



OPEN

Efficient dynamic task offloading and resource allocation in UAV-assisted MEC for large sport event

Chen Peng^{1,3}✉, Qiqi Wang^{2,3} & Desheng Zhang¹

With the rapid development of Internet of Things (IoT) technology, the people's demand for the viewing experience of large-scale sport events is increasing. However, due to the significant concentration of time and space in large-scale sports events, which leads to a surge in computation-intensive tasks, making traditional network models difficult to cope with such high demands. Fortunately, with the advantages of flexible deployment of Unmanned Aerial Vehicles (UAVs), UAV-assisted edge computing technology provides an innovative solution. This paper studies the resource allocation problem in UAV-assisted edge computing system for large-scale sport events. Our goal is to minimize system energy consumption while satisfying system performance. We formulate the problem as a long-term stochastic optimization problem. To address this issue, we propose the efficient dynamic resource allocation (EDRA) algorithm. By employing stochastic optimization techniques, the original problem is decomposed into multiple sub-problems that can be solved in parallel. We solve each subproblem through convex optimization and linear programming. Through theoretical analysis, we prove the gap between the proposed solution and the optimal solution. Experiments shows that the EDRA algorithm can reduce energy consumption by 32.4% compared to advanced algorithms while ensuring stronger system performance.

With the rapid development of the Internet of Things (IoT) technology, edge computing, as a new computing paradigm, has attracted extensive attention in recent years. Edge computing significantly reduces data transmission latency, improves data processing efficiency, and reduces network bandwidth consumption by pushing computing resources and application services from the cloud to the edge of the network¹. However, in some scenarios, traditional base stations are difficult to work or provide sufficient computing resources, such as rescue operations, forest fires, disaster scenarios, and temporary large-scale events². Taking large-scale sports events as an example, due to the increasing demand for viewing experience from spectators and the large number of computation-intensive tasks generated during the competition, the traditional network model is difficult to cope with such high demand^{3,4}. Especially during the critical moments of the competition, the demand for real-time data updates, high-definition video streaming and other services increases sharply, which undoubtedly poses a huge challenge to the existing computing and network resources⁵.

Unmanned aerial vehicle (UAV)-assisted edge computing technology provides an innovative solution⁶. UAVs play an increasingly important role in edge computing systems due to their flexible deployment, high maneuverability, and extensive coverage. Deployed in the air, UAVs can provide better channel conditions for ground devices. Moreover, the flexible deployment capabilities of UAVs enable them to provide high-quality short-term services for temporary high-density scenarios^{7,8}. As edge nodes, UAVs can provide fast and reliable computing and communication support for large-scale sports events, thus meeting the spectators' demands for high-quality viewing experience⁹⁻¹¹.

Although previous efforts have conducted some research on UAV-assisted edge computing in large-scale events¹², there are still many challenges to face. First, numerous computation-intensive tasks are generated suddenly, and it is difficult to accurately predict the generation time of these tasks, and the communication conditions in the environment are also random, making it challenging to optimize the system without prior information. Second, the computing and communication resources of UAVs and devices are limited, requiring us to allocate these resources reasonably while ensuring system performance and cost. Finally, while the addition of UAVs can provide better communication conditions for the system, it also increases the complexity of the problem. Therefore, ensuring system performance while minimizing system energy consumption in this UAV-assisted MEC system is challenging.

¹Department of Journalism and Communication, Wuhan Sports University, Wuhan, China. ²Department of Physical Education, Hubei University, Wuhan, China. ³Chen Peng and Qiqi Wang these have contributed equally to this work. email: 202142002@whsu.edu.cn

This paper studies the task offloading and resource allocation problem in a three-tier UAV-assisted MEC system. We aim to minimize the energy consumption of the system while satisfying multiple constraints such as computing and communication resources. The CPU frequency and offloading power of devices and UAVs are considered as our decision variables. Since the problem is NP-hard¹³, we utilize mathematical methods to transform the problem and further decouple it into four subproblems. We propose the EDRA algorithm, which can stabilize the task queue of the system while minimizing system energy consumption as much as possible. Theoretical analysis proves the effectiveness of our algorithm. A series of experiments have been conducted to demonstrate that EDRA has excellent performance in solving the task offloading and resource allocation problem in UAV-assisted MEC systems.

Our main contributions are as follows:

1. We study the task offloading problem in a scenario that includes multiple devices, UAVs with edge servers, and a cloud server, while taking into account the randomness of tasks, channel conditions, and user locations. Under the premise of satisfying long-term constraints and system performance, we reformulate the problem as a long-term stochastic optimization problem to minimize system energy consumption.
2. As the problem is NP-hard, we transform it into a deterministic optimization problem through a series of mathematical manipulations. Combining rigorous proofs, we decompose the problem into multiple subproblems, each containing only one decision variable, which can achieve (approximate) optimal solutions. The efficient dynamic resource allocation (EDRA) algorithm is proposed to solve multiple subproblems online and in parallel.
3. Through theoretical analysis, we prove the gap between the proposed algorithm and the optimal algorithm, ensuring the performance of the proposed algorithm. Through a series of parameter analyses, we demonstrate that the proposed algorithm can adapt to different task arrival rates and device quantities. Through comparative experiments, we show that the system can effectively reduce energy consumption while ensuring the task queue length.

The structure of the remainder of this paper is as follows. Section [Related work](#) introduces related work. Section [System model and problem formulation](#) presents the system model and problem formulation. Section [EDRA for task offloading and resource allocation](#) transforms and decomposes the problem into multiple subproblems, and propose the EDRA algorithm. Section [Algorithm analysis for EDRA](#) provides a theoretical analysis of the EDRA algorithm. Section [Evaluation](#) demonstrates the performance of the EDRA algorithm in different environments through experimental results and compares it with advanced algorithms. Finally, Sect. [Conclusion](#) summarizes the paper.

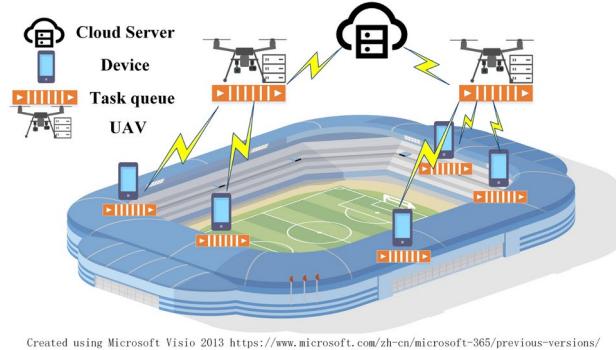
Related work

Recently, many efforts have focused on task offloading and resource allocation problem in the MEC system. Ding et al.¹⁴ studied a generalized MEC system that incorporates both OMA and NOMA transmission methods. They proposed a multi-objective optimization problem to achieve minimal system energy consumption while achieving low-complexity resource allocation. Ernest et al.¹⁵ applied multi-agent reinforcement learning techniques to the computation offloading problem in MEC vehicular networks, obtaining an offloading strategy that can achieve maximum energy efficiency. Chu et al.¹⁶ studied the caching and task offloading issues in multi-access edge computing networks. They focused on maximizing user QoE by consolidating a logical resource pool, and proposed a method based on approximation and decomposition theory to solve this problem. Tang et al.¹⁷ investigated the offloading problem in multi-access edge computing. They achieved minimized offloading delay for IoT terminals by jointly optimizing computing and communication resources. However, the aforementioned studies did not take into account the randomness of channel conditions and the unpredictability of task arrivals in the environment.

Considering the dynamic variability of the environment, Liu et al.¹⁸ designed a dynamic optimization algorithm for intensive requests from terminal devices in MEC systems. This algorithm can significantly improve system performance by adjusting the offloading decisions and power allocation strategies. Ali et al.¹⁹ studied a multi-user MEC system, where they optimized the offloading decisions from a life-cycle perspective without prior information and proposed a SAC-based lifespan maximization algorithm. Li et al.²⁰ investigated the problem of minimizing computational delay and energy consumption in MEC systems. They utilized the Lyapunov optimization framework to optimize the system while balancing both factors. Sun et al.²¹ studied the long-term energy-efficient task allocation and offloading in MEC systems. By decomposing the original problem into multiple subproblems, they achieved an online energy-saving strategy. However, the aforementioned works did not consider utilizing UAVs to assist edge computing, thus unable to provide edge computing for scenarios requiring temporary, intensive, and low-latency computations.

