Robust Design for Computing Uncertainty and Flying Jittering in Multi-UAV-Assisted MEC

Yucheng Gong[1], **Bin Li**[2*]

[1, 2] School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China
202383840003@nuist.edu.cn, bin.li@nuist.edu.cn

Abstract

This paper investigates the problem of joint computation and communication optimization in multiple unmanned-aerial-vehicles (UAVs)-assisted mobile edge computing (MEC) networks, where the practical challenges arising from task data sizes uncertainty and UAV fly jittering are taken into account. We propose a robust optimization framework to address these uncertainties, aiming to minimize the weighted energy consumption. This is achieved by jointly optimizing UAV trajectories, task partitioning, and the allocation of computation and communication resources across multiple UAVs. The problem is solved using a distributionally robust optimization soft actor–critic algorithm, which enhances system robustness by accounting for worst-case task demand distributions. Numerical simulations demonstrate that the proposed algorithm significantly outperforms conventional deep reinforcement learning approaches in terms of energy consumption, while ensuring reliable task completion in dynamic environments.

Introduction

To efficiently deal with massive amounts of data at the network edges, mobile edge computing (MEC) has been introduced as a promising paradigm. In MEC, nearby servers function as edge clouds, offering substantial computational power to user equipments (UEs) while significantly minimizing computation offloading delays (Zhao et al. 2025). However, serving computationally intensive tasks in remote regions presents significant challenges owing to inadequate communication conditions and unpredictable MEC environments (Jia et al. 2025). Additionally, in hotspot regions where numerous UEs demand services simultaneously, the constrained computational resources present a significant challenge for MEC servers in ensuring the satisfactory user experience. Recent advances in unmanned aerial vehicle (UAV) technologies have established UAVs as a promising platform for the MEC network due to their exceptional mobility and capacity to enhance coverage. By serving as flexible aerial edge servers, they can provide computing support in remote areas and mitigate congestion in hotspots, thereby ensuring high-quality computing services (Zhang et al. 2024).

In UAV-assisted MEC systems, effective flight trajectory design and computational task allocation is essential for ensuring the effective operation of the system. However, several practical uncertainties persist. First, the performance of computation offloading is often hindered by unpredictable transmission latency and packet loss in MEC networks, which can degrade the reliability of edge computing nodes (Lu et al. 2020). On the other hand, the application scenarios of MEC have the characteristic of task suddenness (Ye et al. 2023). Specifically, the tasks generated by UE are highly dynamic. Its scale and probability distribution both exhibit randomness and uncertainty. Regarding the stability of the system and the quality of service poses a serious challenge.

To guarantee worst-case performance and prevent system failures, a robust computation offloading strategy for multi-UAV MEC networks is essential. This paper thus presents such a strategy, accounting for uncertainties in task data size and UAV trajectories to enhance system robustness and minimize weighted energy consumption. The main contributions are as follows:

1) This paper addresses uncertainties in multi-UAV MEC systems by formulating an energy minimization problem. The solution involves the joint optimization of UAV trajectories, UE associations, task segmentation, and resource allocation to enhance computation offloading reliability.
2) The formulated problem is challenging to solve globally due to the highly interdependent optimization variables under uncertainty constraints. To this end, we resort to the deep reinforcement learning (DRL) and propose a distributionally robust optimization soft actor–critic (DRO-SAC) algorithm.
3) We evaluate the complexity of the DRO-SAC algorithm and demonstrate its effectiveness in ensuring convergence. The algorithm's robustness is validated through numerical simulations, showcasing its ability to minimize energy consumption while accounting for bounded estimation errors.

System Model and Problem Formulation

As depicted in Fig. 1, our considered multi-UAV-assisted MEC network is composed of J UAVs and I UEs. Each UAV is fitted with a uniform planar array (UPA) antenna

[*] Corresponding author.

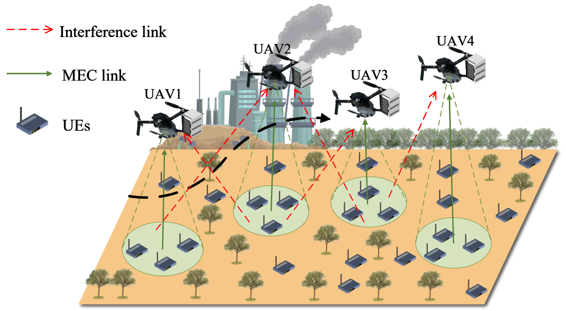

Figure 1: The collaborative MEC system with multiple UAVs

of dimensions $A_r = A_x \times A_y$, whereas every UE is assumed to possess a single antenna. To facilitate subsequent analysis, let $\mathcal{J} \triangleq \{1, 2, \cdots, J\}$, $\mathcal{I} \triangleq \{1, 2, \cdots, I\}$, and $\mathcal{L} \triangleq \{1, 2, \cdots, L\}$ denote the index sets for the J UAVs, I UEs, and L time slots, respectively. The total flight period is $T = L\delta$, where δ is the slot duration. In each slot $l \in \mathcal{L}$, each UE generates resource-intensive computation tasks with uncertain input data size. To mitigate interference, we enforce a constraint where each UE can be associated with only one UAV at any given time step, whereas a single UAV is allowed to serve multiple UEs simultaneously (Chen et al. 2023). The corresponding association variable is defined as

$$\sum_{j=1}^{J} \alpha_{i,j} \leq 1, \forall i \in \mathcal{I}, \tag{1}$$

$$\alpha_{i,j} \in \{0, 1\}, \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{2}$$

where $\alpha_{i,j} = 1$ indicates that UE i selects UAV j for computation offloading, otherwise $\alpha_{i,j} = 0$.

