Multi-UAV Cooperative Task Offloading in Blockchain-Enabled MEC for Consumer Electronics

Jinfeng Jing, Yaozong Yang, Xiaokang Zhou, *Senior Member, IEEE*, Jiwei Huang, *Senior Member, IEEE*, Lianyong Qi, *Senior Member, IEEE*, and Ying Chen, *Senior Member, IEEE*

***Abstract*—The development of the Internet of Things (IoT) has made the traditional Consumer Electronics (CE) evolve into a next-generation CE with many intelligent application which generates a large number of computing-intensive tasks and is sensitive to latency. The combination of blockchain technology and multi-UAV-assisted edge computing is considered a promising technology that can process tasks near CE devices instead of sending data to remote data centers, and it has advantages in flexible deployment, differentiated services, and security assurance. This paper considers a Blockchain-enabled MEC scenario where CE devices can offload tasks to multiple-UAV-edge layers for decentralized cooperative computing, and utilize the reputation mechanism of blockchain to identify abnormal nodes in edge offloading. To solve the problem of computation offloading and resource allocation in this scenario, we jointly optimize the CPU cycle frequencies and offloading resources of UAVs and CE devices, and the decision of UAVs. Our goal is to minimize system energy consumption while ensuring the quality of service. The stochastic optimization technique is employed to transform the original stochastic optimization problem into a deterministic optimization problem, decomposing it into multiple sub-problems which can be solved in parallel. Additionally, we design a dynamic priority scoring strategy to facilitate cooperative processing among multiple UAVs and propose a Blockchain-Based Multi-UAV Cooperative Task Offloading (BMCTO) algorithm. Through theoretical analysis and simulation experiments, we demonstrate that the BMCTO algorithm can effectively reduce system energy consumption while maintaining system performance.**

***Index Terms*—Multi-UAV cooperation, resource allocation, blockchain, multi-access edge computing (MEC), stochastic optimization.**

Received 2 July 2024; revised 1 September 2024, 13 October 2024, and 19 October 2024; accepted 19 October 2024. Date of publication 24 October 2024; date of current version 12 June 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62472039; in part by the Beijing Natural Science Foundation under Grant L232050; in part by the Project of Cultivation for Young Top-Motch Talents of Beijing Municipal Institutions under Grant BPHR202203225; and in part by the Young Elite Scientists Sponsorship Program by BAST under Grant BYESS2023031. *(Corresponding authors: Xiaokang Zhou; Ying Chen.)*

Jinfeng Jing, Yaozong Yang, and Ying Chen are with the School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China (e-mail: jingjinfeng@bistu.edu.cn; yyz@bistu.edu.cn; chenying@bistu.edu.cn).

Xiaokang Zhou is with the Faculty of Business and Data Science, Kansai University, Suita 565-0823, Japan (e-mail: zhou@kansai-u.ac.jp).

Jiwei Huang is with the Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum, Beijing 102249, China (e-mail: huangjw@cup.edu.cn).

Lianyong Qi is with the College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266024, China (e-mail: lianyongqi@gmail.com).

Digital Object Identifier 10.1109/TCE.2024.3485633

I. Introduction

With the rapid development of Internet of Things (IoT) and communication technology [1], [2], types of applications on Consumer Electronics (CE) devices are experiencing explosive growth, such as facial recognition and augmented reality [3], [4]. However, these applications generate a large number of computation-intensive tasks which are sensitive to delay and consume a considerable amount of energy [5]. CE devices are limited by computational resources and battery capacity, making it difficult to handle these tasks locally. To solve these challenges, multi-access edge computing (MEC) is proposed [6], [7]. Blockchain-enabled MEC supports CE devices to offload some complex computation-intensive tasks to servers at the network edge [8], effectively overcoming the limitations of CE device energy and computational capacity, and meeting the demand for strong computing power, low latency, low energy consumption, and more security [9], [10], [11].

Traditional edge servers are usually installed in fixed ground locations in MEC, which makes it difficult to apply them to remote areas that lack sufficient ground infrastructure, areas affected by natural disasters, and emergency rescue operations [12]. The emergence of unmanned aerial vehicle (UAV) assisted MEC systems effectively addresses these challenges [13]. UAVs have the advantages of flexible deployment and wide coverage. The edge server can be deployed on the UAV as a flexible relay station or mobile base station to provide edge computing services for CE devices [14]. However, the system's performance is affected by the capabilities and reliability of the UAV. There are many methods in the existing research to ensure the reliability of UAVs [15], [16], [17]. Adopting a multi-UAV collaborative processing architecture can reduce the impact of UAV reliability. Combining the reliability research of UAVs with multi-UAV MEC systems can effectively avoid the impact of UAV malfunction or performance degradation on the system, thereby improving the overall performance of the system.