Sun et al.²² formulated the Time and Energy Minimization Communication Multi Objective Optimization Problem (TECMOP) for CB in UAV networks and proposed an improved Multi Objective Ant Lion Optimization (IMOLO) algorithm to solve the problem. Picano et al.²³ studied the computational offloading problem in UAV-assisted MEC systems. They proposed an efficient offloading decision algorithm based on a matching algorithm while satisfying different QoS constraints. Chen et al.²⁴ investigated a LEO Satellite-Terrestrial Edge Computing Systems with terrestrial base stations. They considered task offloading under limited resources and proposed an offloading algorithm based on potential game theory. Gao et al.²⁵ studied the low-latency QoE problem in UAV-MEC systems. They proposed a service experience-oriented framework to cache user-required content on the edge servers on UAVs, thus improving the overall system performance. Chen et al.²⁶ studied the scenario of using UAV to assist users in edge computing when the ground base station cannot

Key factors	Papers										
	[14]	[15]	[16]	[17]	[18]	[19]	[21]	[22]	[25]	[26]	Ours
Consideration of random information	No	Yes	No	No	Yes	Yes	Yes	No	Yes	No	Yes
UAV Scenario	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes
System stability assurance	No	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Support for distributed solving	No	No	Yes	No	No	No	No	Yes	Yes	No	Yes

Table 1. Comparison of related work based on key factors.**Fig. 1.** An example of a system model comprising multiple UAVs equipped with edge servers, multiple devices, and a cloud server.

work. A hybrid natural inspired optimization algorithm (HNIO) and its discrete optimization version were proposed to solve the energy consumption optimization problem in this scenario.

Despite significant efforts in UAV-assisted edge computing, existing works have not fully addressed the challenges of random environmental information and system stability while providing an online solution. As shown in Table 1, our work takes these issues into consideration. Compared to robust optimization and RL^[14,15,18,19], we balance robustness and efficiency through the dynamic transformation of the stochastic optimization framework, explicitly modeling the stochastic optimization problem. In contrast to the above works' approaches to ensuring system performance^[15,17,22,27], we dynamically guarantee long-term system performance by introducing task queues.

Specifically, we formulate the Task Offloading and Resource Allocation in UAV-Assisted MEC for Large Sport Events as a stochastic optimization problem. By leveraging stochastic optimization techniques, we can provide adaptive solutions online without requiring prior knowledge of task arrival rates and channel conditions. Additionally, we ensure system stability by introducing virtual task queues and incorporating them into the optimization objective, thereby transforming the originally constrained virtual task queues into part of our optimization goal.

System model and problem formulation

System model

As shown in Fig. 1, this paper considers a three-tier UAV-assisted edge computing network, which includes I terminal devices, J UAVs equipped with edge servers, and a Cloud Server (CS) that can provide cloud services. To provide long-term optimization, time is divided into T equal-length discrete time slots, with each slot having a duration of τ , where $\mathcal{T} \in \{1, 2, 3, \dots, T\}$. For convenience, $N_{i,j}$ is used to represent the i -th device served by UAV j and \mathcal{I}_j represents the set of devices which communicating with UAV j . In each time slot, there are $A_{j,i}(t)$ amount of tasks arriving at device $N_{i,j}$, which is influenced by factors such as whether, position movement, etc., and the device can choose to process the tasks locally or offload them to the edge server on UAV j . The edge server can select to further transmit part of the tasks to the CS for processing based on its own task backlog. Table 2 shows the primary notations and definitions for this paper.

Communication model

UAVs equipped with edge servers provide better communication conditions compared to traditional ground base stations. This is due to the higher probability of Line-of-Sight (LoS) path communication in ground-to-air communication. Therefore, this paper consider a probabilistic geometric LoS model to account for the channel gain between devices and UAVs in large-scale sports events.

We use H_j to represent the flight height of UAV j , and for simplicity, the height of all devices is assumed to be 0. The two-dimensional coordinates of UAV j are denoted as $(x_j(t), y_j(t))$, and the two-dimensional coordinates of device $N_{i,j}$ are represented as $(x_{i,j}(t), y_{i,j}(t))$. The probabilistic geometric LoS channel model

Notion	Definition
I	Number of terminal devices
J	Number of UAVs equipped with edge servers
T	Number of time slots
τ	Duration of each time slot
$N_{i,j}$	Device i served by UAV j
$A_{j,i}$	Amount of tasks arriving at device $N_{i,j}$
$f_{i,j}(t)$	CPU frequency of device $N_{i,j}$ at time t
f_{\max}^l	Maximum CPU frequency of the devices
$P_{i,j}(t)$	Transmission power of device $N_{i,j}$ at time t
P_{\max}^l	Maximum transmission power of the devices
$f_j(t)$	CPU frequency of UAV j at time t
$P_j(t)$	Transmission power of UAV j at time t
f_{\max}^j	Maximum CPU frequency of the UAVs
P_{\max}^j	Maximum transmission power of the UAVs
γ_c	Energy efficiency factor
$Q_{i,j}(t)$	Queue backlog of device $N_{i,j}$ at time t
$U_{i,j}(t)$	Queue backlog of device $N_{i,j}$ on UAV j at time t
$R_{j,i}(t)$	Transmission rate for offloading tasks from device $N_{i,j}$ to UAV j
$g_{j,i}$	Channel gain between UAV j and device $N_{i,j}$
B_j	Bandwidth allocated for communication between device $N_{i,j}$ and UAV j
$N_0(t)$	Noise power spectral density
$\sigma_{i,j}$	Number of CPU cycles required to process one bit of task on device $N_{i,j}$
σ_j	Number of CPU cycles required to process one bit of task on UAV j

Table 2. Notations and definitions.

depends on the angle between the communication entities. We can calculate the angle between UAV j and device $(x_{i,j}, y_{i,j})$ in the t -th time slot based on the horizontal difference as:

$$\theta_{i,j}(t) = \frac{180}{\pi} \arctan\left(\frac{H_j}{\|p_j(t) - p_{i,j}(t)\|}\right). \quad (1)$$

After obtaining the UAV j and the device's coordinates $(x_{i,j}, y_{i,j})$, we can derive the probabilistic geometric LoS (Line-of-Sight) loss model²⁸ between them,

$$P(LoS, \theta_{i,j}(t)) = \frac{1}{1 + a \exp(-b(\theta_{i,j}(t) - a))}, \quad (2)$$

Where a and b are environment-dependent parameters. Additionally, the probability of NLoS link can be easily derived as $P(NLoS, \theta_{i,j}(t)) = 1 - P(LoS, \theta_{i,j}(t))$. Moreover, during the transmission of wireless signals, there is also the free-space path loss given by:

$$FL_{j,i}(t) = 20 \log_{10} \|p_j(t) - p_{i,j}(t)\| + 20 \log_{10} \left(\frac{4\pi f_c}{c} \right), \quad (3)$$

Where this formula incorporates the path loss due to distance, related to the Euclidean distance between the UAV and the device, and the path loss caused by the diffusion of electromagnetic waves in free space. Here, f_c is the frequency, and c represents the speed of light.

In summary, similar to^{29,30}, the path loss between the UAV and the device can be expressed as:

$$L_{j,i}(t) = P(LoS, \theta_{i,j}(t)) \times \eta^{LoS} + P(NLoS, \theta_{i,j}(t)) \times \eta^{NLoS} + FL_{j,i}(t), \quad (4)$$

Where η^{LoS} and η^{NLoS} are the additional path loss factors for LoS and NLoS conditions, respectively. Then, the channel gain is given by $g_{j,i} = 10^{-\frac{L_{j,i}(t)}{10}}$.

Similar to³¹, we assume that the devices communicate with the UAVs using Frequency Division Multiple Access (FDMA) transmission. According to Shannon's theorem³², the transmission rate in the ideal condition for a device to offload tasks to the UAV can be expressed as:

$$R_{j,i}(t) = B_j \log_2 \left(1 + \frac{g_{j,i}(t) P_{i,j}(t)}{N_0(t)} \right), \quad (5)$$

Where B_j is the bandwidth allocated for communication between device $N_{i,j}$ and UAV j , $P_{i,j}(t)$ is the transmit power of device $N_{i,j}$, and $N_0(t)$ is the noise power spectral density.

Task and offloading model

The system is optimized for long-term performance and divided into multiple time slots. In each time slot, tasks arrive at device $N_{i,j}$ following a Discrete-Time Poisson Process, with the task arrival rate denoted by $A_{i,j}(t)$. Upon receiving tasks³³, the device can choose to process them locally or offload them to the edge server on the UAV. Unlike some binary offloading models, this work considers scaling the device's CPU frequency $f_{i,j}(t)$ and adjusting the offload power $P_{i,j}(t)$ for more precise partial offloading. Similarly, the UAV can scale its CPU frequency $f_j(t)$ and offload power $P_j(t)$ to partially process tasks and offload some to the cloud. This section models the task amount and energy consumption in the system.

