Joint Energy-Efficient Task Scheduling and Trajectory Optimization for Multi-UAV MEC Networks in Low-Altitude Economy

Zhiying Wang, Jinghui Chen, Gang Sun, *Senior Member, IEEE*, Hongfang Yu, *Senior Member, IEEE*, and Mohsen Guizani, *Life Fellow, IEEE*

***Abstract*— Driven by the increasing deployment of Internet of Things (IoT) devices and significant progress in Uncrewed Aerial Vehicle (UAV) technologies, UAV-assisted Mobile Edge Computing (UMEC) has emerged as an effective approach to address the challenges of constrained ground network coverage and insufficient computational resources. Leveraging their mobility, flexible deployment, and low cost, UAVs can dynamically support edge networks by providing computation and communication services for latency and energy sensitive tasks. Nevertheless, task scheduling and UAV trajectory optimization in UMEC systems still face significant challenges. To address the task scheduling problem for energy-sensitive tasks in the UMEC system, this paper first formulates a mathematical model of the system's energy consumption and establishes an optimization objective aimed to optimize the energy cost of UMEC system. Building upon this, we propose a task scheduling algorithm that integrates Multi-Agent Proximal Policy Optimization (MAPPO) with a bargaining game-based resource coordination mechanism. The proposed algorithm reduces the overall energy consumption and enhances the system performance through a bargaining-based resource allocation and matching strategy, alongside MAPPO-driven trajectory optimization. Simulation results show that our method outperforms the baseline algorithms in terms of system utility, achieving up to a 7.4% improvement.**

***Index Terms*— Mobile edge computing, uncrewed aerial vehicle (UAV), task offloading, trajectory optimization, deep reinforcement learning.**

I. Introduction

In RECENT years, the rapid advancement of cutting-edge communication technologies such as Beyond 5G (B5G) and 6G has brought transformative breakthroughs to the Internet of Things (IoT), accelerating the large-scale deployment of edge nodes and broadening the scope of their application scenarios [1]. However, the surge in computational demands has exposed the limitations of the processing power available at conventional edge nodes, posing significant challenges to scalability in massive deployments. To address this issue, Mobile Edge Computing (MEC) has become a promising solution by bringing computational resources closer to end devices, thereby partially mitigating the gap in computing capabilities [2], [3], [4].

Nonetheless, traditional fixed-position edge servers often face limitations due to high deployment costs and poor scalability, reducing their effectiveness in dynamic or time-sensitive environments. Uncrewed Aerial Vehicle-assisted Mobile Edge Computing (UMEC) systems-where MEC nodes are integrated into UAVs—have garnered increasing interest due to their flexibility, cost-efficiency, and rapid deployment capabilities [5]. These advantages make UMEC systems especially attractive within the context of the emerging low-altitude economy, which refers to the economic activities and industrial ecosystem enabled by low-altitude airspace and UAV technologies, including applications in urban infrastructure, logistics, and intelligent sensing networks.

Moreover, with the advent of Artificial General Intelligence (AGI), UMEC systems are poised to evolve into even more autonomous and intelligent edge infrastructures. The integration of AGI can enable on-board learning, dynamic resource orchestration, and situational awareness, further enhancing the responsiveness and adaptability of the UMEC system [6], [7], [8]. These trends have sparked widespread interest from both academia and industry, prompting deeper investigation into the potential of UMEC in various contemporary applications.

Although the UMEC system can effectively address the limited coverage and inflexible deployment of traditional MEC architectures, they also introduce increased system complexity and new challenges. These include task scheduling, resource allocation, trajectory optimization, and energy constraints, all of which must be carefully managed to ensure robust and efficient system operation. The existing literature has extensively studied these issues and proposed various optimization strategies. For instance, [9] proposes a joint optimization framework that coordinates task assignment, resources management, and UAV positioning, for accelerating IoT task completion and reduce UAV energy consumption. In another study, [10] introduces an iterative optimization algorithm

Received 1 July 2025; revised 25 August 2025 and 15 October 2025; accepted 28 October 2025. Date of publication 3 November 2025; date of current version 31 December 2025. This research was partially supported by the National Key Research and Development Program of China (2019YFB1802800). The associate editor coordinating the review of this article and approving it for publication was W. Yuan. *(Corresponding author: Gang Sun.)*

Zhiying Wang, Jinghui Chen, Gang Sun, and Hongfang Yu are with the Key Laboratory of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: zhiyingwang@std.uestc.edu.cn; chenjh@std.uestc.edu.cn; gangsun@uestc.edu.cn; yuhf@uestc.edu.cn).

Mohsen Guizani is with the Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, United Arab Emirates (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/TCCN.2025.3627914

that jointly considers user-to-MEC server association, power control, and UAV mobility management to minimize system latency and energy consumption, while satisfying transmission power constraints. Furthermore, [11] focuses on a Multiple-Input Single-Output (MISO) UMEC system and proposes a joint optimization strategy that minimizes overall system energy consumption by optimizing UAV beamforming vectors and central processing unit frequencies. These advancements are particularly significant in the context of the low-altitude economy, where UAV-based edge infrastructure serves as a critical enabler for scalable, responsive, and intelligent services across urban airspace. As the deployment of autonomous aerial networks becomes more prevalent, the integration of AGI is expected to further enhance the cognitive capabilities of UMEC systems [12]. AGI-enabled UMEC nodes may autonomously learn environmental patterns, predict computational loads, and self-optimize their mobility and resource orchestration strategies. This convergence of UMEC, AGI, and low-altitude economic applications paves the way for resilient, adaptive, and intelligent edge ecosystems of the future.

To address the aforementioned energy-related challenges in the context of the low-altitude economy, this paper investigates resource scheduling and intelligent UAV trajectory optimization in an energy-constrained UMEC system. The key contributions of this work are summarized as:

- We construct holistic system energy consumption and utility models in the context of the UMEC system, encompassing both edge nodes and drones. By quantitatively modeling the energy expenditure and benefits of each participant, we formulate a mathematical problem to achieve maximum system-wide utility. Our model effectively balances energy conservation and task execution efficiency, thereby laying a solid theoretical foundation for the subsequent algorithm design.
- We propose an intelligent algorithm that integrates a bargaining game mechanism with Multi-Agent Proximal Policy Optimization (MAPPO). By performing joint optimization over resource scheduling and matching strategy with UAV trajectory optimization, our algorithm achieves fair utility distribution between users and UAVs, while significantly strengthening overall system consistency and energy effectiveness.
- Extensive simulations are conducted to assess the efficiency of the proposed approach. The results demonstrate that, compared to conventional approaches, our method yields notable improvements in system stability, fairness of utility allocation, and energy consumption control, and achieves performance gains of up to 7.4% in system utility.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the system model and formulates the corresponding optimization problem. Section IV introduces our alternating optimization algorithm and analyzes its computational complexity. Section V reports the experiment results and performance evaluation. Finally, Section VI summarizes our findings and concludes the paper.

II. Related Work

A. Energy Consumption Issues in the Low-Altitude Economy

With the swift growth of the low-altitude economy, the energy consumption of UAV-assisted MEC systems has become a critical concern. Recent studies have focused on minimizing energy usage while maintaining service quality, especially in scenarios involving intensive task offloading and long-duration UAV flights. The work in [13] explored energy efficiency and task collection maximization in low-altitude economy scenarios, proposing a multi-objective reinforcement learning algorithm. This study effectively balances safety and energy efficiency by introducing a population intelligence algorithm based on Conditional Variational Autoencoder (CVAE). In [14], the authors tackled the energy minimization problem in low-altitude full-duplex systems. Because the problem is non-convex, they designed a block optimization method to decompose the optimization problem into tractable subproblems. In [15], a Rate-Splitting Multiple Access (RSMA) system was introduced to address computation capacity and power consumption challenges in low-altitude networks, significantly alleviating energy consumption pressure. Finally, [16] considered urban space limitations and efficiency requirements in low-altitude economic. They proposed a trajectory optimization framework based on Deep Reinforcement Learning (DRL) integrated with a Large Language Model to enhance safety and cost-effectiveness in UAV routing.

B. Network Resource Scheduling in UMEC Systems

Resource scheduling in UAV-assisted UMEC systems has become a key research area, with methods mainly categorized into four types: traditional optimization, heuristic algorithms, game-theoretic approaches, and learning-based methods. Traditional optimization techniques, such as convex and fractional programming, have been applied to address non-convex scheduling problems. In [17], the authors used the Alternating Direction Method of Multipliers (ADMM) in a 6G UAV-based Vehicular Edge Computing (UVEC) framework to analyze and optimize network robustness. In [18], a centralized algorithm based on Semidefinite Programming (SDP) was designed to minimize energy consumption in Software-Defined Wireless Sensor Networks (SD-WSNs), achieving global optimality. Heuristic algorithms, such as greedy algorithms, water-filling, genetic algorithms, simulated annealing, and ant colony optimization, solve problems based on experience-inspired strategies. In [19], the Ant Colony Optimization (ACO) algorithm was improved with chaotic mapping techniques to pre-plan UAV flight paths under uncertain factors in UMEC, effectively avoiding obstacles. Resource competition and cooperation has been modeled using both non-cooperative and coalition game theories. In [20], a static game model was established for cross-cell Device-to-Device (D2D) communication, and the total system rate and gain were improved through a Repeated Game (RG). Reinforcement learning (RL), especially multi-agent RL, has shown promise in dynamic and large-scale UMEC scenarios. Algorithms like Actor-Critic (AC) [21], Deep Deterministic Policy Gradient (DDPG) [22], Proximal Policy Optimization (PPO) [23], and

Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [24] address task offloading, trajectory control, and multi-objective optimization. Overall, UMEC scheduling methods are evolving toward adaptive, collaborative intelligence. Future research is expected to integrate optimization theory, game models, and deep reinforcement learning to build scalable and intelligent scheduling frameworks. In [25], the authors solved the problem of aerial target search driven by the development of the low-altitude economy. They proposed an efficient resource management method that enables effective target search and resistance to jamming attacks under limited battery life constraints.

C. Optimization of UAV Trajectory

Trajectory optimization for UAV is a core component in the UMEC system for enabling efficient task offloading and collaborative resource management. By dynamically adjusting UAV positions, coverage, and service delivery across different regions can be achieved, thereby optimizing energy consumption, reducing task latency, and balancing communication coverage with computational load. In [26], the authors considered UAV trajectory optimization on a 2D plane. They modeled the problem based on UAV coverage strategies and approximated it as a Traveling Salesman Problem (TSP) targeting coverage shrinkage optimization. UAV trajectories were derived by solving small-scale integer programming problems. Although computationally efficient, this method sacrifices trajectory control precision and is suitable for constrained or simple scenarios. In [27], the environment was discretized into a 3D grid, and UAV movements were modeled as integer-based trajectory decisions in three-dimensional space. DRL was used to solve the discrete trajectories, supporting obstacle avoidance through learning. However, control precision remained limited compared to continuous trajectory decision-making. In [28], continuous 3D trajectory decisions were explored using Successive Convex Approximation (SCA), improving spatial control accuracy but facing challenges such as high computational complexity and multivariable coupling. To enhance Quality of Service (QoS) through decreased latency and lower energy usage in UMEC systems, the study in [29] tackled the trajectory design problem for a single drone through an optimistic AC approach. In [30], the researchers independently planned the UAV's two-dimensional flight path to enhance task offloading efficiency and included a corresponding trajectory illustration in their experimental results. Meanwhile, [31] aimed at minimizing Age of Information (AoI) and energy usage through the development of both task offloading strategies and a two-dimensional trajectory design. Their simulations demonstrated the convergence of their proposed approach and visualized the movement result. Despite extensive research in the field, most existing studies on UAV trajectory optimization remain focused on hovering or two-dimensional flight, with limited exploration of flexible and dynamic three-dimensional trajectories.