UAV Movement Model

In the considered MEC system, multiple UAVs are dispatched to assist a group of ground UEs. We adopt a 3D Cartesian coordinate system in this work. Each UE i is assumed to be static and located at $\mathbf{u}_i = (x_i, y_j)$, while the coordinates of the UAV j are denoted as $\mathbf{q}_j[l] = (x_j[l], y_j[l], H)$ during the l-th time slot.

Unlike many prior works that assume UAVs strictly follow their planned trajectories, in practice UAV flight paths are inevitably affected by external factors such as positioning inaccuracy, imperfect flight control, and air turbulence. To accurately model this realistic phenomenon, we employ a disturbance model where the actual trajectory is represented as the superposition of the planned trajectory and a random perturbation

$$\mathbf{q}_j[l] = \hat{\mathbf{q}}_j[l] + \Delta\mathbf{q}_j[l], \tag{3}$$

where $\hat{\mathbf{q}}_j[l]$ represents the planned position and $\Delta\mathbf{q}_j[l]$ denotes the positional deviation induced by UAV jitter.

To ensure safe operation and avoid conflicts, each UAV must consider the states of other UAVs during trajectory planning. Accordingly, the transformations of UAV positions across different time slots, determined by the flight velocity $\mathbf{v}_j[l]$ and acceleration $\mathbf{a}_j[l]$, must adhere to the following constraints

$$\mathbf{q}_j[l+1] = \mathbf{q}_j[l] + \mathbf{v}_j[l]\,\delta + \frac{1}{2}\mathbf{a}_j[l]\,\delta^2, \tag{4}$$

$$\|\mathbf{q}_k[l] - \mathbf{q}_m[l]\|^2 \geq d_{\text{dim}}^2, \tag{5}$$

where d_{dim} denotes the minimum safe distance between UAVs during flight.

Furthermore, instead of directly imposing a deterministic velocity bound, we employ a probabilistic velocity constraint. Specifically, the probability that a UAV's actual speed exceeds the maximum allowable speed $v_{\max}$ must not surpass a given threshold ρ_v, which is calculated as

$$\mathbb{P}(\|\mathbf{q}_j[l+1] - \mathbf{q}_j[l]\|/\delta \leq v_{\max}) \geq 1 - \rho_v. \tag{6}$$

The flying energy consumption is another critical component in UAV-assisted MEC networks. When a UAV moves at velocity $\mathbf{v}_j[l]$, its propulsion power consumption can be modeled as(Liu et al. 2022)

$$\begin{aligned} P_j^{\text{fly}}[l] &= \tfrac{1}{2}\, d_0\, \rho\, g\, A_0 \,\|\mathbf{v}_j[l]\|^3 + P_1\left(1 + \frac{3\|\mathbf{v}_j[l]\|^2}{U_{\text{tip}}^2}\right) \\ &+ P_2\left(\sqrt{1 + \frac{\|\mathbf{v}_j[l]\|^4}{4v_0^4}} - \frac{\|\mathbf{v}_j[l]\|^2}{2v_0^2}\right)^{\frac{1}{2}}, \end{aligned} \tag{7}$$

where d_0 denotes the fuselage drag ratio, ρ is the air density, and A_0 represents the area of rotors. Furthermore, g corresponds to the rotor solidity, P_1 and P_2 are the blade power and the hovering induced power, respectively. Finally, v_0 and U_{tip} represent the mean rotor velocity and the tip speed of the blade. The corresponding flight energy in slot l is then written as

$$E_j^{\text{fly}}[l] = P_j^{\text{fly}}[l]\delta. \tag{8}$$

Communication Model

In the considered multi-UAV-assisted MEC system, all UEs communicate with UAVs through wireless links in the uplink direction. However, due to the existence of obstacles in practical deployment scenarios, Line-of-Sight (LoS) transmission between UEs and UAVs cannot always be guaranteed. As a result, the wireless channel is modeled as a Rician fading channel that combines both LoS and Non-LoS (NLoS) components. Specifically, the estimated CSI between UE i and UAV j at slot l is calculated as(Hua et al. 2021)

$$\mathbf{h}_{i,j}[l] = \sqrt{\rho d_{i,j}^{-\beta}[l]}\left(\sqrt{\tfrac{\varsigma}{\varsigma+1}}\bar{\mathbf{h}}_{i,j}^{L}[l] + \sqrt{\tfrac{1}{1+\varsigma}}\tilde{\mathbf{h}}_{i,j}^{N}[l]\right), \tag{9}$$

where $d_{i,j}[l]$ is the distance between UE i and UAV j, β denotes the path-loss exponent, and ς is the Rician factor.

LoS component from UAV j to UE i during the l-th time slot is represented by $\bar{\mathbf{h}}_{i,j}^{L}[l] \in \mathbb{C}^{A_r \times 1}$. Additionally, the NLoS component $\tilde{\mathbf{h}}_{i,j}^{N} \in \mathbb{C}^{A_r \times 1}$ is modeled as a complex Gaussian distribution with zero mean and unit variance, i.e., $\tilde{\mathbf{h}}_{i,j}^{N} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$.