Energy consumption and task amount of devices

The local computation amount $C_{i,j}^l(t)$ of the device is related to its hardware characteristics and CPU frequency, while the offloaded task amount $C_{i,j}^o(t)$ can be changed by adjusting the offload power to alter the transmission rate. They can be expressed as follows:

$$C_{i,j}^l(t) = \frac{f_{i,j}(t)\tau}{\sigma_{i,j}}, \quad C_{i,j}^o(t) = \tau R_{i,j}(t). \quad (6)$$

Where $\sigma_{i,j}$ is the number of CPU cycles required by device i to process one bit of task. It is also necessary to satisfy $C_{i,j}^l(t) + C_{i,j}^o(t) \leq A_{i,j}(t)$.

The energy consumption of the device is also related to the CPU frequency and offload power^{34,35}. According to Joule's Law, it can be expressed as:

$$E_{i,j}(t) = \gamma_c f_{i,j}^3(t)\tau + \tau P_{i,j}(t), \quad (7)$$

Where γ_c is the energy efficiency factor. The $f_{i,j}(t)$ and $P_{i,j}(t)$ are subject to the following constraints:

$$0 \leq f_{i,j}(t) \leq f_{max}^l, \quad 0 \leq P_{i,j}(t) \leq P_{max}^l. \quad (8)$$

Energy consumption and task amount of UAVs

Each UAV's server can provide services to multiple devices, resulting in the UAV's CPU being occupied by multiple different tasks. We logically divide the CPU frequency of the UAV into $f_j^j(t)$ to represent the CPU frequency allocated by UAV j to device $N_{i,j}$ in time slot t . The same applies to the offload power. Then, the task amount processed by the UAV can be expressed as:

$$D_i^{l,j}(t) = \frac{f_j^j(t)\tau}{\sigma_j}, \quad D_i^{o,j}(t) = \tau R_j^j(t), \quad (9)$$

Where σ_j is the number of CPU cycles required for device i to process one bit of task. $D_i^{l,j}(t)$ represents the amount of tasks processed locally, and $D_i^{o,j}(t)$ represents the amount of tasks offloaded to the cloud server for processing. The energy consumption can be expressed as:

$$E_j(t) = \sum_{i=0}^{\mathcal{I}} \gamma_c f_i^j(t)^3 \tau + \tau P_j^j(t). \quad (10)$$

Unlike devices that are only constrained by themselves, all tasks on a UAV share the computing and communication resources of that UAV, affecting each other. Task processing and offloading on the UAV need to satisfy the following constraints:

$$0 \leq \sum_{i=1}^{\mathcal{I}_j} f_i^j(t) \leq f_{max}^l, \quad 0 \leq \sum_{i=1}^{\mathcal{I}_j} P_i^j(t) \leq P_{max}^j, \quad \forall j \in \mathcal{J}. \quad (11)$$

Task queue model

To achieve long-term optimization of the system, we need to ensure that the system is in a state where it can process tasks normally. That is, the backlog of tasks processed by the system cannot grow indefinitely. To this end, we set up two sets of queues (they are all in a First-In-First-Out (FIFO) manner and processed one by one) to ensure that the task backlog on devices and UAVs is not excessive (the cloud service has powerful computing capabilities, so the task backlog at the cloud level is not considered for the moment). First, the queue on the device can be expressed as:

$$Q_{i,j}(t+1) = \max\{Q_{i,j}(t) + A_{i,j}(t) - C_{i,j}^l(t) - C_{i,j}^o(t), 0\} \quad \forall t \in \mathcal{T}, \quad (12)$$

Where $Q_{i,j}(t)$ represents the task backlog status of device $N_{i,j}$ at time t , the queue entry variable is the task arrival rate $A_{i,j}(t)$, and the exit variables are the local computation amount $C_{i,j}^l(t)$ and the offloaded task amount $C_{i,j}^o(t)$.

We use $U_{i,j}$ to represent the backlog queue for the tasks of device i on UAV j , as follows:

$$U_{i,j}(t+1) = \max\{U_{i,j}(t) + C_{i,j}^o(t) - D_{i,j}^l(t) - D_{i,j}^o(t), 0\} \quad \forall t \in \mathcal{T}, \quad (13)$$

Where the entry variable is the offloaded task amount $C_{i,j}^o(t)$ from device i , and the exit variables are the local computation amount and the offloaded task amount.

Problem formulation

Given the energy consumption formula for task processing in the system, considering our optimization objective. We minimize the energy consumption of the system by optimizing the CPU frequency, offload power of devices, CPU frequency, and offload power of UAVs, while satisfying multiple constraints. This is a long-term stochastic optimization problem and can be expressed as:

$$\text{P1: } \min_{\phi_{i,j}(t)} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^{I_j} E_{i,j}(t) + E_j(t), \quad (14a)$$

$$\text{s.t. } \lim_{T \rightarrow \infty} \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^{I_j} \mathbb{E}\{Q_{i,j}(t)\} < \infty, \quad (14b)$$

$$\lim_{T \rightarrow \infty} \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^{I_j} \mathbb{E}\{U_{i,j}(t)\} < \infty, \quad (14c)$$

$$C_{i,j}^l(t) + C_{i,j}^o(t) \leq A_{i,j}(t), \forall i, j, t \quad (14d)$$

$$D_{i,j}^l(t) + D_{i,j}^o(t) \leq C_{i,j}^o(t), \forall i, j, t, \quad (14e)$$

(8), and (9).

Where $\phi_{i,j}(t) = f_{i,j}(t), P_{i,j}(t), f_i^j(t), P_i^j(t)$ is a set of vector variables. $E_{i,j}(t)$ is the energy consumption of device $N_{i,j}$ at time slot t , $E_j(t)$ represents the energy consumption of UAV j , and (14a) is our optimization goal, which is to minimize the long-term energy consumption of the devices and UAVs. (14b) and (14c) represent the long-term constraints that need to be satisfied by the device layer and the UAV layer respectively. (14d) represents the task processing capacity constraint of the devices. (14e) represents the task processing capacity constraint of the UAVs. (8) and (9) limit the CPU frequency and power of the devices and UAVs respectively.

In fact, solving this long-term stochastic optimization problem is quite challenging. Since the task arrival rate, channel conditions in the environment, and other factors are random, it is difficult to make precise predictions. Mathematically, this problem is formulated as a mixed-integer nonlinear programming (MINLP) problem, which is known to be NP-hard. Traditional mathematical methods find it difficult to provide an optimal solution within a reasonable time. Therefore, in the next section, we use stochastic optimization techniques to transform the problem into a more tractable static optimization problem.

EDRA for task offloading and resource allocation

Recall our goal of minimizing the system's energy consumption by optimizing $\phi_{i,j}(t)$. However, in P1, multiple variables in $\phi_{i,j}(t)$ are coupled high-dimensional vectors. In this section, we will first transform the direct optimization of P1 into optimizing the upper bound of P1, aiming to reduce long-term energy consumption by decreasing the upper bound of P1. Additionally, we decouple the multiple variables in $\phi_{i,j}(t)$ and decompose them into multiple independent subproblems. This also means that these problems can be solved in parallel, improving the efficiency of the solution process.

Problem transform

First, we use the vector $\omega(t) = \{Q_{1,1}(t), \dots, Q_{i,j}(t)\} \cup \{U_{1,1}(t), \dots, U_{i,j}(t)\}$ to represent the overall queue backlog and stability conditions of the system at time t and $L(\omega(t)) = \frac{1}{2} \sum_{j=1}^J \sum_{i=1}^{I_j} (Q_{i,j}^2(t) + U_{i,j}^2(t))$. Clearly, $L(\omega(t))$ is non-negative, and we can judge the backlog situation of the system based on its magnitude. To achieve long-term stability, we can express the drift function as:

$$\Delta(\omega(t)) = \mathbb{E}\{L(\omega(t+1)) - L(\omega(t)|\omega(t)\}. \quad (15)$$

Reducing this drift function is equivalent to reducing the backlog in the queue. However, while we stabilize the backlog in the system, we also need to minimize energy consumption as much as possible. In this case, we introduce a penalty term into the drift function to obtain the drift-plus-penalty function in order to achieve this goal.

$$D(\omega(t)) = \Delta(\omega(t)) + V\mathbb{E}\{E_{total}(t)|\omega(t)\} \quad (16)$$

Where $E_{total}(t)$ represents the total energy consumption of the system at time t . V is a trade-off factor used to adjust the optimization focus between the queue and energy consumption. After constructing this drift-plus-penalty function, we can optimize the overall energy consumption of the system by optimizing the upper bound of this function, while also keeping the system in a stable state. The upper bound of $D(\omega(t))$ can be obtained by Theorem 1.