In summary, existing studies on energy consumption control, resource scheduling, and UAV trajectory optimization have each made important contributions but also exhibit certain limitations. Most approaches focus on a single aspect, such as energy efficiency or trajectory design, while lacking a unified framework to balance trade-offs among multiple objectives. Moreover, scalability and coordination across multiple UMEC systems are often insufficient. To address these issues, our work proposes an integrated framework that combines bargaining game-based resource allocation with MAPPO-driven trajectory optimization, thereby achieving joint optimization of energy efficiency and system-wide performance.

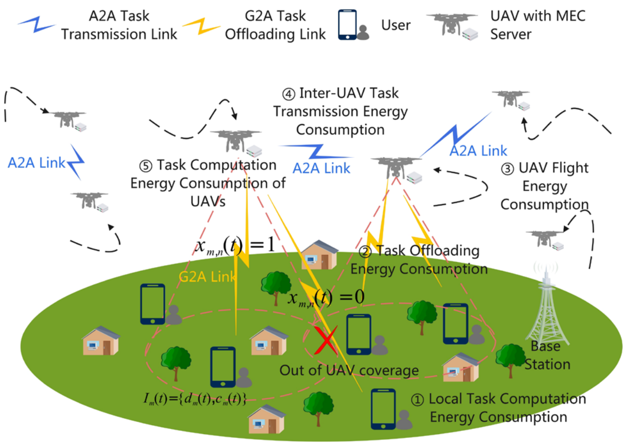

A2A Task Transmission Link
G2A Task Offloading Link
User
UAV with MEC Server
④ Inter-UAV Task Transmission Energy Consumption
A2A Link
⑤ Task Computation Energy Consumption of UAVs
A2A Link
A2A Link
③ UAV Flight Energy Consumption
$x_{m,n}(t)=1$
② Task Offloading Energy Consumption
G2A Link
$x_{m,n}(t)=0$
Out of UAV coverage
Base Station
$I_m(t)=\{d_m(t),c_m(t)\}$
① Local Task Computation Energy Consumption

Fig. 1. UAV-enabled MEC scheduling and trajectory optimization model.

III. Problem Description And Modeling

We intend to tackle the challenge of reducing energy consumption in UMEC systems with constrained resources. With the increasing application of UAV-assisted computing and communication systems in low-power networks, achieving efficient task execution and response under limited energy resources has become a critical challenge that demands urgent attention.

A. System Model

Figure 1 presents the considered task offloading scenario in a UMEC system with multiple UAVs. For long-term performance optimization, we adopt a dynamic MEC model spanning several time slots. The entire period is segmented into uniform time slots, defined as $\mathcal{T} = \{1, 2, \ldots, T\}$, each of duration δ_t. Here, $t \in \mathcal{T}$ represents the index of a specific time slot.

The system includes M user devices, represented by $\mathcal{M} = \{1, 2, \ldots, M\}$, and N UAVs as movable edge servers, represented by $\mathcal{N} = \{1, 2, \ldots, N\}$. The position of user m at time slot t is $\boldsymbol{q}_m(t) = [x_m(t), y_m(t), 0]$, while the position of UAV n is $\boldsymbol{r}_n(t) = [x_n(t), y_n(t), z_n(t)]$. Each user generates a non-divisible, computation-intensive task in every time slot, expressed as $I_m(t) = \{d_m(t), c_m(t)\}$, with $d_m(t)$ denoting the task data size in bits and $c_m(t)$ the number of cycles needed per bit.

The energy consumption model primarily consists of two components: user energy consumption and UAV energy consumption:

1) **User Energy Consumption**: Includes the energy consumed for local task computation and the energy used for uploading tasks;
2) **UAV Energy Consumption**: Includes the energy consumed for UAV flight, inter-UAV task transmission, and task computation offloaded to the UAV.

The following sections are organized to present the UAVs movement model, communication model, computation model, user energy consumption model, and problem formulation in sequence.

B. UAVs Movement Model

We define the operational space of the UAV as a cuboidal three-dimensional region bounded by the vectors $\left[0, 0, Z^{\min}\right]^T$ and $[X^{\max}, Y^{\max}, Z^{\max}]^T$, subject to the following constraints:

$$0 \leq x_n(t) \leq X^{max}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{1}$$
$$0 \leq y_n(t) \leq Y^{max}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{2}$$
$$Z^{min} \leq z_n(t) \leq Z^{max}, \forall n \in \mathcal{N}, t \in \mathcal{T}. \tag{3}$$

To ensure safe navigation, UAVs must comply with constraints on maximum velocity $v^{\max}$ and minimum allowable distance $D^{\min}$, where $D^{\min}$ represents the minimum separation required to avoid potential collisions, as expressed below:

$$\|\boldsymbol{r}_n(t+1) - \boldsymbol{r}_n(t)\| \leq v^{max}\delta_t, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{4}$$
$$\|\boldsymbol{r}_n(t) - \boldsymbol{r}_{n'}(t)\| \geq D^{min}, \forall n, n' \in \mathcal{N}, n \neq n'. \tag{5}$$

Moreover, given that UAV n has a maximum elevation angle ϕ_n, which reflects the hardware constraint of its onboard antenna or sensor, its maximum horizontal coverage radius at time t can be determined as follows [32]:

$$C_n^{max}(t) = z_n(t)\tan(\phi_n), \forall n \in \mathcal{N}, t \in \mathcal{T}. \tag{6}$$

User m can offload its task to UAV n only if it resides within the UAV's coverage region. The horizontal position of UAV n is given by $h_n(t) = [x_n(t), y_n(t), 0]^T$, leading to the following constraint:

$$x_{m,n}(t) \cdot \|\boldsymbol{h}_n(t) - \boldsymbol{q}_m(t)\| \leq C_n^{max}(t),$$
$$\forall m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}, \tag{7}$$

The binary variable $x_{m,n}(t) \in \{0, 1\}$ indicates whether task $I_m(t)$ is offloaded to UAV n. The UAV's maximum coverage radius $C_n^{\max}(t)$ which determined by the UAV's altitude $z_n(t)$ and maximum elevation angle ϕ_n. By limiting the horizontal distance between the communicating parties, offloading is ensured to occur only under reliable communication conditions.

C. Communication Model

1) Ground-to-Air (G2A): To consider obstacles like buildings and trees, we use a probabilistic LoS/NLoS channel model, commonly applied in UAV communications due to its practical balance of accuracy and simplicity [33]. The LoS probability depends on the environment and the elevation angle $\theta_{m,n}(t) = \frac{180}{\pi} \cdot \arctan\left(\frac{z_n(t)}{\|\boldsymbol{q}_m(t) - \boldsymbol{r}_n(t)\|}\right)$ between UAV n and user m at time t that can be formulated as $\mathbb{P}(LoS, \theta_{m,n}(t)) = \frac{1}{1+a\cdot\exp\left(-b(\theta_{m,n}(t)-a)\right)}$. Correspondingly, the NLoS probability can be given by $\mathbb{P}(NLoS, \theta_{m,n}(t)) = 1 - \mathbb{P}(LoS, \theta_{m,n}(t))$. The environmental constants a and b influence the probability of the LoS channel occurrence. Since a single time slot is typically short, similar to [34], the channel probability is assumed constant within each slot. Consequently, the average path loss from the ground user to the drone is given by:

$$\overline{PL}_{m,n}(t) = \mathbb{P}(LoS, \theta_{m,n}(t))\left(L_{m,n}(t) + \eta^L\right)$$
$$+ \mathbb{P}(NLoS, \theta_{m,n}(t))\left(L_{m,n}(t) + \eta^N\right), \tag{8}$$

where $L_{m,n}(t) = 20\lg\left(\frac{4\pi \cdot fr_c \cdot d_{m,n}(t)}{c}\right)$ represents the free-space path loss, fr_c is the carrier frequency, η^L is the excess path loss for the LoS channel, while η^N corresponds to the NLoS channel.

We assume that users offload tasks to UAVs via Orthogonal Frequency Division Multiple Access (OFDMA), with each user assigned a unique sub-band to avoid intra-UAV interference. Each user is associated with a single UAV at any time, and frequency reuse among UAVs is not considered, thus inter-UAV interference is neglected. This assumption, consistent with [35], focuses the analysis on trajectory and resource optimization. The expected uplink rate from user m to UAV n is given by [33]:

$$r_{m,n}(t) = B_{m,n}^{G2A}\log_2\left(1 + \frac{p_m(t) \cdot 10^{-\overline{PL}_{m,n}(t)/10}}{N_G}\right), \tag{9}$$

where $\overline{PL}_{m,n}(t)$ denotes the average path loss in decibels (dB), which is converted to linear scale using $10^{\overline{PL}_{m,n}(t)/10}$ for the purpose of data rate calculation. Here, $p_m(t)$ and N_G represent the transmit power and the noise power, respectively, and $B_{m,n}^{G2A}$ denotes the uplink bandwidth from user m to UAV n. The allocation of limited bandwidth resources should comply with the following constraints:

$$\sum_{m\in\mathcal{M}} B_{m,n}^{G2A} \leq B^{G2A}, \forall m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}, \tag{10}$$
$$B_{m,n}^{G2A} \geq 0, \forall m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}. \tag{11}$$

2) Air-to-Air (A2A): Considering that LoS connections are usually dominant in A2A links [36], the communication between UAV n and n' is modeled using the free-space path loss approach to accurately capture signal attenuation, expressed as:

$$PL_{n,n'} = 20\lg\left(\frac{\|\boldsymbol{r}_n(t) - \boldsymbol{r}_{n'}(t)\|}{1000}\right) + 20\lg(f_{r_c}) + 32.45. \tag{12}$$

Therefore, the data transfer rate can be expressed as:

$$r_{n,n'}(t) = B_{n,n'}^{A2A}\log_2\left(1 + \frac{p_n(t) \cdot 10^{-PL_{n,n'}/10}}{N_A}\right), \tag{13}$$

where $p_n(t)$ and N_A denote the transmit and noise power, respectively, and $B_{n,n'}^{A2A}$ is the bandwidth allocated from

UAV n to n'. There exists bandwidth constraints similar to (10) and (11):

$$\sum_{m\in\mathcal{M}} B^{A2A}_{n,n'} \leq B^{A2A}, \forall n, n' \in \mathcal{N}, \tag{14}$$

$$B^{A2A}_{n,n'} \geq 0, \forall n, n' \in \mathcal{N}, t \in \mathcal{T}. \tag{15}$$

D. Computation Model

Each task $I_m(t)$ is processed either locally or via edge computation. Since there are N UAVs available for offloading, the edge mode offers $(N+1)$ possible offloading options. We introduce a extend set $\mathcal{N}^\dagger = \{0\} \cup \mathcal{N}$. Then, the offloading decisions $\{x_{m,n}(t), \forall m \in \mathcal{M}, n \in \mathcal{N}^\dagger\}$ form a binary matrix of size $M \times (N+1)$.