A. Motivation

Multi-UAV collaboration MEC system requires effective scheduler and management mechanisms to ensure task offloading of CE devices and cooperative work among UAVs, involving complex issues such as task offloading, resource allocation, and security mechanisms [18]. While some efforts

have been made to address these issues, but many challenges remain.

In the past works, most studies [19], [20], [21] utilize UAVs to serve ground users, focusing on resource allocation and task offloading in more complex air-to-ground communication scenarios. However, data leakage may occur during the task offloading from CE devices to UAVs and the collaborative work between UAVs. Only a minority of researches consider the security and privacy of multi-UAV-assisted MEC systems. In these studies [22], [23], the utilization of blockchain's reputation mechanism to identify abnormal nodes (malicious attacks, insufficient performance) in MEC systems has been considered [24]. However, improper use of reputation mechanisms may cause excessive task concentration on certain nodes, leading to load imbalance among nodes. In addition, integrating blockchain technology [25], [26] enhances system security while increasing the complexity of problem-solving, making already complex problems more challenging.

The arrival of tasks on CE devices is random and sudden, and the communication status of wireless channels is unstable, making it difficult to predict task arrivals and allocate resources ahead of time accurately. The task receiving and processing capabilities of UAVs are also affected by fluctuations in channel conditions, further exacerbating the challenge of resource scheduling. While the computing power of edge servers equipped on UAVs far exceeds CE devices, it is still limited. Excessive loads can affect task processing efficiency, and environmental factors can affect channel conditions between users and UAVs, as well as UAVs and UAVs. In this complex environment, coordination of tasks and resource allocation is critical. Choosing the appropriate offloading target and coordinating cooperation among UAVs to achieve load balancing are challenges in multi-UAV collaboration. In addition, effective collaboration between UAVs is also a challenge. Each UAV needs to share its load, available resources, and communication link quality information to make optimal task allocation decisions.

To sum up, Multi-UAV collaboration in blockchain-enabled MEC system faces challenges on multiple fronts. Therefore, considering the security of the system, coordination among multiple UAVs, uncertainty of tasks, and dynamic communication environments, designing an efficient algorithm to solve these problems is critical to advancing the development and deployment of powerful, efficient, and safe multi-UAV MEC systems.

B. Contribution

This paper investigates the problem of Multi-UAV Cooperative Task Offloading in the blockchain-enabled MEC systems. The goal is to achieve optimal energy consumption while ensuring system performance and meeting long-term constraints. We jointly optimize the CPU cycle frequencies and offloading resources of UAVs and CE devices, and the decision of UAVs. The stochastic optimization technique is employed to transform the original stochastic optimization problem into a deterministic optimization problem, decomposing it into multiple sub-problems which can be solved in parallel. We apply convex optimization and linear programming to solve different subproblems, design a dynamic priority scoring strategy to facilitate cooperative processing among multiple UAVs and propose the Blockchain-Based Multi-UAV Cooperative Task Offloading (BMCTO) algorithm. Through a series of theoretical proofs and simulation experiments, we validate the excellent performance of the BMCTO algorithm. The main contributions of this paper are as follows:

1) We investigate the problem of task offloading and resource allocation in multi-UAV cooperative blockchain-enabled MEC systems, aiming to minimize the energy cost of the system. Considering that the task arrival of CE devices is difficult to predict, and the communication quality of wireless channels is dynamically random, we use the CPU cycle frequencies and offloading resources of UAVs and CE devices, and the decision of UAVs as our decision variables to re-model the problem.
2) To solve the problem mentioned above, we use stochastic optimization technology to transform the original problem into a deterministic optimization problem and decompose it into five sub-problems. We propose the BMCTO algorithm where convex optimization and linear programming are used to solve these sub-problems of resource allocation. What's more, considering load balancing and offloading security, we design a dynamic priority scoring strategy based on blockchain to solve the problem of UAVs' decisions, which supports UAV's decentralized cooperation and allows tasks to be offloaded from each other.
3) Through theoretical analysis, we prove that the gap between the BMCTO algorithm and the optimal solution is constant. We conduct a large number of simulation experiments. Through parameter analysis, we demonstrate that the algorithm can adapt to different environmental information. Through comparative experiments, we show that the proposed algorithm can effectively reduce system energy consumption while ensuring system stability.