Leveraging UAVs for edge computing by offloading computation tasks to them offers significant advantages. After task completion, the computed results are delivered to the UEs through the downlink. To model this, we define the transmitted signal for task $D_i[l]$ as

$$x_i[l] = \sqrt{p_i[l]} s_i[l], \tag{10}$$

where $p_i[l]$ represents the transmission power of UE i at time slot l, and $s_i[l]$ denotes the unit-norm associated with task $D_i[l]$. To reduce interference between channels, beamforming techniques are applied at UAVs. Consequently, the signal received by UAV j is expressed as

$$\begin{aligned} y_{i,j}[l] = &\mathbf{w}_{i,j}^{\mathrm{H}}[l]\,\mathbf{h}_{i,j}[l]\sqrt{p_i[l]} s_i[l] + \\ &\sum_{m=1}^{J}\sum_{k=1,k\neq i}^{I} \mathbf{w}_{i,j}^{\mathrm{H}}[l]\,\alpha_{k,m}\mathbf{h}_{k,m}[l]\sqrt{p_k[l]} s_k[l] \\ &+ \mathbf{w}_{i,j}^{\mathrm{H}}[l]\,\mathbf{n}, \end{aligned} \tag{11}$$

where $\mathbf{w}_{i,j}[l]$ denotes the unit-norm receive beamforming vector between UE i and UAV j at time slot l, with the constraint $\mathbf{w}_{i,j}^{\mathrm{H}}[l]\,\mathbf{w}_{i,j}[l] = 1$. Additionally, $\mathbf{n} \sim \mathcal{CN}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$ denotes additive white Gaussian noise vector with variance σ^2. Consequently, the corresponding signal-to-interference-plus-noise ratio is computed as

$$\Gamma_{i,j}[l] = \frac{|\mathbf{w}_{i,j}^{\mathrm{H}}[l]\,\mathbf{h}_{i,j}[l]|^2 p_i[l]}{\sum\limits_{m=1}^{J}\sum\limits_{k=1,k\neq i}^{I} \alpha_{k,m}|\mathbf{w}_{i,j}^{\mathrm{H}}[l]\,\mathbf{h}_{k,m}[l]|^2 p_k[l] + \sigma^2}. \tag{12}$$

Thus, the offloading rate from UE i to UAV j at time slot l can be formulated as

$$R_{i,j}[l] = B\log_2\left(1 + \Gamma_{i,j}[l]\right), \tag{13}$$

where B represents the channel bandwidth.

Distributionally Robust Task Uncertainty

In practical MEC systems, the task size generated by each UE varies over time due to network dynamics, user behaviors, and device workloads. To avoid making fragile decisions based on a single deterministic model, we adopt a DRO approach that considers a neighborhood of the empirical task-size distribution.

Let $\mathbb{P}_i^0$ denote the empirical distribution of the task size D_i for UE i over the sample space $\Omega = \{\varphi_1, \ldots, \varphi_R\}$ comprising R discrete values. The reference probability distribution can be represented as $\mathbb{P}_i^0 = \{p_{i,1}^0, p_{i,2}^0, \ldots, p_{i,R}^0\}$, where $p_{i,r}^0$ represents the probability that UE i generates a task of size φ_r. Specifically, this paper assumes that each D_i shares a common sample space Ω, where each D_i has a different probability distribution $\mathbb{P}_i$. Let $\{d_i^{(q)}\}_{q=1}^{Q}$ denote the dataset of Q historical observations of D_i, the empirical probability $p_{i,r}^0$ is computed as

$$p_{i,r}^0 = \frac{1}{Q}\sum_{q=1}^{Q}\delta^r\big(d_i^{(q)}\big), \forall r \in 1, 2, \ldots, R, \tag{14}$$

in which $\delta^r(d_i^{(q)})$ is an indicator function, where $\delta^r(d_i^{(q)}) = 1$ when $d^r \leq d_i^{(q)} < d^{r+1}$, and $\delta^r(d_i^{(q)}) = 0$ otherwise. $d^r \leq d_i^{(q)} < d^{r+1}$ indicates that the task size D_i lies within the interval $[d^r, d^{r+1})$. To hedge against sampling errors, we construct an ambiguity set around $\mathbb{P}_i^0$ using the $\mathcal{L}_1$ distance, which is computed as

$$\mathcal{D}_i = \left\{\mathbb{P}_i : d_{\mathcal{L}_1}(\mathbb{P}_i, \mathbb{P}_i^0) \leq \epsilon_i^{\mathcal{L}_1}\right\}. \tag{15}$$

The $\mathcal{L}_1$ distance between two distributions is given by

$$d_{\mathcal{L}_1}(\mathbb{P}_i, \mathbb{P}_i^0) = \sum_{r=1}^{R}\left|p_{i,r} - p_{i,r}^0\right|. \tag{16}$$

Based on the $\mathcal{L}_1$ norm of the distributional distance, the confidence level $\nu^{\mathcal{L}_1}$ is calculated as

$$\Pr\big(d_{\mathcal{L}_1}(\mathbb{P}_i, \mathbb{P}_i^0) \leq \epsilon^{\mathcal{L}_1}\big) \geq 1 - 2R\exp\left(-\frac{2Q\,\epsilon^{\mathcal{L}_1}}{R}\right) = \nu^{\mathcal{L}_1}. \tag{17}$$

Thus, based on the $\mathcal{L}_1$ norm of the uncertainty set, the radius $\epsilon^{\mathcal{L}_1}$ can be formulated as

$$\epsilon^{\mathcal{L}_1} = \sqrt{\frac{R}{2Q}\ln\left(\frac{2R}{1 - \nu^{\mathcal{L}_1}}\right)}. \tag{18}$$

Computation Model

The computing task of UE i in the l-th time slot is characterized by the tuple $\mathbf{\Phi}_i[l] = (D_i[l], C_i[l])$. Here, the random variable $D_i[l]$ represents the task data size, and $C_i[l]$ represents the average CPU cycles required per bit.