1) Local Mode When $x_{m,0}(t) = 1$: Task $I_m(t)$ is processed on the user's device itself, resulting in zero transmission delay. The total delay is thus determined solely by the local execution time, given by $T^{\text{loc}}_m(t) = \frac{d_m(t)\cdot c_m(t)}{f_m(t)}$, where $f_m(t)$ denotes the local CPU frequency of user m.

2) Edge Mode When $x_{m,n'}(t) = 1$: Task $I_m(t)$ is offloaded to UAV n' for processing. Because UAVs have a limited coverage area, it must be verified whether user m is located inside the coverage zone of UAV n'. If user m is covered, the task is transmitted directly over the G2A link. Otherwise, the task is first sent to UAV n, which then relays it to UAV n' via the A2A link. Assuming a full-duplex communication framework allowing UAVs to receive and forward data simultaneously, the transmission delay is calculated as $T^{off}_{m,n'}(t) = \max\left\{\frac{d_m(t)}{r_{m,n}(t)}, \frac{d_m(t)}{r_{n,n'}(t)}\right\}$. Since UAVs operate over LoS channels with minimal interference and higher transmit power than users [33], we have $r_{m,n}(t) \ll r_{n,n'}(t)$, and the transmission delay simplifies to $T^{off}_{m,n'}(t) = \frac{d_m(t)}{r_{m,n}(t)}$.

Once task $I_m(t)$ is offloaded to UAV n', it is executed at the edge. Let $f_{m,n'}(t)$ denote the CPU timeslice allocated to the task, then the execution time on UAV n' is given by $T^{\text{exe}}_{m,n'}(t) = \frac{d_m(t)\cdot c_m(t)}{f_{m,n'}(t)}$.

In summary, the total delay for task $I_m(t)$ in the edge mode is formulated as:

$$T^{edge}_{m,n'}(t) = T^{off}_{m,n'}(t) + T^{exe}_{m,n'}(t). \tag{16}$$

Additionally, the allocation of limited computing resources should be subject to the following constraints:

$$\sum_{m\in\mathcal{M}} f_{m,n}(t) \leq f_n, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{17}$$

$$f_{m,n}(t) \geq 0, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{18}$$

where f_n indicates the maximum processing capability of UAV n in terms of CPU frequency.

E. User Energy Consumption Model

1) Local Task Computation Energy: Local computation energy follows the dynamic power model of Complementary Metal-Oxide-Semiconductor (CMOS) circuits, where power scales with the cube of the CPU frequency. Thus, the energy consumption is proportional to both the computation time and the cube of the frequency. The local computation energy for user m at time slot t is given by:

$$E^{local}_m(t) = \varphi_m (f_m(t))^3 T^{loc}_m(t), \tag{19}$$

where φ_m is the effective capacitance coefficient specific to user m's device hardware.

2) Task Offloading Energy: The energy consumed for task offloading depends on the user's transmit power $p^{trans}_m(t)$ and the transmission delay $T^{off}_{m,n}(t)$. Specifically, the energy for user m to offload the task to UAV n is given by:

$$E^{off}_{m,n}(t) = p^{trans}_m(t) \cdot T^{off}_{m,n}(t). \tag{20}$$

3) Total User Energy Consumption: The total energy consumption of user m consists of either the offloading energy or the local computation energy, depending on the selected computation mode. Specifically, the total energy consumption is given by:

$$E^{user}_m(t) = x_{m,n}(t) \cdot E^{off}_{m,n}(t) + x_{m,0}(t) \cdot E^{local}_m(t). \tag{21}$$

F. UAV Energy Consumption Model

1) UAV Flight Energy: The propulsion power of a UAV during flight is given by [11]:

$$\begin{aligned} p^{fly}_n(t) =& \frac{1}{2} d_0 \rho g S_0 (v_n(t))^3 + P_{blade}\left(1 + \frac{3(v_n(t))^2}{V^2_{tip}}\right) \\ &+ P_{hover}\left(\sqrt{1 + \frac{(v_n(t))^4}{4v_0^4}} - \frac{(v_n(t))^2}{2v_0^2}\right)^{\frac{1}{2}}, \end{aligned} \tag{22}$$

where d_0 denotes the UAV drag ratio, $v_n(t)$ denotes the velocity of UAV n (m/s), ρ is the air density (kg/m^3), and g is the gravitational acceleration (m/s^2). P_{blade} is the blade profile power, V_{tip} is the blade tip speed, P_{hover} denotes the hovering induced power, and v_0 represents the average induced velocity of the rotor. The flight energy of UAV n is therefore:

$$E^{fly}_n(t) = p^{fly}_n(t) \cdot \delta_t. \tag{23}$$

2) Inter-UAV Task Transmission Energy: When UAV n chooses to transmit the task to UAV n' for execution, the transmission time is:

$$T_{n,n'}(t) = \frac{d_m(t)}{r_{n,n'}(t)}. \tag{24}$$

The corresponding transmission energy consumption is:

$$E^{trans}_{n,n'}(t) = p^{trans}_n(t) \cdot T_{n,n'}(t), \tag{25}$$

where $p^{\text{trans}}_n(t)$ denotes the transmission power of UAV n, which is assumed to be a fixed constant in this work, consistent with typical UAV communication hardware settings.

3) Task Computation Energy of UAVs: The energy consumed by UAV n to execute task $I_m(t)$ depends on its computation frequency $f_{m,n}(t)$ and execution time $T^{exe}_{m,n}(t)$, following the same dynamic power model as in local computation. The corresponding computation energy consumption is:

$$E^{exe}_{m,n}(t) = \varphi_n (f_{m,n}(t))^3 T^{exe}_{m,n}(t), \tag{26}$$

where φ_n is the effective switching capacitance of UAV n.

4) Total UAV Energy Consumption: The total energy consumption of UAV n at time t consists of three parts: propulsion energy for flying, transmission energy for forwarding tasks to other UAVs, and computation energy for executing user tasks. It is expressed as:

$$E_n^{UAVs}(t) = E_n^{fly}(t) + \sum_{n' \in \mathcal{N}} E_{n,n'}^{trans}(t) + \sum_{m \in \mathcal{M}} E_{m,n}^{exe}(t). \tag{27}$$

G. Problem Formulation

1) User Utility: The utility of UAV n for user m is defined as the user's Quality of Experience (QoE), which is similar to [37] and consists of two components: task completion satisfaction and energy consumption:

$$U_m^n(t) = \omega_m \cdot \frac{\log\left(1 + T_m^{max} - T_{m,n}^{user}(t)\right)}{\log\left(1 + T_m^{max}\right)} - (1 - \omega_m) \cdot \frac{E_m^{user}(t)}{E_m^{max}(t)}. \tag{28}$$

Due to the diminishing marginal effect of task completion, where user satisfaction or perceived value increases at a decreasing rate as task completion improves, a logarithmic function is adopted to model the satisfaction. Both satisfaction and energy consumption are normalized. ω_m is a weight factor representing the importance of task satisfaction versus energy consumption. It is set as a fixed parameter to reflect different user service sensitivity types, where a larger value indicates delay-sensitive users and a smaller value corresponds to energy-sensitive users. T_m^{max} is the maximum task completion time constraint, and $T_{m,n}^{user}(t) = x_{m,n}(t) \cdot T_{m,n}^{edge}(t) + x_{m,0}(t) \cdot T_m^{loc}(t)$ denotes the actual task completion time. $E_m^{max}(t)$ is the maximum energy budget for user m.

2) UAV Utility: The utility of user m for UAV n represents the revenue of the MEC server and consists of the service fee collected from the user and the energy consumed by the UAV:

$$U_n^m(t) = \omega_n \cdot \frac{f_{m,n}(t)p_n}{f_n^{max}p_n^{max}} - (1-\omega_n) \cdot \frac{E_n^{fly}(t) + E_{n,n'}^{trans}(t) + E_n^{exe}(t)}{E_n^{max}(t)}, \tag{29}$$

where, ω_n is a weight factor representing the trade-off between service revenue and energy cost. p_n is the unit price paid by the user for computation resources, p_n^{max} is the maximum allowable unit price, and $E_n^{max}(t)$ is the maximum energy constraint of UAV n.

3) Total System Utility: The total system utility comprises both user utility and UAV utility:

$$U(t) = \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \left[\omega^{ut} U_m^n(t) + (1 - \omega^{ut}) U_n^m(t)\right], \tag{30}$$

where, $\omega^{ut} \in [0, 1]$ is the utility allocation weight, which adjusts the relative importance of user utility and UAV utility in the total system utility. When ω^{ut} is close to 1, the system prioritizes user utility; when it is close to 0, it emphasizes UAV utility.

4) Optimization Objective: We aim to minimize the energy consumption of the UMEC system while ensuring the required service quality. To this end, we formulate an objective function that maximizes the overall system utility $U(t)$. This is achieved by jointly optimizing the offloading decisions $\boldsymbol{X} = \{x_{m,n}(t), \forall m \in \mathcal{M}, n \in \mathcal{N}^\dagger, t \in \mathcal{T}\}$, UAV trajectories $\boldsymbol{R} = \{\boldsymbol{r}_n(t), \forall n \in \mathcal{N}, t \in \mathcal{T}\}$, resource allocation strategies $\boldsymbol{B} = \{B_{m,n}^{G2A}(t), \forall m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}\}$, $\boldsymbol{F} = \{f_{m,n}(t), \forall m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}\}$.

The energy-aware overall system utility optimization problem can be formulated as follows:

$$\max_{\boldsymbol{X},\boldsymbol{R},\boldsymbol{B},\boldsymbol{F}} \sum_{t \in \mathcal{T}} U(t) \tag{31}$$

$$\text{s.t. } (1) - (5), (7), (10), (11), (14), (15), (17), (18)$$

$$x_{m,n}(t) \in \{0, 1\}, \forall m \in \mathcal{M}, n \in \mathcal{N}^\dagger, t \in \mathcal{T} \tag{32}$$

$$\sum_{n=0}^{N+1} x_{m,n}(t) = 1, \forall m \in \mathcal{M}, n \in \mathcal{N}^\dagger, t \in \mathcal{T} \tag{33}$$

$$T_{m,n}^{user}(t) < T_m^{max}, \forall m \in \mathcal{M} \tag{34}$$

$$E_m^{user}(t) < E_m^{max}(t), \forall m \in \mathcal{M} \tag{35}$$

$$E_n^{UAVs}(t) < E_n^{max}(t), \forall n \in \mathcal{N}, \tag{36}$$

where constraints (1)-(5) define the UAVs' location and movement limitations, (7) ensures that tasks can only be offloaded to UAVs within the user's coverage area, (10) caps the maximum G2A transmission bandwidth, (11) guarantees that the allocated G2A bandwidth is strictly positive. Similar resource allocation constraints also exist (14), (15), (17), (18). Constraint (32) and (33) enforces a task indivisibility. Constraint (34) ensures that task completion meets time deadlines, (35) enforces user energy limits, and (36) limits UAV energy usage.

Since total system utility is influenced by multiple users and UAVs, it forms a multi-agent optimization problem. Moreover, the utility considered here spans over time, making it a long-term optimization problem rather than an instantaneous one. In the following sections, we develop an effective approach that integrates DRL and game theory to optimize long-term system utility, while ensuring efficient and coordinated resource scheduling among all participants.