The rest of this paper is organized as follows. Section II introduces related work. Section III establishes the system model and formulates the problem. Section IV discusses the transformation of the original problem and proposes relevant algorithms. Section V analyzes the performance of the algorithms. Section VI discusses the results of the simulation experiments. Finally, Section VII concludes the paper.

II. Related Work

In recent years, there has been a lot of research on UAV-assisted blockchain-enabled MEC systems, and certain achievements have been made. Guo and Liu [27] investigated the problem of energy efficiency enhancement in UAV-assisted MEC system through methods such as intelligent offloading decisions, uplink and downlink transmission resource allocation, and UAV trajectory design. Yang et al. [28] studied a UAV-enabled edge computing system that provides services for multiple energy harvesting (EH) devices, with

optimization objectives of UAV task processing rate and energy. Hu et al. [29] addressed a MEC architecture where UAVs can act as computing servers or relays, and proposed a scheme based on alternating optimization. These studies primarily focused on scenarios involving a single UAV. However, due to limitations in computational capabilities and battery capacity of individual UAVs, they cannot adequately meet the computational demands of CE devices over the long term. Consequently, research into multi-UAV assisted MEC systems has emerged.

The process of task offloading in MEC systems faces security and privacy issues, while the confidentiality, integrity, access control, and privacy functions of blockchain technology can be used to ensure the security and reliability of the task offloading process. Combining blockchain technology and multi-UAV-assisted edge computing has many actual use cases, for example, in areas such as disaster monitoring and agricultural management. The integration of MEC and blockchain technology has brought tremendous potential to the relevant research of MEC systems. Samy et al. [30] studied the security issues of task offloading in MEC systems and proposed a task offloading model based on blockchain, using deep learning algorithms to provide optimal task offloading decisions while ensuring system performance. Hosseinpour and Yaghmaee [31] aimed at enhancing the user's experience quality parameters and proposed a novel MEC system framework, utilizing blockchain technology to solve the issues of task offloading in MEC systems and the Proof of Work (PoW) puzzle in public blockchains. Wang et al. [32] addressed the security and privacy issues in computation offloading by integrating blockchain technology into drone-supported MEC networks, proposing an innovative Delegated Proof of Stake (DPoS) consensus mechanism. Xiao et al. [33] studied the resource allocation and task offloading issues on the Mobile Edge Platoon Cloud (MEPC) in the vehicular networking scenario, transforming the problem into a Stackelberg game and introducing a consortium blockchain to ensure the security and privacy of service transactions.

In addition, the issue of collaboration among multiple UAVs is also a current research hotspot. He et al. [34] proposed a multi-hop task offloading dynamic computation scheme, where computational tasks are offloaded through multiple UAVs multiple times to achieve a more powerful multi-UAV remote edge computing network. Qi et al. [35] leveraged the cooperative advantage of multiple UAVs in a formation and proposed a dual-computation offloading mechanism based on the Connected Dominating Set (CDS) for multi-UAV-assisted MEC networks, addressing the optimization problem of maximizing energy efficiency in the system. In the context of multi-UAV assisted aerial offloading systems described in [36], tasks can be cooperatively forwarded by UAVs, and a heuristic algorithm was proposed to jointly optimize offloading delay, MIMO channels, transmit power, and UAV deployment. Zeng et al. [37] considered UAVs as computing nodes under energy harvesting conditions and designed a chain offloading algorithm to minimize execution delay. However, in the above work, cooperative UAVs were given by default without considering how to select UAVs as cooperative objects. In our research, we provided UAVs' decisions. In addition, the above work only focuses on the resource allocation of CE devices or UAVs unilaterally, but our work takes them into account comprehensively.

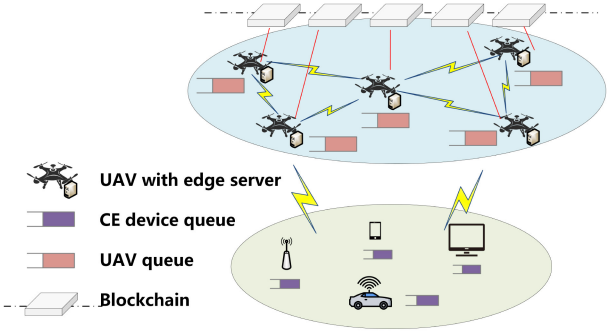

Fig. 1. Multi-UAV cooperative task offloading in blockchain-enabled MEC system model for CE.