This paper adopts a partial offloading strategy, wherein tasks are divided into two components. The portion with data size $D_i^o[l] = \rho_i[l]D_i[l]$ is offloaded to UAV j for execution, while the remainder with data size $D_i^l[l] = (1 - \rho_i[l])D_i[l]$ is processed locally. Here, $\rho_i[l](0 \leq \rho_i[l] \leq 1)$ is defined as the task partitioning factor.

Local computation When UE i processes the computational task locally, the computational delay is calculated as

$$t_i^{\mathrm{loc}}[l] = \frac{D_i^l[l]\,C_i[l]}{f_i[l]}, \tag{19}$$

where $f_i[l]$ denotes the CPU frequency of UE i during time slot l.

The energy consumption during local execution for UE i is modeled as

$$E_i^{\mathrm{loc}}[l] = \kappa D_i^l[l]\,C_i[l] f_i[l]^2, \tag{20}$$

where κ denotes the effective capacitance coefficient, which is determined by the chip architecture.

Computation offloading When task is offloaded from UE i to UAV j, the transmission delay is given by

$$t_{i,j}^{\mathrm{comm}}[l] = \frac{\alpha_{i,j}[l]D_i^o[l]}{R_{i,j}[l]}, \tag{21}$$

where $R_{i,j}[l]$ represents the achievable communication rate between UE i and UAV j. The transmission energy is

$$E_i^{\rm comm}[l] = p_i[l] \cdot t_{i,j}^{\rm comm}[l] = p_i[l] \cdot \frac{\alpha_{i,j}[l] D_i^o[l]}{R_{i,j}[l]}, \qquad (22)$$

where $p_i[l]$ is the transmission power.

After receiving the offloaded task data, the UAV executes the computation using its on-board MEC resources. The computation delay at the UAV is

$$t_{i,j}^{\rm comp}[l] = \frac{\alpha_{i,j}[l] D_i^o[l] C_i[l]}{f_{i,j}^u[l]}, \qquad (23)$$

in which $f_{i,j}^u[l]$ denotes the computation resource assigned by UAV j to UE i.

Accordingly, the service delay for UE i can be expressed as

$$t_i\left[l\right] = \max\left\{ \sum_{j=1}^{J} \left(t_{i,j}^{\rm comm}[l] + t_{i,j}^{\rm comp}[l] \right), t_i^{\rm loc}[l] \right\}. \qquad (24)$$

The energy required for UAV j to compute the offloaded tasks is

$$E_j^{\rm comp}[l] = \kappa \sum_{i=1}^{I} \alpha_{i,j}[l] D_i^o[l] C_i[l] \left(f_{i,j}^u[l] \right)^2. \qquad (25)$$

The aggregate energy consumption comprises local computation energy, transmission energy, UAV computation energy, and UAV flying energy, which can be represented as

$$\begin{aligned} E^{\rm total}[l] = & \sum_{i=1}^{I} \left(E_i^{\rm loc}[l] + E_i^{\rm comm}[l] \right) \\ & + \omega \sum_{j=1}^{J} \left(E_j^{\rm comp}[l] + E_j^{\rm fly}[l] \right), \end{aligned} \qquad (26)$$

where ω is a balancing coefficient between UE and UAV energy consumption.

Problem Formulation

Building upon the established system models, our objective is to pursue the minimization of the weighted system energy consumption while satisfying all task completion constraints. It is achieved through the joint optimization of the following variables, including the UAV trajectories $\mathbf{q} \triangleq \{\mathbf{q}_j\left[l\right], \forall l \in \mathcal{L}, j \in \mathcal{J}\}$, the communication beamforming vectors $\mathbf{w} \triangleq \{\mathbf{w}_{i,j}\left[l\right], \forall l \in \mathcal{L}, j \in \mathcal{J}, i \in \mathcal{I}\}$, the task offloading factors $\boldsymbol{\rho} \triangleq \{\rho_i\left[l\right], \forall i \in \mathcal{I}, l \in \mathcal{L}\}$, the matching factor between UAVs and UEs $\boldsymbol{\alpha} \triangleq \{\alpha_{i,j}\left[l\right], \forall i \in \mathcal{I}, j \in \mathcal{J}\}$, the local CPU frequencies of UEs $\mathbf{f}_l \triangleq \{f_i\left[l\right], \forall l \in \mathcal{L}, i \in \mathcal{I}\}$, and the computational resource allocation of UAVs $\mathbf{f}_u \triangleq \{f_{i,j}^u\left[l\right], \forall l \in \mathcal{L}, j \in \mathcal{J}, i \in \mathcal{I}\}$. Aggregating over the entire flight horizon of L slots, the cumulative system energy consumption is expressed as $\sum_{l=1}^{L} E^{\rm total}[l]$. Since the task data sizes $D_i[l]$ are uncertain, the actual energy cost is also uncertain. Accordingly, the optimization problem is formulated as