IV. Algorithm Design

Based on the aforementioned system model and optimization problem formulation, we design an energy-efficient resource scheduling algorithm for the UMEC system, named UMEC Offloading and Utility-aware Resource Strategy (UOURS). It's mainly includes two parts: (1) resource scheduling and matching strategy design, and (2) flight trajectory optimization and optimization.

First, a bargaining-based resource scheduling and user-UAV matching strategy is designed. In this phase, UAV locations are fixed. Through a bargaining mechanism, resource allocation is optimized to allow UAVs to efficiently assign computing tasks and enhance overall system performance by matching users with appropriate UAVs. Subsequently, fixing the resource allocation and matching strategy, the algorithm proceeds to the flight trajectory optimization phase, which is based on MAPPO. In this part, each UAV is treated as an agent that leverages MAPPO to plan and optimize its trajectory. Specifically, using a policy optimization method based on unsupervised learning, agents adapt their flight paths during interaction with the environment to maximize total system utility.

A. Resource Scheduling and Matching Strategy

At the beginning of each time slot, the algorithm calculates the resource scheduling and matching strategy based on the current positions of the UAVs.

1) Resource Scheduling: As described earlier, users purchase computing resources from UAVs to offload tasks at a certain cost, while UAVs earn revenue by providing computing resources $\boldsymbol{F}$ and communication resources $\boldsymbol{B}$, at the expense of energy consumption. This process can be seen as a market-like transaction: users act as buyers and UAVs act as sellers. Thus, we adopt a bargaining mechanism to facilitate negotiation between users and UAVs regarding the allocation of on-demand computing and communication resources.

To analyze the curvature of $U(t)$ with respect to bandwidth allocation $B_{m,n}(t)$, we compute the second derivative:

$$\begin{aligned}\frac{\partial^2 U(t)}{\partial^2 B^*_{m,n}(t)} &= (1-\omega_m)\cdot\frac{c_m(t)}{E_m^{max}(t)}\cdot\left(-\frac{2}{B_{m,n}(t)^3}\right)\\ &+\omega_m\cdot\frac{c_m(t)\cdot\left(-\frac{1}{u^2 B_{m,n}(t)^2}\cdot\frac{c_m(t)}{B_{m,n}(t)^2}-\frac{2}{uB_{m,n}(t)^3}\right)}{\log_2\left(1+T_m^{max}\right)\cdot\ln 2},\end{aligned} \tag{37}$$

where $u = T_m^{max} - \frac{c_m(t)}{B_{m,n}(t)} - \frac{d_m(t)}{f_{m,n}(t)^2}$ and $c_m(t) = \frac{d_m(t)}{\log_2\left(1+\frac{p_m^{trans}(t)}{PL_{m,n}(t)N_G}\right)}$. Since the second derivative is negative, $U(t)$ is a concave function of $B_{m,n}(t)$, implying the existence of a maximum. Given fixed computation resource $f_{m,n}(t)$, the optimal bandwidth allocation $B^*_{m,n}(t)$, where the first derivative is zero, is given by:

$$B^*_{m,n}(t) = \frac{c_m(t)}{1+T_m^{max}-\frac{c_m^2(t)}{f_{m,n}(t)-\frac{\omega_m E_m^{max}(t)}{A}}}, \tag{38}$$

where $A = \log(1+T_m^{max})\ln 2(\omega_m - 1)c_m(t)$.

To ensure feasibility, both user and UAV utilities must be positive, i.e., $U_m^n(t) > 0$ and $U_n^m(t) > 0$. Solving these inequalities yields the minimum and maximum bounds of computation resources $f_{m,n}(t)$:

$$\begin{aligned}&f_{m,n}^{min}(t) =\\ &\frac{\omega_m d_m(t)c_m(t)}{\omega_m(T_m^{max})-\omega_m T_{m,n}^{off}(t)-(1-\omega_m)\cdot\frac{E_{m,n}^{off}(t)}{E_m^{max}(t)}\cdot(T_m^{max})},\end{aligned} \tag{39}$$

$$f_{m,n}^{max}(t) = \frac{\omega_n p_n E_n^{max}(t)-\sqrt{(\omega_n p_n E_n^{max}(t))^2-4AC}}{2A}, \tag{40}$$

where $A = (1-\omega_n)f_n^{max}p^{max}\varphi_n d_m(t)c_m(t)$ and $C = \left(E_n^{\text{fly}}(t)+E_{n,n'}^{\text{trans}}(t)\right)$.

The size of the computable resource interval is then $\Delta f_{m,n}(t) = f_{m,n}^{max}(t) - f_{m,n}^{min}(t)$. The resource transaction between user m and UAV n during the interval $\Delta f_{m,n}(t)$ is formulated as a bargaining game [38]. Clearly, both players prefer to reach an agreement early, as their utilities are discounted over time. A discount factor is introduced to describe this time sensitivity. A smaller discount factor implies lower patience and higher urgency.

The discount factors for user m and UAV n are defined as:

$$\epsilon_m(t) = 1 - \frac{T_{m,n}^{edge}(t)}{T_m^{max}(t)}, \tag{41}$$

$$\epsilon_n(t) = 1 - \frac{T_{m,n}^{exe}(t)}{T_m^{max}(t)}. \tag{42}$$

A longer task upload time results in lower $\epsilon_m(t)$ for the user, indicating greater impatience. Similarly, longer execution time yields lower $\epsilon_n(t)$ for the UAV. Additionally, a larger deadline $T_m^{max}(t)$ indicates greater tolerance for delay, resulting in higher discount factors for both.

The bargaining game yields a unique optimal allocation ratio that reflects a fair distribution of resources between user m and UAV n. Assuming the negotiation unfolds over a maximum of T_b discrete rounds, with user m initiating the first offer, the final allocation is determined through backward induction. This method solves the game from the final round back to the beginning, ensuring rational decision-making at each stage. Two possible scenarios are considered depending on which party makes the last proposal in round T_b: (1) if user m makes the final offer, they gain an advantage in securing a more favorable outcome; (2) if UAV n makes the final offer, the bargaining power shifts in its favor. In both cases, backward induction leads to a stable and efficient allocation strategy acceptable to both sides.

When user m proposes an allocation ratio of $(1, 0)$ in round T_b, UAV n will make a proposal in round $T_b - 1$. To prevent user m from rejecting this proposal, UAV n offers an allocation ratio of $(\epsilon_m, 1-\epsilon_m)$ in round $T_b - 1$. In this case, UAV n implies: if you reject this, you will only get ϵ_m in the next round, making the user rationally accept the current offer. Here, the computational resource allocated to user m by UAV n equals what the user would receive in the next round upon rejection. Thus, UAV n maximizes its own remaining computational resource as $1-\epsilon_m$. Similarly, in round $T_b - 2$, user m offers $(1-\epsilon_n(1-\epsilon_m), \epsilon_n(1-\epsilon_m))$. The remaining rounds follow the same pattern.

When T_b is even, the optimal resource share for user m in the first round is:

$$\begin{aligned}r_m^{m*}(t) =& \epsilon_m(t)\left(1-\epsilon_n(t)\left(1-\epsilon_m(t)(\cdots)\right)\right)\\ =& \epsilon_m(t)-\epsilon_m(t)\epsilon_n(t)+\epsilon_m(t)^2\epsilon_n(t)\\ &-\epsilon_m(t)^2\epsilon_n(t)^2-\cdots-\left(\epsilon_m(t)\right)^{\frac{T_b}{2}-1}\left(\epsilon_n(t)\right)^{\frac{T_b}{2}-1}\\ &+\epsilon_m(t)^{\frac{T_b}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b}{2}-1}\end{aligned}$$

$$\begin{aligned}&=\sum_{n=0}^{\frac{T_b}{2}-2}\left[\epsilon_m(t)\left(1-\epsilon_m(t)\right)\left(\epsilon_m(t)\epsilon_n(t)\right)^n\right]\\&\quad+\epsilon_m(t)^{\frac{T_b}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b}{2}-1}\\&=\frac{\epsilon_m(t)\left(1-\epsilon_m(t)\right)\left(1-\left(\epsilon_m(t)\epsilon_n(t)\right)^{\frac{T_b}{2}-1}\right)}{1-\epsilon_m(t)\epsilon_n(t)}\\&\quad+\epsilon_m(t)^{\frac{T_b}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b}{2}-1}.\end{aligned} \tag{43}$$

When T_b is odd, the optimal allocation share for user m when UAV n makes the first proposal is:

$$\begin{aligned}r_m^{m*}(t)=&1-\epsilon_n(t)\left(1-\epsilon_m(t)(\cdots)\right)\\=&1-\epsilon_n(t)+\epsilon_m(t)\epsilon_n(t)-\epsilon_m(t)\epsilon_n(t)^2\\&+\epsilon_m(t)^2\epsilon_n(t)^2-\cdots-\left(\epsilon_m(t)\right)^{\frac{T_b-3}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b-1}{2}}\\&+\epsilon_m(t)^{\frac{T_b-1}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b-1}{2}}\\=&\sum_{n=0}^{\frac{T_b-3}{2}}\left[\left(1-\epsilon_m(t)\right)\left(\epsilon_m(t)\epsilon_n(t)\right)^n\right]\\&+\epsilon_m(t)^{\frac{T_b-1}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b-1}{2}}\\=&\frac{\left(1-\epsilon_m(t)\right)\left(1-\left(\epsilon_m(t)\epsilon_n(t)\right)^{\frac{T_b-1}{2}}\right)}{1-\epsilon_m(t)\epsilon_n(t)}\\&+\epsilon_m(t)^{\frac{T_b-1}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b-1}{2}}.\end{aligned} \tag{44}$$

Therefore, when user m makes the proposal in round T_b, the optimal allocation ratio for user m in the first round is:

$$r_m^{m*}(t)=\begin{cases}\dfrac{\epsilon_m(t)\left(1-\epsilon_m(t)\right)\left(1-\left(\epsilon_m(t)\epsilon_n(t)\right)^{\frac{T_b}{2}-1}\right)}{1-\epsilon_m(t)\epsilon_n(t)}\\+\left(\epsilon_m(t)\right)^{\frac{T_b}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b}{2}-1},\\\text{when } T_b \text{ is even,}\\\dfrac{\left(1-\epsilon_m(t)\right)\left(1-\left(\epsilon_m(t)\epsilon_n(t)\right)^{\frac{T_b-1}{2}}\right)}{1-\epsilon_m(t)\epsilon_n(t)}\\+\left(\epsilon_m(t)\right)^{\frac{T_b-1}{2}}\left(\epsilon_n(t)\right)^{\frac{T_b-1}{2}},\\\text{when } T_b \text{ is odd.}\end{cases} \tag{45}$$

When UAV n makes a proposal of $(0,1)$ in round T_b, the derivation is similar and omitted due to space limitations. The optimal allocation ratio for user m in the first round is:

$$r_m^{n*}(t)=\begin{cases}\dfrac{\left(1-\epsilon_n(t)\right)\left(1-\left(\epsilon_m(t)\epsilon_n(t)\right)^{\frac{T_b}{2}}\right)}{1-\epsilon_m(t)\epsilon_n(t)},\\\text{when } T_b \text{ is even,}\\\dfrac{\epsilon_m(t)\left(1-\epsilon_n(t)\right)\left(1-\epsilon_m(t)^{\frac{T_b+1}{2}}\epsilon_n(t)^{\frac{T_b-1}{2}}\right)}{1-\epsilon_m(t)\epsilon_n(t)},\\\text{when } T_b \text{ is odd.}\end{cases} \tag{46}$$