Theorem 1 *If there is an upper limit A_{max} for the task arrival rate $A_{i,j}(t)$, then the following relationship holds:*

$$\begin{aligned} & \Delta(\omega(t)) + V\mathbb{E}\{E_{total}(t)|\omega(t)\} \\ & \leq Z + V\mathbb{E}\{E_{total}(t)|\omega(t)\} \\ & + \sum_{j=1}^J \sum_{i=1}^{I_j} Q_{i,j}(t) \{A_{i,j}(t) - C_{i,j}^l(t) - C_{i,j}^o(t)|\omega(t)\} \\ & + \sum_{j=1}^J \sum_{i=1}^{I_j} U_{i,j}(t) \{C_{i,j}^o(t) - D_{i,j}^l(t) - D_{i,j}^o(t)|\omega(t)\} \end{aligned} \quad (17)$$

Where Z is a constant and

$$Z = \sum_{j=1}^J \sum_{i=1}^{I_j} [A_{i,j}^2(t) + C_{i,j}^l(t) + 2C_{i,j}^o(t)C_{i,j}^l(t) + 2C_{i,j}^o(t)^2 + (D_{i,j}^l(t) + D_{i,j}^o(t))^2].$$

Proof Squaring both sides of the formula (17) can yield:

$$Q_{i,j}^2(t+1) \leq \{Q_{i,j}(t) + A_{i,j}(t) - C_{i,j}^l(t) - C_{i,j}^o(t)\}^2. \quad (18)$$

Expanding the right side of the formula gives:

$$Q_{i,j}^2(t+1) \leq Q_{i,j}^2(t) + A_{i,j}^2(t) + (C_{i,j}^l(t) + C_{i,j}^o(t))^2 - A_{i,j}(t)[C_{i,j}^l(t) + C_{i,j}^o(t)] + 2Q_{i,j}(t)[A_{i,j}(t) - C_{i,j}^l(t) - C_{i,j}^o(t)]. \quad (19)$$

Extracting the part with upper limit and setting $Z_1 = (A_{i,j}^{max}(t))^2 + (C_{i,j}^{l,max}(t) + C_{i,j}^{o,max}(t))^2$, and subtracting $Q_{i,j}^2(t)$ from both sides, we can get:

$$Q_{i,j}^2(t+1) - Q_{i,j}^2(t) \leq +Z_1 + 2Q_{i,j}(t)[A_{i,j}(t) - C_{i,j}^l(t) - C_{i,j}^o(t)] - A_{i,j}(t)[C_{i,j}^l(t) + C_{i,j}^o(t)]. \quad (20)$$

Since $A_{i,j}(t)[C_{i,j}^l(t) + C_{i,j}^o(t)]$ is always greater than or equal to 0, we can obtain the following inequality through scaling equations:

$$Q_{i,j}^2(t+1) - Q_{i,j}^2(t) \leq +Z_1 + 2Q_{i,j}(t)[A_{i,j}(t) - C_{i,j}^l(t) - C_{i,j}^o(t)]. \quad (21)$$

Similarly, we can get:

$$U_{i,j}^2(t+1) - U_{i,j}^2(t) \leq +Z_2 + 2U_{i,j}(t)[C_{i,j}^o(t) - D_{i,j}^l(t) - D_{i,j}^o(t)], \quad (22)$$

Where $Z_2 = (C_{i,j}^{o,max}(t))^2 + (D_{i,j}^{l,max}(t) + D_{i,j}^{o,max}(t))^2$.

Combining the (20) and (21), the Theorem 1 is proven.

By substituting (7) and (10) into (17), we can obtain P2. According to Theorem 1, we can transform the original optimization of P1 into the optimization of P2. For brevity, we have simplified the summation and expectation of P2, and represent P2 as follows:

$$\begin{aligned} \text{P2: } & \min_{\phi_{i,j}(t)} V\tau[\gamma_c f_{i,j}^3(t) + P_{i,j}(t) + \gamma_c f_i^j(t)^3 + P_i^j(t)] \\ & + Q_{i,j}(t)[A_{i,j}(t) - \frac{f_{i,j}(t)\tau}{\sigma_{i,j}} - \tau R_{i,j}(t)] \\ & + U_{i,j}(t)[\tau R_{i,j}(t) - \frac{f_i^j(t)\tau}{\sigma_j} - \tau R_i^j(t)] \end{aligned} \quad (23)$$

Decomposing subproblem

After proving that the drift function with penalty term has an upper bound, we can observe that its upper bound is a multivariate function containing four decision variables. Fortunately, after transforming it into an upper bound problem, the originally coupled four decision variables become separable. In this section, we decompose the original problem into four subproblems and solve them separately using different mathematical methods.

CPU frequency of the device subproblem

We extract the part related to CPU frequency from the upper bound problem and decouple the problem to obtain the following subproblem:

$$\begin{aligned} \text{P2-1:} \quad & \min_{f_{i,j}(t)} -Q_{i,j}(t) \frac{f_{i,j}(t)\tau}{\sigma_{i,j}} + V\gamma_c f_{i,j}^3(t)\tau, \\ \text{s.t.} \quad & 0 \leq f_{i,j}(t) \leq \min\{f_{max}^l, \frac{Q_{i,j}(t)\sigma_{i,j}}{\tau}\}. \end{aligned} \quad (24)$$

The CPU frequency determines the task execution time on each device and UAV. Higher CPU frequencies reduce task execution time but increase energy consumption due to the cubic relationship between energy consumption and CPU frequency. To optimize the CPU frequency, we formulated it as a convex optimization problem. By taking the second derivative of P2-1, we can obtain the extreme points of the function $f_{i,j}(t) = \sqrt{[Q_{i,j}(t)/3V\gamma_c\sigma_{i,j}]}$. By observing the relationship between the extreme points and the feasible region, we can conclude:

$$f_{i,j}^*(t) = \begin{cases} \sqrt{\frac{Q_{i,j}(t)}{3V\gamma_c\sigma_{i,j}}}, & 0 \leq \sqrt{\frac{Q_{i,j}(t)}{3V\gamma_c\sigma_{i,j}}} \leq f^m \\ f^m, & \text{otherwise,} \end{cases} \quad (25)$$

Where $f^m = \min\{f_{max}^l, \frac{(Q_{i,j}(t)+A_{i,j}(t))\sigma_{i,j}}{\tau}\}$.

Offloading power of the device subproblem

By decoupling the part related to offloading power in P-2, we obtain the following subproblem:

$$\begin{aligned} \text{P2-2:} \quad & \min_{P_{i,j}(t)} -B_j(Q_{i,j}(t) - U_{i,j}(t)) \\ & \log_2(1 + \frac{g_{j,i}(t)P_{i,j}(t)}{N_0(t)}) + V\tau P_{i,j}(t) \\ \text{s.t.} \quad & 0 \leq P_{i,j}(t) \leq P_{max}^l. \end{aligned} \quad (26)$$

This is a nonlinear optimization problem. We denote the function of P2-1 as $F(P_{i,j}(t))$, and introduce $x_{i,j}(t)$ as a substitution for $\frac{g_{j,i}(t)}{N_0(t)}$ and $y_{i,j}(t)$ for $-B_j(Q_{i,j}(t) - U_{i,j}(t))$. Then, the first-order derivative is:

$$\frac{dF(P_{i,j}(t))}{dP_{i,j}(t)} = \frac{-y_{i,j}(t)\tau x_{i,j}(t) + \tau V(1 + x_{i,j}(t)P_{i,j}(t))\ln 2}{(1 + x_{i,j}(t)P_{i,j}(t))\ln 2} \quad (27)$$

And the second-order derivative is:

$$\frac{dF^2(P_{i,j}(t))}{dP_{i,j}(t)^2} = y_{i,j}(t)\tau \frac{x_{i,j}^2(t)}{(1 + x_{i,j}(t)P_{i,j}(t))\ln 2} \quad (28)$$

Analyzing the value of $y_{i,j}(t)$, we have: when $y_{i,j}(t) < 0$, the first-order derivative is always greater than 0. When $y_{i,j}(t) = 0$, $F(P_{i,j}(t))$ is linearly dependent on $P_{i,j}(t)$. When $y_{i,j}(t) > 0$, the second-order derivative is always positive, indicating a minimum occurs when the first-order derivative is 0. This minimum occurs at $P_{i,j}(t) = \frac{y_{i,j}(t)}{V\ln 2} - \frac{1}{x_{i,j}(t)}$. Combining these analyses, we obtain the solution:

$$P_{i,j}^*(t) = \begin{cases} \min\{P_{max}^l, \frac{y_{i,j}(t)g_{i,j}(t)}{V\ln 2} - \frac{N_0(t)}{g_{i,j}(t)}\}, & V \leq Z(t), \\ 0, & \text{otherwise,} \end{cases} \quad (29)$$

Where for simplicity, we use $Z(t)$ to represent $\frac{x_{i,j}(t)y_{i,j}(t)}{\ln 2}$.