$$\min_{\mathbf{w},\boldsymbol{\rho},\mathbf{q},\boldsymbol{\alpha},\mathbf{f}_l,\mathbf{f}_u} \sup_{\mathbb{P}_i} \mathbb{E}_{\mathbb{P}_i}\left[\sum_{l=1}^{L} E^{\rm total}[l] \right] \qquad (27a)$$

$$\text{s.t. } 0 \le \rho_i[l] \le 1, \forall l \in \mathcal{L}, i \in \mathcal{I}, \qquad (27b)$$

$$\sum_{j=1}^{J} \alpha_{i,j} \le 1, \forall i \in \mathcal{I}, \qquad (27c)$$

$$\alpha_{i,j} \in \{0,1\}, \forall j \in \mathcal{J}, i \in \mathcal{I}, \qquad (27d)$$

$$\|\mathbf{a}_j[l]\| \le a_{\max}, \forall l \in \mathcal{L}, j \in \mathcal{J}, \qquad (27e)$$

$$\mathbb{P}(\|\mathbf{v}_j\left[l\right]\| \le v_{\max}) \ge 1 - \rho_v, \forall l \in \mathcal{L}, j \in \mathcal{J}, \qquad (27f)$$

$$\|\mathbf{q}_k[l] - \mathbf{q}_m[l]\|^2 \ge d_{\rm dim}^2, \forall k, m \in \mathcal{J}, k \ne m, \qquad (27g)$$

$$0 \le p_i[l] \le p_i^{\max}, \forall l \in \mathcal{L}, i \in \mathcal{I}, \qquad (27h)$$

$$0 \le f_i[l] \le f_i^{\max}, \forall l \in \mathcal{L}, i \in \mathcal{I}, \qquad (27i)$$

$$0 \le f_{i,j}^u[l] \le f_u^{\max}, \forall i \in \mathcal{I}, l \in \mathcal{L}, j \in \mathcal{J}, \qquad (27j)$$

$$0 \le \sum_{i=1}^{I} \alpha_{i,j}[l] f_{i,j}^u[l] \le f_u^{\max}, \forall j \in \mathcal{J}, l \in \mathcal{L}, \qquad (27k)$$

$$\max_{\|\Delta h_{i,j}[l]\|} t_i[l] \le \delta, \forall i \in \mathcal{I}, l \in \mathcal{L}, \qquad (27l)$$

$$\|\Delta \mathbf{h}_{i,j}[l]\| \le \varepsilon_{i,j}, \forall j \in \mathcal{J}, l \in \mathcal{L}, i \in \mathcal{I}, \qquad (27m)$$

$$\mathbb{P}_i \in \mathcal{D}_i, \ i \in \mathcal{I}. \qquad (27n)$$

Algorithm Design

The multi-UAV-assisted MEC network optimization involves task offloading, UAV trajectories, power allocation, and computation resources, forming a non-convex problem under distributional uncertainties. Conventional methods fail in scalability and adaptivity to time-varying environments. To solve this, we propose the DRO-SAC algorithm, which incorporates DRO to adapt to worst-case variations in task arrivals and dynamically adjust UAV trajectories and resource allocation, as shown in Fig. 2.

The state at time l is defined by the task features $D_i[l]$ and $C_i[l]$ for each UE i, which are modeled within the DRO framework to capture uncertainties. It also includes the UAV mobility, represented by the position $\mathbf{q}_j[l]$ and velocity $\mathbf{v}_j[l]$, which are subject to trajectory disturbances and velocity constraints. Additionally, the state incorporates the channel coefficients $\mathbf{h}_{i,j}[l]$ for the UAV-UE links, modeled using Rician fading, as well as the remaining energy and computational capacity of the UAV.

Thus, the state vector at time l is

$$\begin{aligned} s_n = \Big\{ & D_i[l],\ C_i[l],\ \mathbf{q}_j[l],\ \mathbf{v}_j[l], \\ & \mathbf{h}_{i,j}[l],\ E_j[l],\ f_{i,j}^u[l] \Big\}_{i \in \mathcal{I},\ j \in \mathcal{J}}. \end{aligned} \qquad (28)$$

The action space at each time l includes the user-task scheduling decision $\alpha_{i,j}$, which determines whether UAV j is selected by UE i for offloading. It also contains the offloading ratio $\rho_i[l] \in [0,1]$, which specifies the fraction of the task offloaded to UAVs, along with the transmit power