III. System Model and Problem Formulation

In this section, we present the system model, problem formulation, and provide detailed descriptions of the communication model, blockchain system model, task and queue model, as well as the energy consumption model within the framework of the Blockchain-Enabled system.

A. Blockchain-Enabled System Model

Our design system is deployed at multiple endpoints, including UAVs and CE devices from different manufacturers. As shown in Fig. 1, the system model consists of two parts: the ground layer which encompasses various CE devices, and the aerial layer which includes multi-UAV clusters. We assume that these devices can exchange their respective key information during the communication process between UAVs and CE devices [38], [39]. In the interaction between the ground and aerial layers, ground to air communication occurs between CE devices and UAVs. Owing to the constraints in computational power and battery capacity, when CE devices generate a large number of computation-intensive tasks, CE devices can offload a part of computational tasks to UAVs within their respective areas to reduce the computational burden on CE devices. Due to the randomness and dynamicity of the system, UAVs may experience high load and idle states. Therefore, high load UAVs can unload some tasks to low load UAVs for computation, thereby reducing the computational pressure on high load UAVs, achieving multi-UAV cooperation, and improving system efficiency.

Although establishing such a system requires specialized hardware and infrastructure, resulting in increased initial costs. However, considering the scenarios that traditional base stations cannot meet, UAV systems' flexibility, and their hardware's high reusability, deploying a blockchain-enabled multi-UAV MEC system is reasonable and has long-term value. In addition, the long-term benefits of using blockchain-enabled multi-UAV MEC systems can offset the operational

TABLE I
KEY SYMBOLS

Symbol	Definition
$\mathcal{N}$	the set of CE devices
$\mathcal{M}$	the set of UAVs
τ	the length of time slot
$\partial_{i,j}$	the decision of CE device i
$\alpha_{j,j'}[t]$	the decision of UAV j
$g_i^j[t]$	the channel gain when CE device i communicates with UAV j
$G_j^{j'}[t]$	the channel gain when UAV j communicates with UAV j'
$\eta_i^j[t]$	the signal-to-noise ratio when CE device i communicates with UAV j
$\chi_j^{j'}[t]$	the signal-to-noise ratio when UAV j communicates with UAV j'
$f_i[t]$	the CPU frequency of CE device i
$F_j[t]$	the CPU frequency of UAV j
c_i	the number of CPU cycles required for CE device i to compute one bit of data
C_j	the number of CPU cycles required for UAV j to compute one bit of data
$l_i^c[t]$	the size of task processed locally by the CE device i
$L_j^c[t]$	the size of task processed locally by the UAV j
$l_i^o[t]$	the size of task offloaded by CE device i
$L_j^o[t]$	the size of task offloaded by UAV j
$h_{j'}^o[t]$	the size of task offloaded from collaborative UAV j'
$a_i[t]$	the arriving data bits of the CE device i
$A_j[t]$	the total task load received by UAV j
$\Gamma[t]$	reputation value
$Q_i[t]$	the queue backlog of CE device i
$X_j[t]$	the queue backlog of UAV j

costs and complexity caused by regular infrastructure maintenance and upgrades. By ensuring secure and reliable task offloading, the system can significantly improve overall system performance and reduce energy consumption, which are critical for the viability and effectiveness of the overall system's long-term execution.

In our system model, CE devices refer to devices with limited battery capacity and computation. The set of CE devices is denoted as $\mathcal{N} = \{0, 1, 2, \ldots, i, \ldots, N-1\}$. Each UAV is equipped with an edge server for processing tasks offloaded from CE devices or other UAVs. The set of UAVs is discretized as $\mathcal{M} = \{0, 1, 2, \ldots, j, \ldots, M-1\}$. We consider equal-length continuous time slots, each with a duration of τ, denoted as $\mathcal{T} = \{0, 1, 2, , \ldots, t, \ldots, T-1\}$. For the convenience of readers' reference, the Key symbols are listed in Table I.