CPU frequency of the UAV subproblem

In this subproblem, we find the optimal CPU Frequency of the UAV through solving following subproblem:

$$\begin{aligned} \text{P2-3:} \quad & \min_{f_j^i(t)} -U_{i,j}(t) \frac{f_j^i(t)\tau}{\sigma_j} + V\gamma_c f_j^i(t)^3\tau, \\ \text{s.t.} \quad & 0 \leq \sum_{i=1}^{\mathcal{I}_j} f_i^j(t) \leq f_{max}^l, \end{aligned} \quad (30)$$

The CPU frequency $f_i^j(t)$ of the UAV j needs to be decoupled, but since a UAV has to handle tasks offloaded from multiple devices, it cannot be solved as a simple convex optimization problem. Here, we utilize a combination of the knapsack problem solving approach and convex optimization. Similar to the approach for P3-1, we first find the analytical solution for the CPU frequency allocated to device i :

$$f_j^{i,*}(t) = \begin{cases} \sqrt{\frac{U_{i,j}(t)}{3V\gamma_c\sigma_j}}, & 0 \leq \sqrt{\frac{U_{i,j}(t)}{3V\gamma_c\sigma_i}} \leq f^n \\ f^n, & \text{otherwise,} \end{cases} \quad (31)$$

Where $f^n = \min\{f_{max}^j, \frac{(U_{i,j}(t) + A_{i,j}(t))\sigma_i}{\tau}\}$. After obtaining the analytical solution for $f_j^i(t)$, we treat it as weight of the task of device i and apply the knapsack problem solving method, considering the total CPU frequency of the UAV as the knapsack capacity. The specific details of the algorithm are outlined in Algorithm 1.

Offloading power of the UAV subproblem

In this subproblem, we decouple the offloading power of UAV j to establish the following subproblem?

$$\begin{aligned} \text{P2-4: } \min_{P_{i,j}(t)} \quad & -B_j U_{i,j}(t) \log_2 \left(1 + \frac{g_i^j(t) P_i^j(t)}{N_0(t)} \right) \\ & + V\tau P_i^j(t) \\ \text{s.t. } \quad & 0 \leq \sum_{i=1}^{\mathcal{I}_j} P_i^j(t) \leq P_{max}^j \end{aligned} \quad (32)$$

Input: $Q_{i,j}(t), U_{i,j}(t)$, the trade-off parameter V

Initialization:

$Q_{i,j}(t) = 0, U_{i,j}(t) = 0, g_{i,j}(t) = 0$

End Initialization

for each UAV $j \in \mathcal{J}$ do

 Compute the $g_{i,j}(t)$ according to (1)-(4)

$f_{max} = f_{max}^j, p_{max} = p_{max}^j$

for each device $i \in \mathcal{I}_j$ do

 Substitute the value of (25) into $f_{i,j}(t)$
 Substitute the value of (29) into $P_{i,j}(t)$

 Compute the value of $\sqrt{\frac{U_{i,j}(t)}{3V\gamma_c\sigma_j}}$

 Sort the $f_{i,j}(t)$ and $P_{i,j}(t)$ respectively in ascending order

if $f_{max} > 0$ **then**

 Substitute the value of (31) into $f_i^j(t)$
 $f_{max} = f_i^j(t)$

end

else $f_i^j(t) = 0$;

 Compute the value of $\frac{B_j U_{i,j}(t) g_i^j(t)}{V \ln 2} - \frac{N_0(t)}{g_i^j(t)}$

 Sort the value in ascending order

if $p_{max} > 0$ **then**

 Substitute the value of (33) into p_i^j
 $p_{max} = p_i^j(t)$

end

else $p_i^j(t) = 0$;

 Update the value of $Q_{i,j}(t)$ and $U_{i,j}(t)$

end

end

Output: $f_{i,j}(t), P_{i,j}(t), f_i^j(t)$, and $p_i^j(t)$

Algorithm 1. The Proposed EDRA Algorithm

Similar to solving P2-3, we get an analytical solution firstly for P2-4:

$$P_{i,j}^*(t) = \begin{cases} \min\{P_j^{max}, \frac{B_j U_{i,j}(t) g_i^j(t)}{V \ln 2} - \frac{N_0(t)}{g_j^i(t)}\}, & V \leq z(t), \\ 0, & \text{otherwise,} \end{cases} \quad (33)$$

Where $z(t) = B_j U_{i,j}(t) g_i^j(t) / N_0(t) \ln 2$.

The proposed EDRA algorithm is executed in a distributed manner across the devices and the UAVs' edge servers. Specifically, Subproblems 1 and 2 are distributed and executed on each device, resulting in locally optimized computing and offloading decisions, with inputs that are only related to the device's own task queue. Subproblems 3 and 4 yield the computing and offloading decisions for the UAV's edge servers, and these subproblems are executed on the UAV. The specific algorithm design can be found in Algorithm 1.

Complexity analysis

The EDRA algorithm is designed to solve the task offloading and resource allocation in UAV-assisted MEC to optimize the energy consumption of system while maintaining the system performance. Previous work has also proposed several methods to address this issue, such as prediction-based methods and Game-based methods^{24,36}. Unlike these methods, EDRA transforms the stochastic optimization problem into a deterministic optimization problem, eliminating the need for specially find the way to predict external environmental information. Moreover, the decoupling of the problem allows the algorithm to avoid complex iterations.

Regarding the complexity analysis of the EDRA algorithm, the algorithm has two loops. In each UAV, it is required to sort according to the offloading workload of devices and process the tasks offloaded by devices, with complexities of $O(I \log_2 I)$ and $O(I)$, respectively. And the algorithm is currently running during T. Therefore, the overall complexity of the algorithm is $O(I \log_2 I JT)$.

Algorithm analysis for EDRA

In this section, we prove the performance of the algorithm through rigorous theoretical analysis. By verifying that the gap between the queue length and energy consumption of the algorithm and the optimal solution is within a constant level, we ensure the performance of the algorithm.

Lemma 1 *No matter what data set λ is provided, an optimal decision α^* can be obtained, which is unaffected by the length of the queue.*

According to this lemma, we can get as follows:

$$\begin{aligned} \mathbb{E}\{E_{total}^{\alpha^*}(t)\} &= E_{total}^*(\lambda), \\ \mathbb{E}\{A_{i,j}^{\alpha^*}(t)\} &\leq \mathbb{E}\{C_{i,j}^{l,\alpha^*}(t) + C_{i,j}^{o,\alpha^*}(t)\}, \\ \mathbb{E}\{C_{i,j}^{o,\alpha^*}(t)\} &\leq \mathbb{E}\{D_{i,j}^{l,\alpha^*}(t) + D_{i,j}^{o,\alpha^*}(t)\}. \end{aligned} \quad (34)$$

Proof The Lemma 1 can be proven by using Caratheodory's theorem.

Lemma 2 *No matter what data set λ is provided, we can get the following equation:*

$$E_{total}^{EDRA} \leq E_{total}^* + \frac{Z}{V}, \quad (35)$$

Proof The decision α and the random data set $\lambda + \varepsilon$ are defined. Thus, we can obtain the following inequality based on Lamma 1:

$$\begin{aligned} \mathbb{E}\{E_{total}^{\alpha}(t)\} &= E_{total}^*(\lambda + \varepsilon) \\ \mathbb{E}\{A_{i,j}^{\alpha}(t)\} + \varepsilon &\leq \mathbb{E}\{C_{i,j}^{l,\alpha}(t) + C_{i,j}^{o,\alpha}(t)\}, \\ \mathbb{E}\{D_{i,j}^{o,\alpha}(t)\} + \varepsilon &\leq \mathbb{E}\{D_{i,j}^{l,\alpha}(t) + D_{i,j}^{o,\alpha}(t)\}. \end{aligned} \quad (36)$$

Recall the inequality (23), and the drift-plus-penalty is as follows when we substitute α into decision set:

$$\begin{aligned} \Delta(\omega(t)) + V\mathbb{E}\{E_{total}(t)|\omega(t)\} &\leq Z + V\mathbb{E}\{o(t)|\omega(t)\} \\ + \mathbb{E}\{\sum_{j=1}^J \sum_{i=1}^{I_j} Q_{i,j}(t)[A_{i,j}^{\alpha}(t) - C_{i,j}^{l,\alpha}(t) - C_{i,j}^{o,\alpha}(t)]\} \\ + \mathbb{E}\{\sum_{j=1}^J \sum_{i=1}^{I_j} H_{i,j}(t)[C_{i,j}^{o,\alpha}(t) - D_{i,j}^{l,\alpha}(t) - D_{i,j}^{o,\alpha}(t)]\} \end{aligned} \quad (37)$$

Combine (36) with (37) and simplify them, we can get:

$$\mathbb{E}\{\Delta(\omega(t))\} + V\mathbb{E}\{E_{total}(t)|\omega(t)\} \leq Z + VE_{total}^*(\alpha + \varepsilon) - \varepsilon \sum_{j=1}^J \sum_{i=1}^{I_j} \mathbb{E}\{Q_{i,j}(t) + U_{i,j}(t)\} \quad (38)$$

Taking the maximum value of t , we get:

$$\begin{aligned} V \sum_{t=0}^{T-1} \mathbb{E}\{E_{total}(t)|\omega(t)\} &\leq ZT + VTE_{total}^*(\lambda + \varepsilon) \\ &\text{varepsilon} \sum_{t=0}^{T-1} \sum_{j=1}^J \sum_{i=1}^{I_j} \mathbb{E}\{Q_{i,j}(t) + U_{i,j}(t)\}. \end{aligned} \quad (39)$$

Since both ε and $Q_{i,j}(t) + U_{i,j}(t)$ are always greater than or equal to 0, we have the ability to scale them up:

$$V \sum_{t=0}^{T-1} \mathbb{E}\{o(t)|\omega(t)\} \leq ZT + VTE_{total}^*(\lambda + \varepsilon). \quad (40)$$

Lemma 2 can be proven when both sides divide VT and set the value of T to infinity.