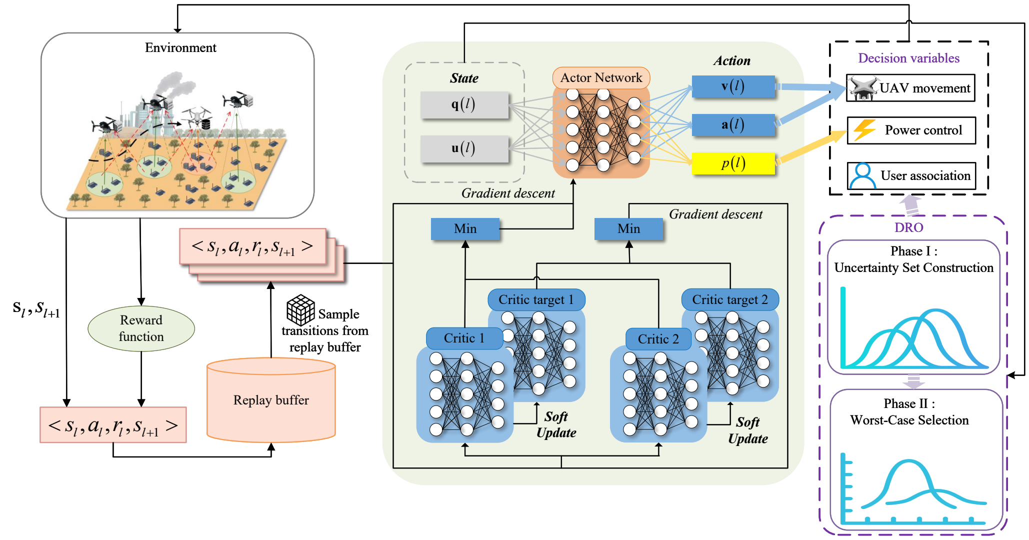

Figure 2: Schematic of the proposed DRO-SAC algorithm.

$p_i[l]$ and local CPU frequency $f_i[l]$ allocated to UE i. Additionally, the action space includes the computation capacity $f^u_{i,j}[l]$ allocated by UAV j to UE i, the adjustments to UAV trajectory $\Delta\mathbf{q}_j[l]$, and the beamforming weight vector $\mathbf{w}_{i,j}[l]$.

Thus, the action vector is

$$\begin{aligned} a_l = \Big\{&\rho_i[l],\ p_i[l],\ f_i[l],\ f^u_{i,j}[l],\\ &\Delta\mathbf{q}_j[l],\ \mathbf{w}_{i,j}[l],\ \tilde{\alpha}_{i,j}\Big\}_{i\in\mathcal{K},j\in\mathcal{J}}. \end{aligned} \tag{29}$$

The reward function emphasizes energy efficiency while penalizing constraint violations. The immediate reward at time l is

$$r(s_l, a_l) = -E^{\text{total}}[l]\cdot P_{\text{dis}}[l]\cdot P_{\text{tm}}[l]\cdot P_o[l], \tag{30}$$

where $P_{\text{tm}}[l]$, $P_{\text{dis}}[l]$, and $P_o[l]$ represent penalties for timeouts, collisions, and UAV boundary violations.

DRO-SAC Training Framework

Unlike traditional SAC, where value functions are trained on empirical samples, DRO-SAC incorporates a DRO strategy to construct an uncertainty set around task demand distributions and selects worst-case samples. This ensures the learned policy is entropy-regularized for stable exploration and robust to variations in task demand, considering the worst-case distribution within the ambiguity set.

Phase I: Uncertainty Set Construction. Using Eq. (15), an uncertainty set is constructed around the empirical distribution of task sizes. This step is crucial for capturing variations in task demand, which traditional models fail to account for.

Phase II: Worst-Case Selection via DRO. After constructing the uncertainty set, the worst-case scenario within the set is identified. The optimization objective is adjusted to account for the worst-case expectation, ensuring robustness to task demand variations while optimizing resource allocation and task offloading.

The SAC framework consists of a policy network $\pi_\phi(a_l|s_l)$, two critic networks $Q_{\theta_1}(s_l, a_l)$ and $Q_{\theta_2}(s_l, a_l)$, and two target critic networks $Q_{\bar{\theta}_1}(s_l, a_l)$ and $Q_{\bar{\theta}_2}(s_l, a_l)$. During training, gradient descent is applied to the critic and policy networks. The loss function for each critic network is given by

$$\begin{aligned} L_Q(\theta) = \mathbb{E}_{(s_l,a_l)\sim\mathcal{R}^\circ}\Bigg[\frac{1}{2}\Big(Q_\theta(s_l,a_l) - \Big(r_l\\ + \gamma\Big(\min_{k=1,2} Q_{\overline{\beta}_k}(s_{l+1},a_{l+1}) - \alpha\log\pi_\phi(a_{l+1}|s_{l+1})\Big)\Big)\Big)^2\Bigg], \end{aligned} \tag{31}$$

The SAC policy network is updated via

$$\begin{aligned} L_\pi(\phi) =&\mathbb{E}_{s\sim\mathcal{R}^\circ,\epsilon_l\in\mathcal{N}_g}\Bigg[\alpha\log\pi_\phi\left(f_\phi(\epsilon_l;s_l)|s_l\right)\\ &-\min_{k=1,2} Q_{\theta_k}\left(s_l, f_\phi(\epsilon_l;s_l)\right)\Bigg]. \end{aligned} \tag{32}$$

The training process involves generating transitions and updating the neural networks, as described in Algorithm 1. The agent optimizes UAV movement and user transmit power based on the trained policy.

Simulation Results and Analyses

This section demonstrates the effectiveness of the DRO-SAC algorithm within a collaborative multi-UAV edge computing network through simulation experiments. We compare the performance of the proposed DRO-SAC algorithm with the following benchmarks.