B. Communication Model

In our model, each UAV communicates only with CE devices within a specific region, and the communication regions of each UAV do not overlap. However, communication is possible between any pair of UAVs. We use $\partial_{i,j} \in \{0, 1\}$ to describe the selection relationship between CE devices and UAVs. If $\partial_{i,j} = 1$, then CE device i selects UAV j as the target for offloading tasks, otherwise $\partial_{i,j} = 0$. Similarly, we define $\alpha_{j,j'}[t] \in \{0, 1\}$ as the decision of UAV j, where $\alpha_{j,j'}[t] = 1$ indicates that UAV j selects UAV j' within its communication range with the minimum task processing cost as the target for task offloading, otherwise $\alpha_{j,j'}[t] = 0$. It is important to note that the decision for CE device i is fixed at the initial time and remains unchanged in each time slot, whereas the decisions of UAVs will change in every time slot. Therefore, the following constraints apply:

$$\sum_{j'=0, j' \neq j}^{M-1} \alpha_{j,j'}[t] = 1, \ \forall j, j' \in \mathcal{M}. \tag{1}$$

In addition, Non-Orthogonal Multiple Access (NOMA) technology is well-suited for real-time applications in IoT scenarios and the interconnection of numerous devices [40], [41]. NOMA divides channels into multiple sub-channels capable of accommodating multiple device connections within each sub-channel. This approach reduces transmission latency and allows for processing more simultaneous connections, thereby significantly improving system efficiency and response speed. Given these characteristics, NOMA technology is suitable for our scenario requirements and is used for communication between UAVs and CE devices, as well as among UAVs. At the same time, we consider environmental factors. From a system perspective, environmental factors ultimately affect arrival rates and communication conditions. To a certain extent, we included the impact of environmental factors on the system in the task arrival rate and channel conditions. We embodied the impact of environmental factors on communication through the loss function.

1) CE Device-UAV Communication Model: As indicated by [42], path loss calculation when CE device i communicates with UAV j is expressed as:

$$L_i^{CE} = 20 \log_{10} \left(\frac{4\pi f_c \sqrt{(d_i^j)^2 + (h_i^j)^2}}{\iota} \right) + \Phi_i^{LoS} \xi_i^{LoS} + \left(1 - \Phi_i^{LoS}\right) \xi_i^{NLoS}, \tag{2}$$

where ι denotes the speed of light, f_c represents the carrier frequency, d_i^j and h_i^j respectively represent the horizontal distance and height distance between CE device i and UAV j, ξ_i^{LoS} and ξ_i^{NLoS} represent the line-of-sight path loss and non-line-of-sight path loss when CE device and UAV communicate, and Φ_i^{LoS} is represented as

$$\Phi_i^{LoS} = \frac{1}{1 + \phi exp\left\{-\varrho\left[\tan^{-1}\left(\frac{h_i^j}{d_i^j}\right) - \phi\right]\right\}}, \tag{3}$$

where the values of ϕ, ϱ, ξ_i^{LoS}, and ξ_i^{NLoS} are dependent on the environmental conditions [43].

Therefore, the channel gain when CE device i communicates with UAV j can be defined as:

$$g_i^j[t] = 10^{-\frac{L_i^{CE}}{10}}. \tag{4}$$

the signal-to-noise ratio can be represented as:

$$\eta_i^j[t] = \frac{\partial_{i,j} P_i[t] g_i^j[t]}{\sum_{i'=0, i' \neq i}^{N-1} \partial_{i',j} P_{i'}[t] g_{i'}^j[t] + \sigma^2}, \tag{5}$$

where $P_i[t]$ is the transmission power of CE device, with $0 \leq P_i[t] \leq P_i^{max}$. $\sum_{i'=0, i' \neq i}^{N-1} \partial_{i',j} P_{i'}[t] g_{i'}^j[t]$ denotes the

communication interference from other CE device when CE device i communicates with UAV j, and σ^2 represents the noise power. In conclusion, the data transmission rate of CE device i can be expressed as:

$$r_i^j[t] = B\log_2\left(1+\eta_i^j[t]\right), \tag{6}$$

where B is the transmission bandwidth of CE device.