Lemma 3 *No matter what data set λ is provided, we can get the following equation:*

$$\bar{L} \leq \frac{Z + V(E_{total}^{max} - E_{total}^*)}{\varepsilon} \quad (41)$$

Where $\bar{L} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{j=1}^J \sum_{i=1}^{I_j} \mathbb{E}\{(Q_{i,j}(t) + U_{i,j}(t))|\omega(t)\}$.

Proof Accodding to (36), we can also derive:

$$\varepsilon \sum_{t=0}^{T-1} \sum_{j=1}^J \sum_{i=1}^{I_j} \mathbb{E}\{(Q_{i,j}(t) + U_{i,j}(t))\} \leq ZT + VTE_{total}^*(\lambda + \varepsilon) - V \sum_{t=0}^{T-1} \mathbb{E}\{E_{total}(t)|\omega(t)\} \quad (42)$$

For any $E_{total}(t)$, there exists $E_{total}(t) \leq E_{total}^{max}(t) - E_{total}^{min}(t)$. Additionally, $V \sum_{t=0}^{T-1} \mathbb{E}\{E_{total}(t)|\omega(t)\} \geq 0$ can be get. Thus, we can derive:

$$\varepsilon \sum_{t=0}^{T-1} \sum_{j=1}^J \sum_{i=1}^{I_j} \mathbb{E}\{Q_{i,j}(t) + U_{i,j}(t)\} \leq ZT + VT(E_{total}^{max}(t) - E_{total}^{min}(t)). \quad (43)$$

Similar to Lemma 2, both sides divide εT and simplify them, then Lemma 3 is proven.

Evaluation

Experiment setup

In this section, we conduct a simulation experiment to analyze the parameters and compare the algorithms on a public EUA dataset³⁷, and similar settings²⁸ has been applied in experiments. The aim is to verify the adaptability of the algorithm to different operating environments and its superiority compared to other algorithms. We simulate a UAV-assisted MEC scenario with dimensions of [500,500,200], which includes 100 devices, 5 UAVs, and a cloud server. The devices are located at a height of 0, with randomly generated horizontal coordinates. Furthermore, the maximum CPU frequency of the devices is 1GHz, and the maximum offloading power is 0.1W. The five UAVs hover randomly within a height range of 100 to 200, and the edge servers on the UAVs have a maximum CPU frequency of 10GHz and a maximum offloading power of 1W. Each UAV has a total bandwidth of 1 MHz, which is allocated to connected devices through FDMA. It is assumed that the cloud server has superior computing capabilities. Devices choose the nearest UAV's edge server for offloading, with each UAV having a maximum capacity of 20 devices. The environmental parameters a and b for communication between devices and UAVs are 4.88 and 0.43, respectively, while the coefficients for LoS and NLoS transmission are 0.1 and 21.

Parameter analysis

As shown in Figs. 2 and 3, we analyze the trade-off parameter V in the drift-plus-penalty algorithm. We use a range of 10^{12} to 9×10^{12} as our values. V serves as a balancing factor that weighs the optimization of the queue and the optimization target (system energy consumption). Since V is the coefficient in front of the penalty term, a larger V results in a greater optimization effort for energy consumption, which in turn may lead to increased

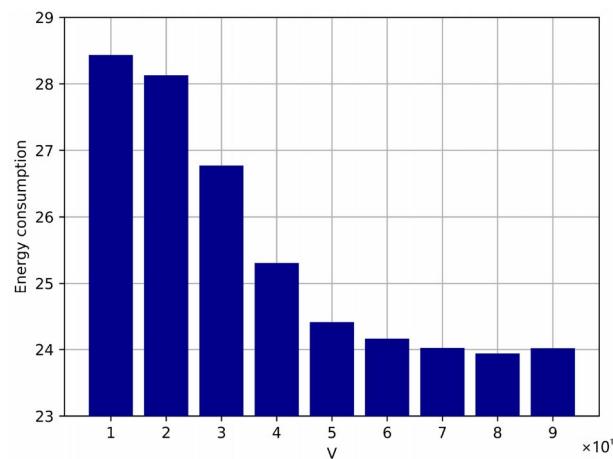


Fig. 2. The energy consumption of system vs. different trade-off parameter V.

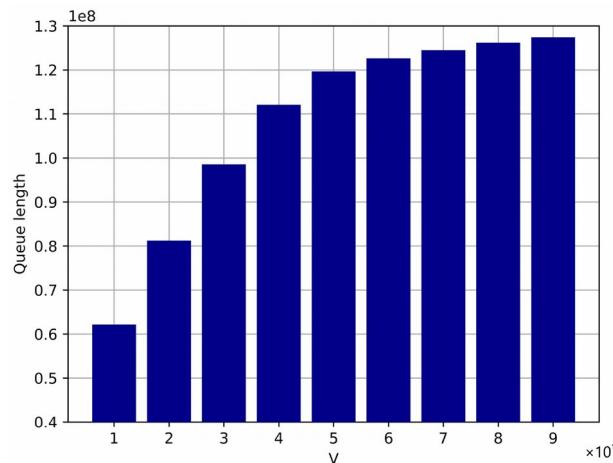


Fig. 3. The queue length of system vs. different trade-off parameter V.

queue backlog. As V increases, both energy consumption and queue backlog gradually converge, consistent with our theoretical analysis. The figures also reflect this understanding.

In Figs. 4 and 5, we multiply the benchmark $A_{i,j}(t) = [0, 10^6]$ bits/s by coefficients of 0.8, 1.0, and 1.2, representing 80%, 100%, and 120% of the standard task arrival rate. This demonstrates the impact of task arrival rates on device energy consumption and backlog, and reflects the ability of the EDRA algorithm to handle different degrees of task arrival rates. It can be seen that under different task arrival rates, EDRA achieves convergence relatively quickly, and a higher task arrival rate leads to greater energy consumption and queue backlog. This is in line with objective conditions.

In Figs. 6 and 7, we test the energy consumption and queue backlog as the number of devices increases from 40 to 120, while keeping the number of UAVs, task arrival rate, and the maximum CPU frequency of each device constant. The results reflect the adaptability of the EDRA algorithm to different numbers of devices, and both energy consumption and queue backlog increase as the number of devices increases.

Comparative experiment

In the comparative experiment section, we prepared four algorithms:

EDRA

This is the algorithm proposed in this paper.

Local-only(LO)

This algorithm allows devices to process local computation without task offloading. It serves as one of the baseline algorithms for comparison.

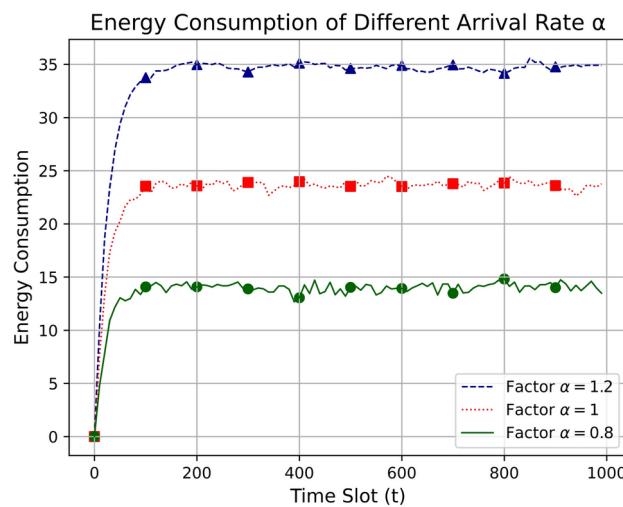


Fig. 4. The energy consumption of system with different α (scale factor of arrival rates).