Algorithm 1: SAC-based deep-reinforcement-learning with DRO strategy

```
Initialize: Set up the replay buffer $\boldsymbol{R}$, policy and critic network parameters: $\phi$ and $\theta_k, k = 1, 2$, and define the sample space $\Omega$;
for episode$= 1, 2, \cdots, E$ do
    Reset the simulation environment and acquire the initial state $s_0$;
    for $t = 1, 2, \cdots, T$ do
        Sample action from the policy distribution $a_l \sim \pi_\phi(a_l|s_l)$;
        Calculate Eq. (14) to construct the empirical distribution $\mathbb{P}_i^0$;
        Construct the uncertainty set $D_i$ around the empirical distribution using Eq. (15);
        Choose $\mathbb{P}_i$ from the uncertainty set $D_i$;
        Execute action $a_l$, obtain reward $r_l$, observe next state $s_{l+1}$;
        Push transition $(s_l, a_l, r_l, s_{l+1})$ into the replay buffer $\boldsymbol{R}$;
        Randomly sample a batch of experiences from $\boldsymbol{R}$;
        Optimize critic networks by minimizing the Eq. (31): $\theta_k \leftarrow \nabla_{\theta_k} J_Q(\theta_k), k \in \{1, 2\}$;
        Update policy network using the Eq. (32): $\phi \leftarrow \nabla_\phi J_\pi(\phi)$;
        Update target network parameters: $\overline{\theta}_k \leftarrow \varsigma\theta_k + (1 - \varsigma)\overline{\theta}_k, k \in \{1, 2\}$, where $\varsigma$ is the soft update parameter.
```

- **PPO (Proximal Policy Optimization):** Our PPO baseline utilizes stochastic policy optimization with Gaussian action distributions. The policy network generates mean actions while exploration is controlled through fixed variance parameters (Shang et al. 2023; Peng et al. 2022).
- **DDPG (Deep Deterministic Policy Gradient):** The DDPG algorithm incorporates deep Q-learning with deterministic policy gradients, employing separate actor-critic networks with target networks for improved learning stability in complex environments (Hazarika et al. 2022).

The simulation environment is configured as follows. A square region measuring 800 meters per side serves as the service area for UAV operations. User equipment is randomly and uniformly distributed throughout this area, while UAV initial positions are randomly assigned within the coordinate range $x, y \in [0, 800]$ meters. Following conventional simulation practices, we utilize the parameter values listed in Table 1 for all experiments unless explicitly stated otherwise.

Fig. 3 compares the convergence of DRO-SAC with PPO and DDPG. All methods improve during training. DDPG rises quickly at the start but settles at lower final rewards and shows large oscillations. Although PPO achieves a performance close to that of DRO-SAC, it converges slower and shows higher volatility. Overall, DRO-SAC converges earlier and achieves higher and more stable rewards, benefiting from its distributionally robust design under environmental uncertainties.

U_{tip}	$120\ \text{m/s}$	P_1	59.03 W
P_2	79.07 W	V_0	$3.6\ \text{m/s}$
A_0	m/s^2	$D_{\min}$	3.5 Mb
$D_{\max}$	4.5 Mb	J	5
$v_{\max}$	$35\ \text{m/s}$	I	20
B	10 MHz	$f_u^{\max}$	10 GHz
$f_i^{\max}$	1 GHz	σ^2	-85 dBm
H	150 m	d_{dim}	3 m
$p_{i,\max}$	0.5 W	$f_u^{\max}$	10 GHz

Table 1: Experimental parameters

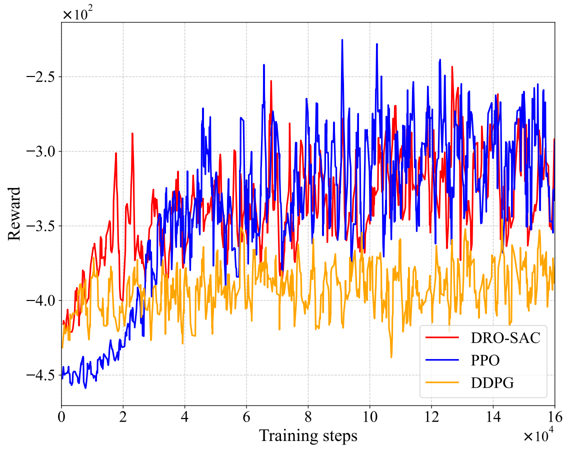

Figure 3: Performance comparison among DRO-SAC and two baseline algorithms.

Fig. 4 evaluates the total energy consumption under different numbers of UEs and position errors. The consumption increases with position error, as larger errors demand more communication and computation. Although growing UEs intensify fluctuations, the overall variation remains small. This demonstrates the scheme's robustness in maintaining stable performance under challenging network conditions.

Fig. 5 shows the effect of task size and bandwidth on weighted energy consumption. Energy consumption decreases with bandwidth due to faster transmission rates, which shortens transmission time. Conversely, it increases with task size because more computation and transmission resources are required.

The trajectories in Fig. 6 demonstrate that UAVs fly to the UEs they serve. Furthermore, they adapt their positions in response to the UE distribution. The results demonstrate how the policy assists UAVs in locating a balanced service area and gradually adjusting their positions to minimize energy consumption during flight.

Conclusion

This paper introduced a robust computation offloading scheme for multi-UAV MEC networks that explicitly handles uncertainties in task data sizes and UAV movements. We formulated a joint optimization problem for energy consumption, trajectories, and offloading, and proposed the

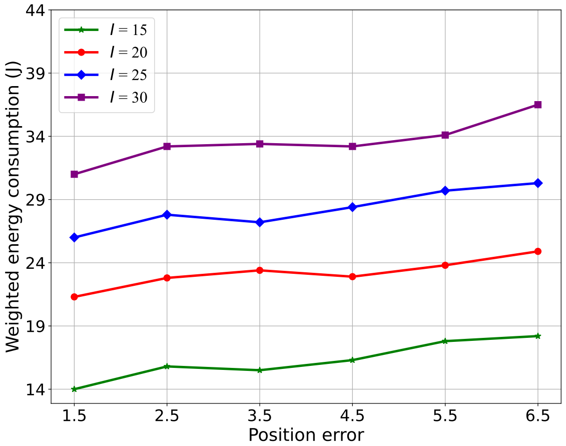

Figure 4: Weighted energy consumption versus position error for different numbers of UEs.