Based on the optimal allocation ratio for user m, the corresponding allocation for UAV n is:

$$\begin{cases}r_n^{m*}(t)=1-r_m^{m*}(t),\\\text{when user } m \text{ makes the proposal in round } T_b,\\r_n^{n*}(t)=1-r_m^{n*}(t),\\\text{when UAV } n \text{ makes the proposal in round } T_b.\end{cases} \tag{47}$$

Accordingly, when user m initiates the proposal, the optimal computational resource requested by user m is:

$$f_m^{user}(t)=f_n^{max}(t)-r_m^{m*}(t)\Delta f_{m,n}(t). \tag{48}$$

The optimal amount of computational resource offered by UAV n is:

$$f_n^{uav}(t)=f_n^{min}(t)+r_n^{m*}(t)\Delta f_{m,n}(t). \tag{49}$$

Since $\Delta f_{m,n}(t)=f_{m,n}^{max}(t)-f_{m,n}^{min}(t)$ and $r_n^{m*}(t)=1-r_m^{m*}(t)$, we have $f_m^{user}(t)=f_n^{uav}(t)$. Therefore, the two parties reach an agreement on the allocated computational resource:

$$f_{m,n}^*(t)=f_m^{user}(t)=f_n^{uav}(t)=f_n^{max}(t)-r_m^{m*}(t)\Delta f_{m,n}(t).$$

Similarly, if UAV n makes the proposal, the optimal allocation is:

$$f_{m,n}^*(t)=f_n^{max}(t)-r_m^{n*}(t)\Delta f_{m,n}(t).$$

In conclusion, the optimal bandwidth and computational resource allocations are:

$$B_{m,n}^*(t)=\frac{c_m(t)}{1+T_m^{max}-\frac{c_m^2(t)}{f_{m,n}(t)-\frac{\omega_m E_m^{max}(t)}{A}}}, \tag{50}$$

$$f_{m,n}^*(t)=\begin{cases}f_n^{max}(t)-r_m^{m*}(t)\Delta f_{m,n}(t),\\\text{if user } m \text{ makes the proposal } (a),\\f_n^{max}(t)-r_m^{n*}(t)\Delta f_{m,n}(t),\\\text{if UAV } n \text{ makes the proposal } (b).\end{cases} \tag{51}$$

Based on the optimal bandwidth allocation strategy $\boldsymbol{B^*}=\{B_{m,n}^*(t),\forall m\in\mathcal{M},n\in\mathcal{N},t\in\mathcal{T}\}$ and computational resource allocation strategy $\boldsymbol{F^*}=\{f_{m,n}^*(t),\forall m\in\mathcal{M},n\in\mathcal{N},t\in\mathcal{T}\}$, the bargaining process between users and UAVs over system utility is defined as Algorithm 1.

2) Matching Strategy: Based on the optimal resource allocation strategy, we design a user-UAV matching strategy using multi-party preference profiles to efficiently handle the complexity between task demand and resource supply. This enables users and UAVs to reach mutually beneficial computation offloading decisions that satisfy QoE and improve resource utilization. Let the set of users who have not yet offloaded their tasks at time slot t be denoted as $\mathcal{M}_{req}(t)=\{\mathcal{M}_m(t)\mid m\in\mathcal{M},t\in\mathcal{T}\}$. A many-to-one matching mechanism is used to determine the offloading strategy for all tasks of edge nodes.

We define the current matching as a triplet $(\mathcal{P}(t),\Theta(t),\Pi(t))$:

- $\mathcal{P}(t)=(\mathcal{M}_{req}(t),\mathcal{N})$ denotes the set of pending offloading tasks and the set of UMEC servers.

Algorithm 1 Bargaining-Based Resource Scheduling Strategy

```
Input: User $m$, UAV $n$.
Output: Optimal bandwidth resource $B^*_{m,n}(t)$ between user $m$ and UAV $n$;
        Optimal computational resource allocation strategy $f^*_{m,n}(t)$.
1  Initialize $U^n_m(t) = 0$, $U^m_n(t) = 0$, $f^*_{m,n}(t) = f^{max}_n(t)$;
2  Initialize bandwidth resource $B^*_{m,n}(t)$ using Equation (50) and $f^*_{m,n}(t)$;
3  if User $m$ makes the proposal then
4      Compute $f^{user*}_{m,n}(t)$ using Equation (51-a);
5      Calculate user utility $U^n_m(t)$ and UAV utility $U^m_n(t)$ via Equations (28) and (29);
6      Compute total utility $U^{user} = U^n_m(t) + U^m_n(t)$;
7  end
8  if UAV $n$ makes the proposal then
9      Compute $f^{uav*}_{m,n}(t)$ using Equation (51-b);
10     Calculate user utility $U^n_m(t)$ and UAV utility $U^m_n(t)$ via Equations (28) and (29);
11     Compute total utility $U^{uav} = U^n_m(t) + U^m_n(t)$;
12 end
13 if $U^{user} > U^{uav}$ then
14     $f^*_{m,n}(t) = f^{user*}_{m,n}(t)$;
15 end
16 else
17     $f^*_{m,n}(t) = f^{uav*}_{m,n}(t)$;
18 end
19 Compute the optimal bandwidth resource $B^*_{m,n}(t)$ using Equation (50).
```

- $\Theta(t) = (\Theta_m(t), \Theta_n(t))$ includes the preference lists of both tasks and UMEC servers. Each task $\mathcal{M}_m(t) \in \mathcal{M}_{req}(t)$ has a descending-ordered preference list over UMEC servers: $\Theta_m(t) = \{n \mid n \in \mathcal{N}, n \succ_{\mathcal{M}_m(t)} n'\}$, where $\succ_{\mathcal{M}_m(t)}$ indicates that task $\mathcal{M}_m(t)$ prefers n over n'. Similarly, each MEC server $n \in \mathcal{N}$ has a descending-ordered preference list over tasks: $\Theta_n = \{\mathcal{M}_m(t) \in \mathcal{M}_{req}(t), \mathcal{M}_m(t) \succ_n \mathcal{M}_{m'}(t)\}$, where $\succ_n$ indicates that UMEC server n prefers task $\mathcal{M}_m(t)$ over $\mathcal{M}_{m'}(t)$.
- $\Pi(t) \subseteq \mathcal{M}_{req}(t) \times \mathcal{N}$ is the many-to-one matching result between tasks and UMEC servers. Each task $\mathcal{M}_m(t) \in \mathcal{M}_{req}(t)$ can be matched with at most one UMEC server, i.e., $\Pi_m(t) \in \mathcal{N}$, while each MEC server $n \in \mathcal{N}$ may serve multiple tasks, i.e., $\Pi_n(t) \subseteq \mathcal{M}_{req}(t)$.

3) Construction of the Preference List: The construction process of the preference list $\Theta(t)$ is as follows:

- **Predict Resource Allocation:** Based on the resource scheduling strategy in Algorithm 1, predict the optimal bandwidth resource $B^*_{m,n}(t)$ and optimal computational resource allocation $f^*_{m,n}(t)$ that UAV n would allocate to user m.
- **Calculate Task Preferences for UMEC Servers:** For each user $m \in \mathcal{M}_{req}(t)$, compute its preference value toward each UMEC server $n \in \mathcal{N}$ as $PV_m(n) = U^n_m(t)$, where $U^n_m(t)$ is the utility gained by user m when offloading the task to UAV n.
- **Construct User Preference Lists:** Sort the UMEC servers in descending order based on $PV_m(n)$ to form user m's preference list $\Theta_m(t)$.
- **Calculate UMEC Server Preferences for Tasks:** For each UMEC server $n \in \mathcal{N}$, compute its preference value toward each task $m \in \mathcal{M}_{req}(t)$ as $PV_n(m) = U^m_n(t)$, where $U^m_n(t)$ is the utility of UAV n when processing task m.
- **Construct UMEC Server Preference Lists:** Sort tasks in descending order based on $PV_n(m)$ to form the UMEC server n's preference list $\Theta_n(t)$.

4) Matching Construction Process: The matching process is as follows:

- **Initialize Rejection Set:** All tasks are initially placed into the rejection set $\mathcal{M}_{rej}(t) = \mathcal{M}_{req}(t)$.
- **Select Most Preferred Server:** For each task $m \in \mathcal{M}_{rej}$, select the top-ranked server $n = \Theta_m(t)[0]$ from its preference list and temporarily add it to the task's matching list $\Pi_m(t)$.
- **Update Server Matching List:** If task m prefers server n, temporarily add m to server n's matching list $\Pi_n(t)$.
- **Update Matching List Based on Resource Constraints:** Server n updates its matching list $\Pi_n(t)$ by keeping the top b_n preferred tasks, where b_n is the number of currently available CPU cores. The update follows these rules:
  - If the number of tasks in $\Pi_n(t)$ does not exceed b_n, and the total allocated computational resources do not exceed f^{max}_n, then keep all matched tasks.
  - Otherwise, remove the less preferred tasks and place them into a removal set D_n.
- **Add Removed Tasks to Rejection Set:** Add all removed tasks D_n into the rejection set $\mathcal{M}_{rej}(t)$.
- **Update Preferences and Matchings for Removed Tasks:** For each task $m \in D_n$, remove server n from its preference list $\Theta_m(t)$ and update its matching list $\Pi_m(t)$ accordingly.
- **Repeat Matching Process:** Repeat the above steps for tasks in $\mathcal{M}_{rej}(t)$ until all tasks are matched with a server, or have been rejected by all servers, in which case the task is executed locally.

Through the above matching mechanism, each task can select the most suitable offloading target according to its own preference and the UMEC servers' preferences under resource constraints. The final matching result is denoted by $\Pi^*(t)$, which is then transformed into the offloading decision $\boldsymbol{X}^* = \{x^*_{m,n}(t), \forall m \in \mathcal{M}, n \in \mathcal{N}^\dagger, t \in \mathcal{T}\}$, based on the following rule:

$$x^*_{m,n}(t) = \begin{cases} 1 & \text{if } \Pi_m(t) = \emptyset \text{ and } n = 0, \\ 1 & \text{if } \Pi_m(t) = n \text{ and } n \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{52}$$

Thus, the optimal offloading decision $\boldsymbol{X}^*$ is obtained. Algorithm 2 outlines the key steps involved in the matching process.

