ESI highly cited papers and 8 ESI hot papers, which have received more than 18,000 Google citations. He was listed as the Clarivate ESI Highly Cited Researcher in 2022 and 2021, the Most Influential Scholar Award in AI-2000 by Aminer in 2021 and World's Top 2% Scientist by Stanford University in 2020 and 2021.



**Geng Sun** (S'17-M'19) received the B.S. degree in communication engineering from Dalian Polytechnic University, and the Ph.D. degree in computer science and technology from Jilin University, in 2011 and 2018, respectively. He was a Visiting Researcher with the School of Electrical and Computer Engineering, Georgia Institute of Technology, USA. He is an Associate Professor in College of Computer Science and Technology at Jilin University, and his research interests include wireless networks, UAV communications,

collaborative beamforming and optimizations.



Jiawen Kang received the M.S. degree and the Ph.D. degree from the Guangdong University of Technology, China, in 2015 and 2018. He is currently a full professor at the Guangdong University of Technology. He was a postdoc at Nanyang Technological University from 2018 to 2021, Singapore. His research interests mainly focus on blockchain, security, and privacy protection in wireless communications and networking.



Yixian Wang received a BS degree in Software engineering from Shanxi University, Shanxi, China, in 2023. She is currently working toward the MS degree in Computer Science and Technology at Jilin University, Changchun, China. Her research interests include vehicular networks and edge computing.



**Dusit Niyato** (Fellow, IEEE) received the B.Eng. degree from the King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include the Internet of Things (IoT), machine learning, and incentive mechanism design.



Victor C. M. Leung (Life Fellow, IEEE) is a Distinguished Professor of computer science and software engineering with Shenzhen University, China. He is also an Emeritus Professor of electrial and computer engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems at the University of British Columbia (UBC). His research is in the broad areas of wireless networks and mobile systems. He has co-authored more than 1300 journal/conference papers and book chapters.

Dr. Leung is serving on the editorial boards of IEEE Transactions on Green Communications and Networking, IEEE Transactions on Cloud Computing, IEEE Access, and several other journals. He received the IEEE Vancouver Section Centennial Award, 2011 UBC Killam Research Prize, 2017 Canadian Award for Telecommunications Research, and 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award. He co-authored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, 2017 IEEE Systems Journal Best Paper Award, 2018 IEEE CSIM Best Journal Paper Award, and 2019 IEEE TCGCC Best Journal Paper Award. He is a Life Fellow of IEEE, and a Fellow of the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of Highly Cited Researchers.