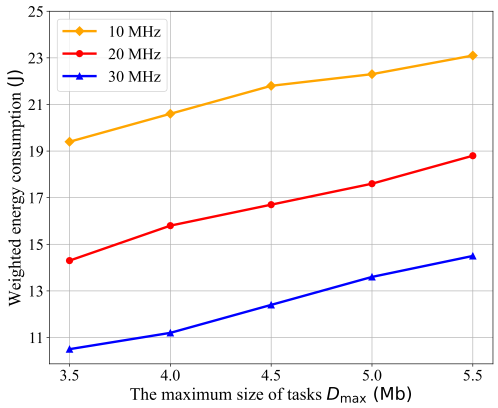

Figure 5: Performance versus task size and bandwidth.

DRO-SAC framework, which integrates DRO with SAC to learn policies that minimize worst-case energy consumption. Extensive experiments demonstrated the scheme's superiority over benchmarks in minimizing energy use while ensuring reliable task completion in dynamic environments.

References

Chen, J.; Cao, X.; Yang, P.; Xiao, M.; Ren, S.; Zhao, Z.; and Wu, D. O. 2023. Deep Reinforcement Learning Based Resource Allocation in Multi-UAV-Aided MEC Networks. *IEEE Transactions on Communications*, 71(1): 296–309.

Hazarika, B.; Singh, K.; Biswas, S.; and Li, C.-P. 2022. DRL-Based Resource Allocation for Computation Offloading in IoV Networks. *IEEE Transactions on Industrial Informatics*, 18(11): 8027–8038.

Hua, M.; Yang, L.; Wu, Q.; Pan, C.; Li, C.; and Swindlehurst, A. L. 2021. UAV-Assisted Intelligent Reflecting Surface Symbiotic Radio System. *IEEE Transactions on Wireless Communications*, 20(9): 5769–5785.

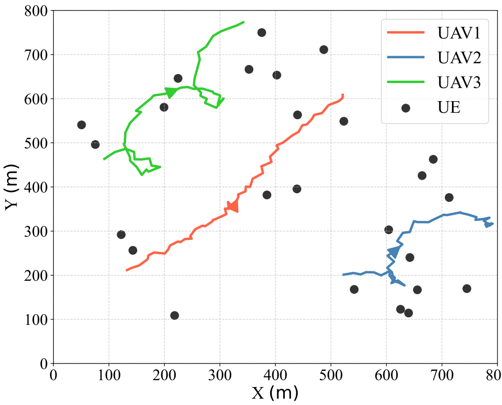

Figure 6: The example of UAV trajectories with $I = 25$ and $J = 3$.

Jia, Z.; Cui, C.; Dong, C.; Wu, Q.; Ling, Z.; Niyato, D.; and Han, Z. 2025. Distributionally Robust Optimization for Aerial Multi-Access Edge Computing via Cooperation of UAVs and HAPs. *IEEE Transactions on Mobile Computing*, 24(10): 10853–10867.

Liu, B.; Wan, Y.; Zhou, F.; Wu, Q.; and Hu, R. Q. 2022. Resource Allocation and Trajectory Design for MISO UAV-Assisted MEC Networks. *IEEE Transactions on Vehicular Technology*, 71(5): 4933–4948.

Lu, D.; Qu, Y.; Wu, F.; Dai, H.; Dong, C.; and Chen, G. 2020. Robust Server Placement for Edge Computing. In *Proc. IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 285–294. New Orleans, USA. ISBN 978-1-7281-6876-0.

Peng, B.; Xie, Y.; Seco-Granados, G.; Wymeersch, H.; and Jorswieck, E. A. 2022. Communication Scheduling by Deep Reinforcement Learning for Remote Traffic State Estimation With Bayesian Inference. *IEEE Transactions on Vehicular Technology*, 71(4): 4287–4300.

Shang, C.; Sun, Y.; Luo, H.; and Guizani, M. 2023. Computation Offloading and Resource Allocation in NOMA–MEC: A Deep Reinforcement Learning Approach. *IEEE Internet of Things Journal*, 10(17): 15464–15476.

Ye, N.; Chen, L.; Ouyang, Q.; and An, J. 2023. Time-Efficient Data Download for Emergency UAV: Joint Optimization of On-Board Computation and Communication Under Energy Constraint. *IEEE Transactions on Vehicular Technology*, 72(10): 13718–13722.

Zhang, X.; Chang, Z.; Hämäläinen, T.; and Min, G. 2024. AoI-Energy Tradeoff for Data Collection in UAV-Assisted Wireless Networks. *IEEE Transactions on Communications*, 72(3): 1849–1861.

Zhao, Y.; Liu, C.; Hu, X.; He, J.; Peng, M.; Wing Kwan Ng, D.; and Quek, T. Q. S. 2025. Joint Content Caching, Service Placement, and Task Offloading in UAV-Enabled Mobile Edge Computing Networks. *IEEE Journal on Selected Areas in Communications*, 43(1): 51–63.