Algorithm 2 Many-to-One Matching Mechanism

```
Input: Pending task set $\mathcal{M}_{req}(t)$, UMEC server set $\mathcal{N}$.
Output: Optimal offloading decision $\boldsymbol{X}^*$, resource allocation $\boldsymbol{B}^*$ and $\boldsymbol{F}^*$.
1  Initialization: $\mathcal{M}_{rej}(t) = \mathcal{M}_{req}(t)$, $\Pi(t) = \emptyset$
2  for each task $m \in \mathcal{M}_{req}(t)$ do
3      for each UMEC server $n \in \mathcal{N}$ do
4          Predict resource allocation $B^*_{m,n}(t)$ and $f^*_{m,n}(t)$ based on Algorithm 1;
5          Compute the task's preference value toward UMEC server $n$: $PV_m(n) = U^n_m(t)$;
6          Compute UMEC server $n$'s preference value toward task $m$: $PV_n(m) = U^m_n(t)$;
7          $PV_m(n) > PV_m(n') \iff n \succ_{\mathcal{M}_m(t)} n'$, thus $\Theta_m(t) = \{n, n'\}$;
8          $PV_n(m) > PV_n(m') \iff m \succ_n m'$, thus $\Theta_n(t) = \{m, m'\}$;
9      end
10 end
11 while there exists $\mathcal{M}_m(t) \in \mathcal{M}_{rej}(t)$ such that $\Theta_m(t) \neq \emptyset$ and $\mathcal{M}_m(t) \notin \Theta_n(t)$ do
12     for each $m \in \mathcal{M}_{rej}(t)$ do
13         Select the most preferred UMEC server $n = \Theta_m(t)[0]$;
14         if $PV_n(m) > 0$ then
15             Temporarily add task $m$ to UMEC server $n$'s matching list $\Pi_n(t)$;
16         end
17     end
18     for each UMEC server $n \in \mathcal{N}$ that received new tasks do
19         if number of tasks in $\Pi_n(t)$ exceeds $b_n$ or total resources exceed $f_n^{max}$ then
20             Keep the top $b_n$ preferred tasks and remove the rest $D_n$;
21             Add removed tasks $D_n$ to the rejection set $\mathcal{M}_{rej}(t)$;
22             for each task $m \in D_n$ do
23                 Update task $m$'s preference list $\Theta_m(t)$ by removing server $n$;
24                 Update task $m$'s matching list $\Pi_m(t)$;
25             end
26         end
27     end
28 end
29 Convert the final optimal matching list $\Pi^*(t)$ into the optimal offloading decision $\boldsymbol{X}^*$ based on Equation (52).
```

B. Flight Trajectory Optimization

Given known resource allocation and matching strategies, the next step is to compute the optimal flight trajectory. This trajectory optimization is first modeled as a Partially Observable Markov Decision Process (POMDP), and then addressed using the MAPPO framework.

1) POMDP Formulation: In a UMEC system, each UAV is modeled as an agent with limited sensing capabilities and only local observation of the dynamic environment. Therefore, the problem is characterized as a POMDP, defined by the tuple $< \mathcal{U}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$, where $\gamma \in [0, 1)$ represents the discount factor. The other parameters are detailed below:

a) Multiple agents: : Each UAV, represented as an agent in the set $\mathcal{U} = \{1, 2, \ldots, U\}$, seeks to maximize the overall system utility by training an optimal flight path scheme based on the current state.

b) State and observation space: : The state $\mathcal{S}$ of the UMEC system is constructed from the individual observations $\mathcal{O}$ made by each UAV agent. As UAVs have access only to their own local environments, the observation for UAV n is defined as $o_n = \boldsymbol{r}_n(t) = [x_n(t), y_n(t), z_n(t)]^T \in \mathcal{O}$, representing its current 3D position. Consequently, the global state space is given by $\mathcal{S} = \{o_1, \ldots, o_N\} = \Big\{[x_1(t), y_1(t), z_1(t)]^T, \ldots, [x_N(t), y_N(t), z_N(t)]^T\Big\}$.

c) Action: : To optimize the UAV trajectory, the action of UAV n at time slot t is defined as $a_n(t) = \{\Delta \boldsymbol{r}_n(t)\} \in \mathcal{A}$, where $\Delta \boldsymbol{r}_n(t) = \|\boldsymbol{r}_n(t+1) - \boldsymbol{r}_n(t)\|$ indicates the UAV's position change at time t.

d) Transition Probability: : $\mathcal{P}(o_n(t+1)|o_n(t), a_n(t))$ denotes the probability of agent n transitioning from state $o_n(t)$ to $o_n(t+1)$ at time slot t given action $a_n(t)$.

e) Reward function $\mathcal{R}$: As the aim of this paper is to maximize total system utility through UAV trajectory optimization, the reward function should be positively correlated with this objective. Given the cooperative strategy among UAVs, a shared reward function is used for all agents:

$$\mathcal{R} = \begin{cases} U(t), & \text{if all constraints are satisfied,} \\ U(t) - P_n, & \text{otherwise.} \end{cases} \tag{53}$$

A positive reward encourages the agents to maximize system utility when constraints are met. When any constraint is violated—such as energy limits of UAVs or users, task delay violations, out-of-bound flights, or collisions—a penalty term P_n is imposed to discourage unsafe or undesirable behavior.

2) MAPPO Framework: MAPPO is an extension of PPO under the Centralized Training with Decentralized Execution (CTDE) paradigm. It optimizes a shared global value function while allowing independent policy execution by each agent. MAPPO provides a distributed learning perspective for intelligent resource allocation and task offloading in UMEC systems, especially under partial observability. The framework includes N agents, each UAV running PPO with its own policy and value networks.

As illustrated in Fig. 2, each agent operates in two phases: (1) centralized training and (2) decentralized execution. The components are defined as follows:

- **Policy Network:**
  - **Input:** Local observation o_n of agent n.
  - **Output:** Action probability distribution $\pi_{\theta_n}(a_n \mid o_n)$.

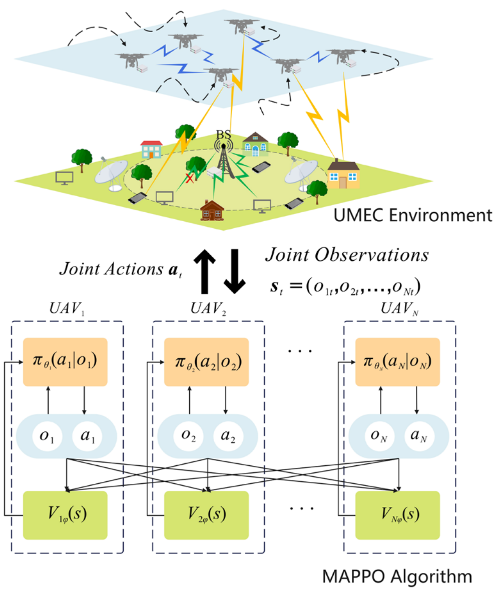

BS
UMEC Environment
Joint Actions $\boldsymbol{a}_t$
Joint Observations
$\boldsymbol{s}_t = (o_{1t}, o_{2t}, \ldots, o_{Nt})$
UAV_1
UAV_2
UAV_N
$\pi_{\theta_1}(a_1|o_1)$
$\pi_{\theta_2}(a_2|o_2)$
$\pi_{\theta_N}(a_N|o_N)$
o_1 a_1
o_2 a_2
o_N a_N
$V_{1\varphi}(s)$
$V_{2\varphi}(s)$
$V_{N\varphi}(s)$
MAPPO Algorithm

Fig. 2. MAPPO framework in UMEC system.

- **Value Network:**
  - **Input:** Global state $\mathbf{s}_t$ (i.e., joint observation $o_{1t}, o_{2t}, \ldots, o_{Nt}$).
  - **Output:** State value estimate $V_\phi(\mathbf{s}_t)$.

To prevent overly large policy updates, the surrogate objective function is clipped:

$$L_n^{\text{CLIP}}(\theta_n) = \mathbb{E}_t\Big[\min\Big(r_{n,t}(\theta_n)A_n(\mathbf{s}_t, \mathbf{a}_t),\\ \text{clip}\left(r_{n,t}(\theta_n), 1-\epsilon, 1+\epsilon\right) A_n(\mathbf{s}_t, \mathbf{a}_t)\Big)\Big], \quad (54)$$

$$r_{n,t}(\theta_n) = \frac{\pi_{\theta_n}(a_{n,t} \mid o_{n,t})}{\pi_{\theta_{\text{old},n}}(a_{n,t} \mid o_{n,t})}, \quad (55)$$

where $r_{n,t}(\theta_n)$ is the importance sampling ratio between new and old policies. The joint action is defined as $\mathbf{a}_t = (a_{1,t}, \ldots, a_{N,t})$. The advantage function $A_n(\mathbf{s}_t, \mathbf{a}_t)$ is estimated using the Generalized Advantage Estimator (GAE):

$$A_n(\mathbf{s}_t, \mathbf{a}_t) = \sum_{l=0}^{T-t} (\gamma\lambda)^l \delta_n^{t+l} \quad (56)$$

The Temporal Difference (TD) error δ_n^{t+l} is computed as:

$$\delta_n^{t+l} = r_n^t + \gamma V_\phi(\mathbf{s}_{t+1}) - V_\phi(\mathbf{s}_t).$$

a) Value Function Loss: The value network is optimized by minimizing the mean squared error between estimated and actual returns:

$$L^{\text{Value}}(\phi) = \mathbb{E}_t\left[\left(V_\phi(\mathbf{s}_t) - R_t\right)^2\right], \quad (57)$$

where $R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$ is the cumulative discounted return.

3) Training Procedure:

- **Data Collection:** All agents interact with the environment using their current policies, storing the trajectory data:

$$\{(\mathbf{s}_t, o_{1,t}, a_{1,t}, \ldots, o_{N,t}, a_{N,t}, r_t)\}, \quad (58)$$

- **Advantage Estimation:** Estimate advantage A_t using Equation (56).
- **Policy Update:** For each agent n, update policy parameters θ_n as:

$$\theta_{\text{new},n} = \arg\max_{\theta_n} L_n^{\text{CLIP}}(\theta_n), \quad (59)$$

  using gradient ascent:

$$\theta_n \leftarrow \theta_n + \alpha_{\text{actor}} \nabla_{\theta_n} L_n^{\text{CLIP}}(\theta_n), \quad (60)$$

  where α_{actor} is the learning rate for the actor network.
- **Value Update:** The global value network parameters ϕ are updated by:

$$\phi_{\text{new}} = \arg\min_{\phi} L^{\text{Value}}(\phi), \quad (61)$$

  using gradient descent:

$$\phi \leftarrow \phi - \alpha_{\text{critic}} \nabla_\phi L^{\text{Value}}(\phi), \quad (62)$$

  where α_{critic} is the learning rate for the critic network.

C. Overall Procedure of UOURS

By integrating the designed resource scheduling and matching strategy with the UAV flight trajectory optimization strategy, this section presents the complete workflow of the UOURS algorithm, as illustrated in Algorithm 4. The UOURS algorithm first utilizes the MAPPO-based UAV trajectory design approach outlined in Algorithm 3, and stores the optimized policy network parameters for UAV flight.

Then, the environment parameters are initialized. At each time slot, each UAV determines its flight trajectory based on the currently observed local state and the learned optimal flight policy. Furthermore, using the many-to-one matching mechanism from Algorithm 2, the optimal offloading decision $\boldsymbol{X}^*$, along with the corresponding resource allocations $\boldsymbol{B}^*$ and $\boldsymbol{F}^*$ for that time slot, are determined.

Based on the UOURS algorithm, efficient and coordinated scheduling of network resources in the UMEC system can be achieved, thereby maximizing the overall energy efficiency of the system.

V. Simulation Results And Analysis

This subsection presents numerical simulation experiments to validate the effectiveness of the UOURS algorithm proposed in this chapter. First, we introduce the simulation environment of the UMEC system and the settings of algorithmic parameters. Then, the performance of UOURS is compared with baseline algorithms in terms of system utility and convergence behavior.

A. Experimental Settings

Table I details the parameters associated with the UMEC model. These parameters were selected based on prior works, particularly [39] and [40].