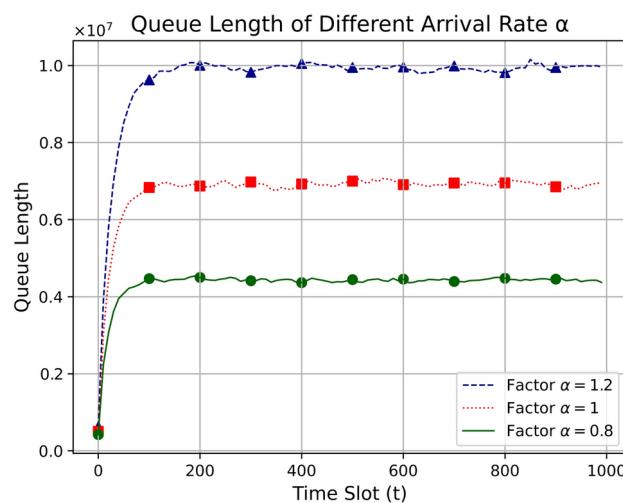


Fig. 5. The queue length of system with different α (scale factor of arrival rates).

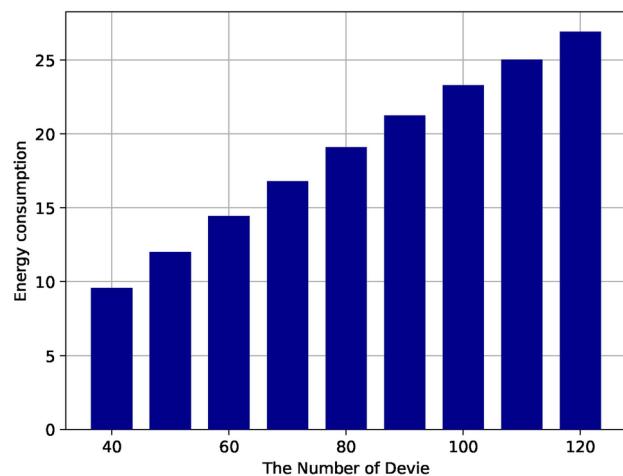


Fig. 6. The energy consumption of system with the different number of devices.

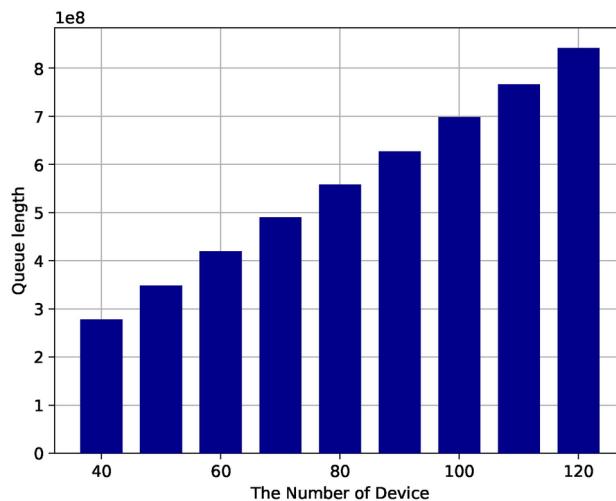


Fig. 7. The queue length of system with the different number of devices.

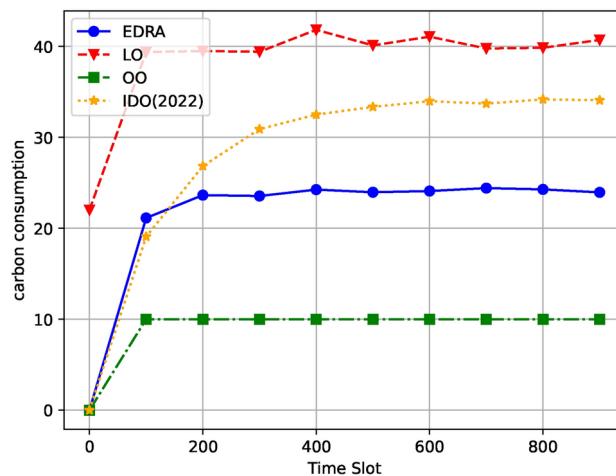


Fig. 8. The energy consumption of system with different algorithms.

Offloading-only(OO)

This algorithm allows devices to offload all tasks to the edge servers for computation. It serves as another baseline algorithm for comparison.

IDO (2022)

This algorithm is adapted from the design concept in the paper²⁰, with some modifications to make it operable in our environment.

Figure 8 shows the performance of the four algorithms in the simulated UAV-assisted MEC system. Among the four algorithms, the OO algorithm has the lowest energy consumption and remains unchanged because when devices offload all tasks to the drones, the processing tasks exceed the maximum offloading power, causing the offloading power to operate at maximum capacity. The second-lowest energy consumption is achieved by the EDRA algorithm proposed in this paper. It can be seen that EDRA converges faster and has lower energy consumption compared to the relatively advanced IDO algorithm, as our algorithm decomposes the problem for parallel processing. The LO algorithm has the highest energy consumption, nearly double that of our proposed algorithm.

Figure 9 shows the comparison the queue backlog of the four algorithms. It can be seen that although the LO algorithm has the lowest energy consumption, unfortunately, it achieves this at the cost of generating a significant amount of computational energy consumption. Our proposed algorithm has the second-lowest queue backlog, significantly lower than the IDO algorithm. As a baseline, the OO algorithm clearly cannot stabilize the queue length, which would lead to significant delays in the system's task processing.

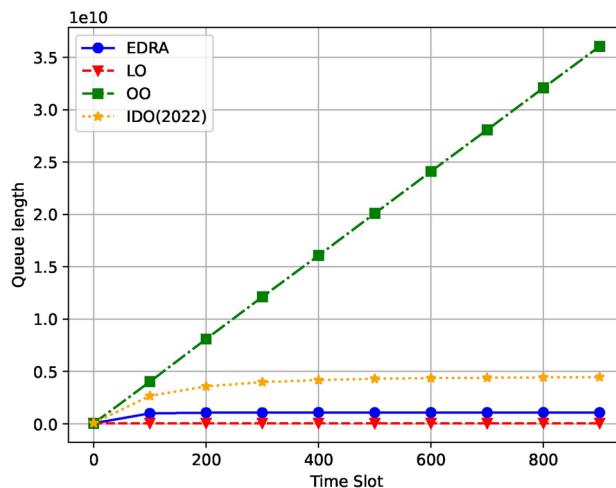


Fig. 9. The queue length of system with different algorithms.

Conclusion

In this paper, we investigate the task offloading and resource allocation problem in a UAV-assisted MEC system for large-scale sports events, where UAVs are deployed near the venue to provide temporary edge computing services. We formulate the system as a stochastic optimization problem and transform the original problem into an upper bound problem, then decompose it into multiple subproblems using stochastic optimization techniques. Specifically, we obtain optimal or suboptimal solutions for each subproblem through convex optimization, linear programming, and other methods. Furthermore, we prove the performance of the algorithm through theoretical analysis. We summarize a series of parameter analyses and comparative experiments to evaluate the adaptability of the problem in different scenarios and the degree of optimization for energy consumption. The experimental results show that the EDRA algorithm can effectively reduce system energy consumption by 32.4% compared to advanced algorithms, and more reliably ensure system stability. In future work, we will address the task offloading problem in UAV-assisted MEC systems by jointly optimizing UAV deployment scheduling and task resource allocation.

Data availability

All data generated or analysed during this study are included in this published article.