Algorithm 3 UAV Trajectory Optimization Algorithm Based on MAPPO

```
Input: Environment parameters: number of users $M$, UAVs $N$, initial environment state $s_0$;
       Training parameters: learning rates $\alpha_{\text{actor}}$, $\alpha_{\text{critic}}$, discount factor $\gamma$, GAE parameter $\lambda$, clipping threshold $\epsilon$, number of training iterations $K$.
Output: Optimized policy network parameters $\{\theta_1^*, \ldots, \theta_N^*\}$ and value network parameters $\phi^*$.
1  Initialize policy network parameters $\{\theta_1, \ldots, \theta_N\}$ and value network parameters $\phi$;
2  for $k = 1$ to $K$ do
3      Reset the environment and obtain initial global state $s_0$;
4      for $t = 1$ to $T$ do
5          for $n = 1$ to $N$ do
6              Sample action $a_{n,t} \sim \pi_{\theta_n}(\cdot \mid o_{n,t})$;
7          end
8          Execute joint action $\mathbf{a}_t = (a_{1,t}, \ldots, a_{N,t})$;
9          Compute optimal matching list $\Pi^*(t)$ and resource allocations $B_{m,n}^*(t)$ and $f_{m,n}^*(t)$ using Algorithm 2;
10         Execute actions and observe reward $r_t$ and next state $\mathbf{s}_{t+1}$;
11         Store trajectory $\left(\mathbf{s}_t, \{o_{n,t}, a_{n,t}\}_{n=1}^{N}, r_t\right)$;
12     end
13     for $t = 1$ to $T$ do
14         Compute advantage $A_t = \sum_{l=0}^{T-t} (\gamma\lambda)^l \delta_{t+l}$ using Equation (56);
15     end
16     for $n = 1$ to $N$ do
17         Compute importance sampling ratio using Equation (55);
18         Compute clipped surrogate objective using Equation (54);
19         Update policy network parameters using Equation (59);
20     end
21     Compute value loss using Equation (57);
22     Update value network parameters using Equation (61);
23 end
```

1) Baseline Algorithms: The baseline algorithms include the MAAC algorithm (multi-agent actor-critic) from [41], and the single-agent PPO algorithm from [42]. We choose these two methods because they are representative in multi-agent and single-agent reinforcement learning, respectively, and are widely used in UAV and edge computing scenarios. MAAC applies centralized training with access to global information but may suffer from high training complexity and sensitivity to the number of agents. PPO, as a single-agent method aggregating the states and actions of all UAVs, tends to struggle with high dimensionality and poor convergence stability. Including both algorithms allows for a comprehensive comparison between centralized multi-agent learning and aggregated single-agent learning approaches.

B. Result Analysis

Figure 3 shows the training reward curve for the reinforcement learning component of the UOURS algorithm, demonstrating convergence performance. In the early training stage (0–280 episodes), the reward fluctuates significantly due to exploration. After 280 episodes, the reward increases sharply from around 0 to approximately 1250. In the later stage (300–500 episodes), the reward stabilizes around 1000, indicating good convergence and training stability.

Algorithm 4 UOURS Algorithm

```
Input: Environment parameters and training parameters.
Output: Optimal UAV trajectories $\boldsymbol{R}^*$, optimal offloading decision $\boldsymbol{X}^*$, and resource allocations $\boldsymbol{B}^*$ and $\boldsymbol{F}^*$.
1  Compute the converged policy network parameters $\{\theta_1^*, \ldots, \theta_N^*\}$ using Algorithm 3;
2  Reset the environment and obtain the initial global state $s_0$;
3  for $t = 1$ to $T$ do
4      for $n = 1$ to $N$ do
5          Sample action $a_{n,t} \sim \pi_{\theta_n^*}(\cdot \mid o_{n,t})$;
6      end
7      Compute the optimal offloading decision $\boldsymbol{X}^*$ and resource allocations $\boldsymbol{B}^*$ and $\boldsymbol{F}^*$ using Algorithm 2;
8      Execute the actions and observe the overall system utility $U(t)$ and the next state $\mathbf{s}_{t+1}$;
9  end
```

TABLE I
MAIN SIMULATION PARAMETERS

Parameter	Value
Number of edge nodes M	5∼30
Number of UAVs N	4
Number of total time slots T	500
The length of the area X^{max}	50m
The width of the area Y^{max}	50m
Minimum altitude of UAVs Z^{min}	10m
Maximum altitude of UAVs Z^{max}	20m
Minimum distance between UAVs D^{min}	3m
Data size of tasks d_m	100∼150kb
CPU cycles per bit c_m	500∼1000cycles/bit
Maximum velocity of UAVs v^{max}	1.73m/s
Elevation angle of UAV ϕ_n	90°
Environmental constants a and b	9.61,0.16
Bandwidth B^{G2A}	20MHz
Maximum transmit power of user p_m	1∼1.2W
Maximum transmit power of UAV p_n	5W
Noise power N_G	-70dBm
Computation resource of user f_m	0.8∼1GHz
Computation resource of UAV f_n	10GHz
Task completion delay constraint $T_m^{\max}$	100 ∼ 200 ms
User energy limit $E_m^{\max}$	1000 ∼ 2000 J
UAV energy limit $E_n^{\max}$	5000 ∼ 8000 J

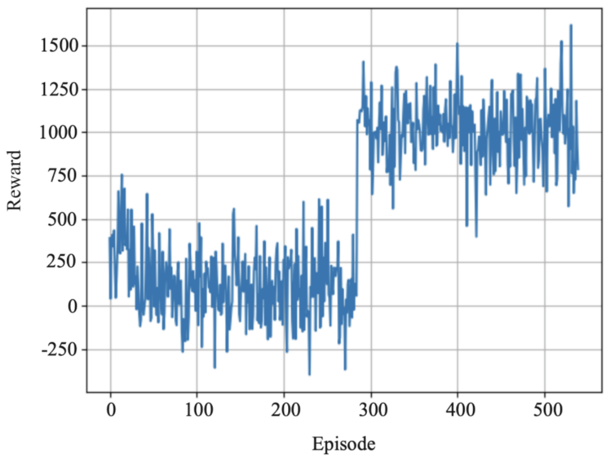
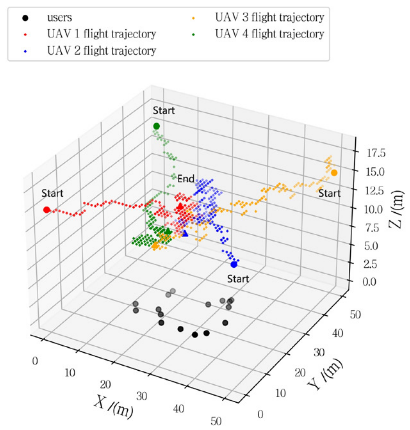
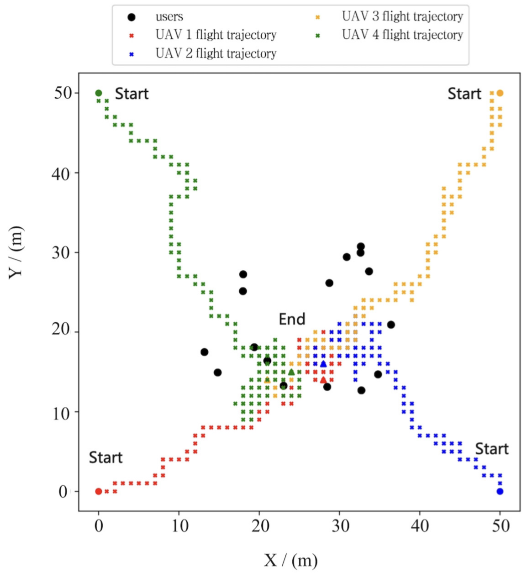


Fig. 3. Convergence performance of the UOURS algorithm in the UMEC system.

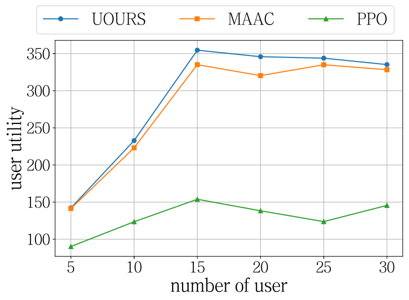


(a) 3D UAV flight trajectories

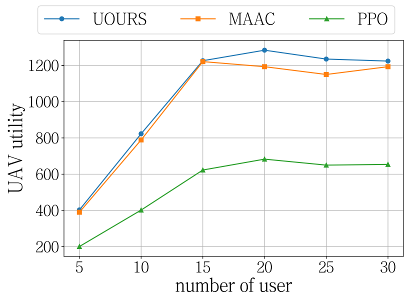


(b) 2D UAV flight trajectories

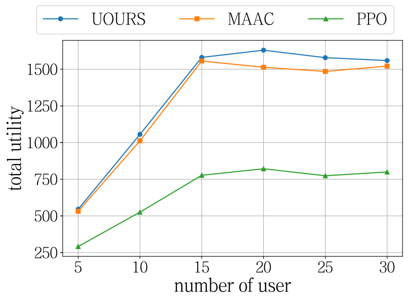
Fig. 4. UAV flight trajectories in the UMEC scenario.



(a) user utility

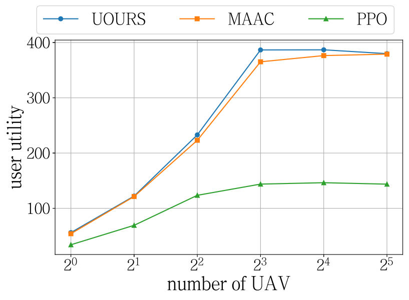


(b) UAV utility

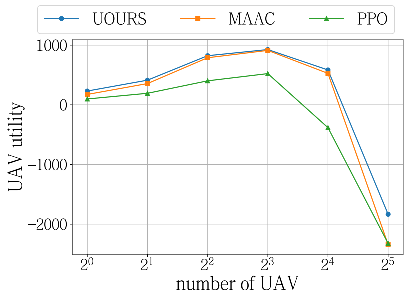


(c) total utility

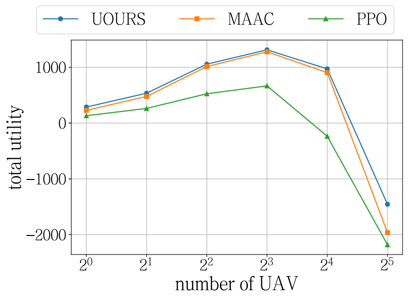
Fig. 5. Utility comparison under different numbers of users.



(a) user utility



(b) UAV utility



(c) total utility

Fig. 6. Utility comparison under different numbers of UAVs.

Figures 4a and 4b show the 3D and 2D UAV flight trajectories under UOURS. The UAVs operate within a 50m × 50m square area with altitude constraints of 10–20m. Black dots represent users, while red, blue, green, and yellow represent four UAVs. Crosses indicate UAV trajectories, circles mark start points, and triangles denote end points. From Figure 4a, we observe purposeful UAV movement towards users with reasonable paths. Figure 4b shows UAVs hovering over user-concentrated areas, reducing task transmission and inter-UAV transfer energy consumption, thereby enhancing system utility and service quality.

Figure 5 compares the utilities of three algorithms under varying numbers of users. Figure 5a shows user utility increasing with user numbers up to 15, due to sufficient MEC resources. Beyond 15 users, utility stabilizes due to resource saturation and matching limitations. Figure 5b shows UAV

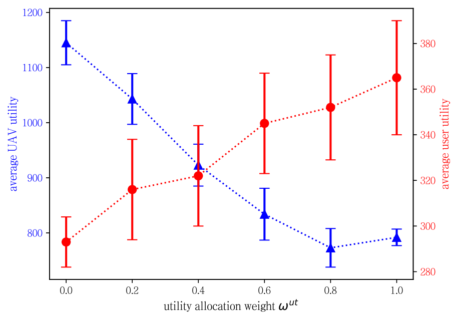


Fig. 7. User utility and UAV utility under different utility allocation weights.

utility following a similar trend. Figure 5c presents total utility (user + UAV). UOURS consistently outperforms, with up to 7.4% improvement over MAAC and 41.7% over PPO.

Figure 6 compares the utilities with varying numbers of UAVs. Figure 6a shows user utility improving with more UAVs up to 2^3, then stabilizing. Figure 6b shows UAV utility peaking and then declining due to idle UAVs consuming energy without task assignments. Figure 6c confirms that UOURS remains superior under different UAV counts, while MAAC and PPO degrade due to increased agent complexity and dimensionality.

Figure 7 shows the user utility and UAV utility under different utility allocation weights ω^{ut}. As ω^{ut} increases from 0 to 1, the system utility gradually shifts from prioritizing UAV utility to emphasizing user utility. It can be observed that the average UAV utility decreases from approximately 1150 to around 800, while the average user utility increases from about 280 to around 350. This indicates that ω^{ut} serves as a trade-off factor between user and UAV utility. By adjusting ω^{ut}, the system can flexibly optimize the strategy according to practical requirements, such as service quality demands, user battery capacity, or UAV endurance, in order to achieve an optimal utility balance.

The superior performance of UOURS can be attributed to its ability to jointly optimize resource allocation and trajectory design in a multi-agent framework, which enhances scalability and stability. In contrast, MAAC suffers from high training complexity and instability as the number of UAVs increases, while PPO struggles with scalability and convergence due to its single-agent aggregation scheme. These limitations become particularly evident in scenarios with a larger UAV population, dynamic task arrivals, and heterogeneous resource demands, where UOURS consistently achieves higher system utility.

VI. Conclusion

This paper focuses on the optimization of system utility in an energy-constrained UAV-assisted UMEC system. First, user and UAV utility are defined, and utility models are constructed from the perspectives of user task processing efficiency, delay satisfaction, and UAV energy consumption and revenue. On this basis, an optimization-based resource scheduling algorithm called UOURS, which integrates MAPPO and a bargaining mechanism, is proposed. By simultaneously optimizing UAV flight paths, computation mode, bandwidth and computational resource management, the algorithm achieves the maximization of total system utility. Experimental evaluations demonstrate that UOURS outperforms the baseline algorithms, achieving improvements of up to 7.4% in system utility. Notably, it maintains high utility and stability even when the number of users increases and system resources become more constrained. This research provides an effective solution for utility optimization in UAV-assisted edge computing networks and holds significant theoretical and practical value.

In future work, we will incorporate realistic UAV constraints such as hardware limitations and flight regulations into the model. We will also explore larger-scale scenarios and uncertain network conditions to further validate the framework.

References

[1] W. Hong et al., "The role of millimeter-wave technologies in 5G/6G wireless communications," *IEEE J. Microw.*, vol. 1, no. 1, pp. 101–122, Jan. 2021.

[2] G. Sun et al., "Aerial reliable collaborative communications for terrestrial mobile users via evolutionary multi-objective deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 24, no. 7, pp. 1–18, Jul. 2025.

[3] J. Wang et al., "Generative AI for integrated sensing and communication: Insights from the physical layer perspective," *IEEE Wireless Commun.*, vol. 31, no. 5, pp. 246–255, Oct. 2024.

[4] S. Yu and R. Langar, "Collaborative computation offloading for multi-access edge computing," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 689–694.

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

[6] J. Wang et al., "Generative AI enabled robust data augmentation for wireless sensing in ISAC networks," *IEEE J. Sel. Areas Commun.*, early access, Sep. 24, 2025, doi: 10.1109/JSAC.2025.3613672.

[7] J. Wang et al., "Feature engineering for wireless communications and networking: Concepts, methodologies, and applications," 2025, *arXiv:2507.19837*.

[8] F. Zhao, B. Yang, C. Li, C. Zhang, L. Zhu, and G. Liang, "Two-layer consensus based on primary-secondary consortium chain data sharing for Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13828–13838, Sep. 2024.

[9] 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, Apr. 2020.

[10] J. Chen et al., "Deep reinforcement learning based resource allocation in multi-UAV-aided MEC networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 296–309, Jan. 2023.

[11] B. Liu, Y. Wan, F. Zhou, Q. Wu, and R. Q. Hu, "Resource allocation and trajectory design for MISO UAV-assisted MEC networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4933–4948, May 2022.

[12] Z. Sun, G. Sun, and M. Yuan, "Online collaborative task offloading and resource allocation for MEC system," in *Proc. ACM Turing Award Celebration Conf.*, Jul. 2024, pp. 224–225, doi: 10.1145/3674399.3674476.

[13] F. Song, M. Deng, H. Xing, Y. Liu, F. Ye, and Z. Xiao, "Energy-efficient trajectory optimization with wireless charging in UAV-assisted MEC based on multi-objective reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 10867–10884, Dec. 2024.

[14] Y. Chen et al., "Full-duplex integrated sensing, communication, and computation over low-altitude wireless networks," 2025, *arXiv:2504.18143*.

[15] Q. Zhou, Y. Liu, H. Feng, and L. Wang, "PRU group allocation and dynamic rate-splitting design for power minimization in IRS-assisted UAV MEC systems with RSMA," *IEEE Trans. Green Commun. Netw.*, vol. 9, no. 3, pp. 1–15, Sep. 2025.

[16] Y. Gong, J. Fan, R. Zhang, D. Niyato, Y. Yao, and X. Chang, "Safe and economical UAV trajectory planning in low-altitude airspace: A hybrid DRL-LLM approach with compliance awareness," 2025, *arXiv:2506.08532*.

[17] R. Huang et al., "Dynamic task offloading for multi-UAVs in vehicular edge computing with delay guarantees: A consensus ADMM-based optimization," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 13696–13712, Dec. 2024.

[18] Y. Zhang, Y. Zhu, F. Yan, Z. Li, and L. Shen, "Semidefinite programming based resource allocation for energy consumption minimization in software defined wireless sensor networks," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–6.

[19] M. Yan, C. Aun Chan, A. F. Gygax, C. Li, A. Nirmalathas, and I. Chih-Lin, "Efficient generation of optimal UAV trajectories with uncertain obstacle avoidance in MEC networks," *IEEE Internet Things J.*, vol. 11, no. 23, pp. 38380–38392, Dec. 2024.

[20] J. Huang, Y. Yin, Y. Zhao, Q. Duan, W. Wang, and S. Yu, "A game-theoretic resource allocation approach for intercell device-to-device communications in cellular networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 4, pp. 475–486, Oct. 2016.

[21] L. Huang, R. Sun, N. Cheng, Y. Hui, and D. Liang, "Delay-oriented knowledge-driven resource allocation in SAGIN-based vehicular networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2023, pp. 1–6.

[22] N. Lin, H. Tang, L. Zhao, S. Wan, A. Hawbani, and M. Guizani, "A PDDQNLP algorithm for energy efficient computation offloading in UAV-assisted MEC," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 8876–8890, Dec. 2023.

[23] G. Sun, Y. Wang, H. Yu, and M. Guizani, "Proportional fairness-aware task scheduling in space-air-ground integrated networks," *IEEE Trans. Services Comput.*, vol. 17, no. 6, pp. 4125–4137, Nov. 2024.

[24] Z. Wang, G. Sun, Y. Wang, F. Yu, and D. Niyato, "Cluster-based multi-agent task scheduling for space-air-ground integrated networks," *IEEE Trans. Cognit. Commun. Netw.*, early access, Mar. 21, 2025, doi: 10.1109/TCCN.2025.3553297.

[25] H. Yang, M. Zheng, Z. Shao, Y. Jiang, and Z. Xiong, "Intelligent computation offloading and trajectory planning for 3-D target search in low-altitude economy scenarios," *IEEE Wireless Commun. Lett.*, vol. 14, no. 4, pp. 949–953, Apr. 2025.

[26] H. He, X. Yang, F. Huang, H. Shen, and H. Tian, "Enhancing QoE in large-scale U-MEC networks via joint optimization of task offloading and UAV trajectories," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 35710–35723, Nov. 2024.

[27] Z. Wang, T. Wei, G. Sun, X. Liu, H. Yu, and D. Niyato, "Multi-UAV enabled MEC networks: Optimizing delay through intelligent 3D trajectory planning and resource allocation," 2024, *arXiv:2409.17882*.

[28] Y. Xu, T. Zhang, D. Yang, Y. Liu, and M. Tao, "Joint resource and trajectory optimization for security in UAV-assisted MEC systems," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 573–588, Jan. 2021.

[29] K. Xiang and Y. He, "UAV-assisted MEC system considering UAV trajectory and task offloading strategy," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 4677–4682.

[30] X. Du, X. Li, N. Zhao, and X. Wang, "A joint trajectory and computation offloading scheme for UAV-MEC networks via multi-agent deep reinforcement learning," in *Proc. IEEE Int. Conf. Commun.*, Rome, Italy, Jun. 2023, pp. 5438–5443.

[31] Y. Yang, J. Yang, H. Xu, J. Hu, and T. Song, "Trajectory and offloading policy optimization in age-of-information-aware UAV-assisted MEC systems," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Aug. 2023, pp. 175–180.

[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. Mobile Comput.*, vol. 21, no. 10, pp. 3536–3550, Oct. 2022.

[33] H. Hao, C. Xu, W. Zhang, S. Yang, and G.-M. Muntean, "Joint task offloading, resource allocation, and trajectory design for multi-UAV cooperative edge computing with task priority," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8649–8663, Sep. 2024.

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

[35] Z. Kuang, Y. Shi, S. Guo, J. Dan, and B. Xiao, "Multi-user offloading game strategy in OFDMA mobile cloud computing system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12190–12201, Dec. 2019.

[36] Y. Zhou et al., "Secure communications for UAV-enabled mobile edge computing systems," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 376–388, Jan. 2020.

[37] Z. Kuang, H. Wang, J. Li, and F. Hou, "Utility-aware UAV deployment and task offloading in multi-UAV edge computing networks," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14755–14770, Apr. 2024.

[38] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109, Jan. 1982.

[39] F. Pervez, A. Sultana, C. Yang, and L. Zhao, "Energy and latency efficient joint communication and computation optimization in a multi-UAV-assisted MEC network," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 1728–1741, Mar. 2024.

[40] J. Wang, C. Jin, Q. Tang, N. N. Xiong, and G. Srivastava, "Intelligent ubiquitous network accessibility for wireless-powered MEC in UAV-assisted B5G," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2801–2813, Oct. 2021.

[41] Z. Gao, J. Fu, Z. Jing, Y. Dai, and L. Yang, "MOIPC-MAAC: Communication-assisted multiobjective MARL for trajectory planning and task offloading in multi-UAV-assisted MEC," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 18483–18502, May 2024.

[42] H. Li, K. Xiong, P. Fan, and K. B. Letaief, "Deep reinforcement learning based task offloading and resource allocation in small cell MEC," in *Proc. IEEE Int. Perform., Comput., Commun. Conf. (IPCCC)*, Nov. 2023, pp. 475–480.