Received: 4 January 2025; Accepted: 1 April 2025

Published online: 07 April 2025

References

1. Tian, S. et al. User preference-based hierarchical offloading for collaborative cloud-edge computing. *IEEE Trans. Serv. Comput.* **16**, 684–697. <https://doi.org/10.1109/TSC.2021.3128603> (2023).
2. Xu, Y. et al. Uav-assisted mec networks with aerial and ground cooperation. *IEEE Trans. Wireless Commun.* **20**, 7712–7727. <https://doi.org/10.1109/TWC.2021.3086521> (2021).
3. Cao, Y. et al. Review on the application of cloud computing in the sports industry. *J. Cloud Comput.* **12**, 152 (2023).
4. Du, Y. et al. Edge computing-based digital management system of game events in the era of internet of things. *J. Cloud Comput.* **12**, 44 (2023).
5. Chen, Y., Zhao, J., Wu, Y., Huang, J. & Shen, X. Qoe-aware decentralized task offloading and resource allocation for end-edge-cloud systems: A game-theoretical approach. *IEEE Trans. Mob. Comput.* **23**, 769–784. <https://doi.org/10.1109/TMC.2022.3223119> (2024).
6. Wang, M. et al. Stackelberg-game-based intelligent offloading incentive mechanism for a multi-uav-assisted mobile-edge computing system. *IEEE Internet Things J.* **10**, 15679–15689. <https://doi.org/10.1109/JIOT.2023.3265432> (2023).
7. Liwang, M. et al. Graph-represented computation-intensive task scheduling over air-ground integrated vehicular networks. *IEEE Trans. Serv. Comput.* **16**, 3397–3411 (2023).
8. Singh, M. B., Singh, H. & Pratap, A. Stable matching based revenue maximization for federated learning in uav-assisted wbans. *IEEE Trans. Serv. Comput.* <https://doi.org/10.1109/TSC.2024.3360692> (2024).
9. Ding, Y. et al. Ddqn-based trajectory and resource optimization for uav-aided mec secure communications. *IEEE Trans. Veh. Technol.* <https://doi.org/10.1109/TVT.2023.3335210> (2023).
10. Zeng, Y. & Tang, J. Mec-assisted real-time data acquisition and processing for uav with general missions. *IEEE Trans. Veh. Technol.* **72**, 1058–1072 (2023).
11. Fu, S., Feng, X., Sultana, A. & Zhao, L. Joint power allocation and 3d deployment for uav-bss: A game theory based deep reinforcement learning approach. *IEEE Trans. Wireless Commun.* **23**, 736–748 (2024).
12. Zheng, C. et al. Multi-agent collaborative optimization of uav trajectory and latency-aware dag task offloading in uav-assisted mec. *IEEE Access* **12**, 42521–42534. <https://doi.org/10.1109/ACCESS.2024.3378512> (2024).
13. Qian, L. P. et al. Optimal sic ordering and computation resource allocation in mec-aware noma nb-iot networks. *IEEE Internet Things J.* **6**, 2806–2816. <https://doi.org/10.1109/JIOT.2018.2875046> (2019).
14. Ding, Z., Xu, D., Schober, R. & Poor, H. V. Hybrid noma offloading in multi-user mec networks. *IEEE Trans. Wireless Commun.* **21**, 5377–5391. <https://doi.org/10.1109/TWC.2021.3139932> (2022).

15. Ernest, T. Z. H. & Madhukumar, A. S. Computation offloading in mec-enabled iot networks: Average energy efficiency analysis and learning-based maximization. *IEEE Trans. Mob. Comput.* **23**, 6074–6087. <https://doi.org/10.1109/TMC.2023.3315275> (2024).
16. Chu, W., Jia, X., Yu, Z., Lui, J. C. & Lin, Y. Joint service caching, resource allocation and task offloading for mec-based networks: A multi-layer optimization approach. *IEEE Trans. Mob. Comput.* **23**, 2958–2975. <https://doi.org/10.1109/TMC.2023.3268048> (2024).
17. Tang, L. & Hu, H. Computation offloading and resource allocation for the internet of things in energy-constrained mec-enabled hetnets. *IEEE Access* **8**, 47509–47521. <https://doi.org/10.1109/ACCESS.2020.2979774> (2020).
18. Liu, L. et al. Multi-user dynamic computation offloading and resource allocation in 5g mec heterogeneous networks with static and dynamic subchannels. *IEEE Trans. Veh. Technol.* **72**, 14924–14938. <https://doi.org/10.1109/TVT.2023.3285069> (2023).
19. Heidarpour, A. R., Heidarpour, M. R., Ardakani, M., Tellambura, C. & Uysal, M. Soft actor-critic-based computation offloading in multiuser mec-enabled iot-a lifetime maximization perspective. *IEEE Internet Things J.* **10**, 17571–17584. <https://doi.org/10.1109/IJOT.2023.3277753> (2023).
20. Li, R., Lim, C. S., Rana, M. E. & Zhou, X. A trade-off task-offloading scheme in multi-user multi-task mobile edge computing. *IEEE Access* **10**, 129884–129898. <https://doi.org/10.1109/ACCESS.2022.3228403> (2022).
21. Sun, Y., Wei, T., Li, H., Zhang, Y. & Wu, W. Energy-efficient multimedia task assignment and computing offloading for mobile edge computing networks. *IEEE Access* **8**, 36702–36713. <https://doi.org/10.1109/ACCESS.2020.2973359> (2020).
22. Sun, G., Li, J., Liu, Y., Liang, S. & Kang, H. Time and energy minimization communications based on collaborative beamforming for uav networks: A multi-objective optimization method. *IEEE J. Sel. Areas Commun.* **39**, 3555–3572. <https://doi.org/10.1109/JSA.C.2021.3088720> (2021).
23. Picanco, B. & Fantacci, R. A combined stochastic network calculus and matching theory approach for computational offloading in a heterogenous mec environment. *IEEE Trans. Netw. Serv. Manage.* **21**, 1958–1968. <https://doi.org/10.1109/TNSM.2023.3343290> (2024).
24. Chen, Y., Yang, Y., Hu, J., Wu, Y. & Huang, J. A game-theoretical approach for distributed computation offloading in leo satellite-terrestrial edge computing systems. *IEEE Trans. Mobile Comput.* <https://doi.org/10.1109/TMC.2025.3526200> (2025).
25. Gao, X. & Zhai, L. Service experience oriented cooperative computing in cache-enabled uavs assisted mec networks. *IEEE Trans. Mobile Comput.* <https://doi.org/10.1109/TMC.2024.3366944> (2024).
26. Chen, Y. et al. Hnio: A hybrid nature-inspired optimization algorithm for energy minimization in uav-assisted mobile edge computing. *IEEE Trans. Netw. Serv. Manage.* **19**, 3264–3275. <https://doi.org/10.1109/TNSM.2022.3176829> (2022).
27. Wang, M. et al. Stackelberg-game-based intelligent offloading incentive mechanism for a multi-uav-assisted mobile-edge computing system. *IEEE Internet Things J.* **10**, 15679–15689. <https://doi.org/10.1109/IJOT.2023.3265432> (2023).
28. Ying, C., Yaozong, Y., Yuan, W., Jiwei, H. & Lian, Z. Joint trajectory optimization and resource allocation in uav-mec systems: A lyapunov-assisted drl approach. *IEEE Trans. Services Comput.* <https://doi.org/10.1109/TSC.2025.3544124> (2024).
29. Chen, Y., Li, K., Wu, Y., Huang, J. & Zhao, L. Energy efficient task offloading and resource allocation in air-ground integrated mec systems: A distributed online approach. *IEEE Trans. Mob. Comput.* **23**, 8129–8142. <https://doi.org/10.1109/TMC.2023.3346431> (2024).
30. Liao, H., Zhou, Z., Zhao, X. & Wang, Y. Learning-based queue-aware task offloading and resource allocation for space-air-ground-integrated power iot. *IEEE Internet Things J.* **8**, 5250–5263. <https://doi.org/10.1109/IJOT.2021.3058236> (2021).
31. Sun, Y., Xu, J. & Cui, S. User association and resource allocation for mec-enabled iot networks. *IEEE Trans. Wireless Commun.* **21**, 8051–8062. <https://doi.org/10.1109/TWC.2022.3163809> (2022).
32. Chen, Y., Li, K., Wu, Y., Huang, J. & Zhao, L. Energy efficient task offloading and resource allocation in air-ground integrated mec systems: A distributed online approach. *IEEE Trans. Mob. Comput.* **23**, 8129–8142. <https://doi.org/10.1109/TMC.2023.3346431> (2024).
33. Jing, J. et al. Multi-uav cooperative task offloading in blockchain-enabled mec for consumer electronics. *IEEE Trans. Consumer Electron.* <https://doi.org/10.1109/TCE.2024.3485633> (2024).
34. Hu, H., Wang, Q., Hu, R. Q. & Zhu, H. Mobility-aware offloading and resource allocation in a mec-enabled iot network with energy harvesting. *IEEE Internet Things J.* **8**, 17541–17556. <https://doi.org/10.1109/IJOT.2021.3081983> (2021).
35. Yang, Y., Chen, Y., Li, K. & Huang, J. Carbon-aware dynamic task offloading in noma-enabled mobile edge computing for iot. *IEEE Internet Things J.* **11**, 15723–15734. <https://doi.org/10.1109/IJOT.2024.3351200> (2024).
36. Xiao, Z. et al. Vehicular task offloading via heat-aware mec cooperation using game-theoretic method. *IEEE Internet Things J.* **7**, 2038–2052. <https://doi.org/10.1109/IJOT.2019.2960631> (2020).
37. He, Q. et al. A game-theoretical approach for user allocation in edge computing environment. *IEEE Trans. Parallel Distrib. Syst.* **31**, 515–529. <https://doi.org/10.1109/TPDS.2019.2938944> (2020).

Acknowledgments

This work was supported by the The National Social Science Fund of China, Project Title: Research on the Construction of Sports All-MediaCommunication Pattern in the New Era (grant number 21&ZD345).

Additional information

Correspondence and requests for materials should be addressed to C.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025