2) UAV-UAV Communication Model: Similarly [42], the path loss calculation when UAVs communicate with each other is:

$$L_j^{UAV} = 20\log_{10}\left(\frac{4\pi f_c\sqrt{\left(D_j^{j'}\right)^2+\left(H_j^{j'}\right)^2}}{\iota}\right) + \Psi_j^{LoS}\zeta_j^{LoS} + \left(1-\Psi_j^{LoS}\right)\zeta_j^{NLoS}, \tag{7}$$

where $D_j^{j'}$ and $H_j^{j'}$ respectively represent the horizontal distance and height distance between UAV j and UAV j', ζ_j^{LoS} and ζ_j^{NLoS} denote the line-of-sight path loss and non-line-of-sight path loss when UAVs communicate with each other, and Ψ_j^{LoS} is represented as

$$\Psi_j^{LoS} = \frac{1}{1+\varkappa exp\left\{-\varphi\left[tan^{-1}\left(\frac{H_j^{j'}}{D_j^{j'}}\right)-\varkappa\right]\right\}}, \tag{8}$$

where the values of $\varkappa$, φ, ζ_j^{LoS}, and ζ_j^{NLoS} are determined by the environment [43].

Furthermore, we can obtain the channel gain $G_j^{j'}[t]$ and signal-to-noise ratio $\chi_j^{j'}[t]$ when UAV j communicates with UAV j' as [42]:

$$G_j^{j'}[t] = 10^{-\frac{L_j^{UAV}}{10}}, \tag{9}$$

$$\chi_j^{j'}[t] = \frac{\sum_{j'=0,j'\neq j}^{M-1}\alpha_{j,j'}[t]P_j'[t]G_j^{j'}[t]}{\sum_{j''=0,j''\neq j,j'}^{M-1}\alpha_{j'',j'}[t]P_{j''}'[t]G_{j''}^{j'}[t]+\sigma^2}, \tag{10}$$

where $P_j'[t]$ is the transmission power of UAV j, with $0 \le P_j'[t] \le P_j^{'max}$, and $\sum_{j''=0,j''\neq j,j'}^{M-1}\alpha_{j'',j'}[t]P_{j''}'[t]G_{j''}^{j'}[t]$ denotes the communication interference from other UAVs when UAV j communicates with UAV j'.

Therefore, the data transmission rate of UAV j is given by:

$$R_j^{j'}[t] = W\log_2\left(1+\chi_j^{j'}[t]\right), \tag{11}$$

where W is the transmission bandwidth of UAV.

C. Blockchain System Model

We use blockchain technology to record all transaction information of the UAVs in the aerial layer. In this layer, UAVs achieve resource sharing and decentralized computing through the blockchain system. We combine blockchain technology with queuing theory to achieve secure, reliable, and efficient distributed computing task offloading. In the blockchain network, all nodes have the capability to collect transaction data from UAVs in the blockchain-enabled MEC system. To ensure the efficiency and security of the system, we select a subset of nodes as consensus nodes (also known as validators or miners), which are responsible for generating new blocks and verifying the validity of transactions. The selection of consensus nodes is based on their available resources and the credibility value reflected by their historical behavior. Then, we design a abnormal node identification model based on blockchain technology to identify and isolate abnormal nodes.

1) Reputation Value Model: We design a abnormal node (malicious attacks, insufficient performance) identification model based on blockchain to grade the reputation value of each UAV node. Specifically, the blockchain adopts the Delegated Proof of Stake (DPoS) consensus protocol. In this model, each UAV node is assigned a reputation value $\Gamma_j[t]$. CE devices and UAVs evaluate the service quality provided by other UAVs and send the obtained reputation value $\Gamma_j[t]$ assessments to active miners. The active miners conduct weighted averaging on the received reputation values to update the reputation values, as follows:

$$\Gamma_j[t+1] = \alpha\frac{\sum_i^I\Gamma_{i,j}[t]+\sum_j^J\Gamma_{j,j'}[t]}{I+J}+(1-\alpha)\Gamma_j[t], \tag{12}$$

where α is a weighting factor that controls the balance between current and historical reputation. The current reputation is composed of ratings from CE devices and other UAVs, while each UAV rates its own reputation as zero.

The reputation value model maintains the reputation score of each node. Newly added nodes are usually only granted a little power after accumulating enough reputation, preventing malicious nodes from generating multiple identities to disrupt the network. The reputation value model is dynamic, and the reputation scores of nodes will be updated over time and with changes in behavior. If a node exhibits malicious behavior, its reputation score will decrease, reducing its influence on the network. Additionally, the model does not require frequent blockchain verification at the level of DPoS and only calculates and updates the reputation value of the UAV within each time slot. These reputation values are used as part of the system offloading decision to ensure the reliability of UAV collaboration. This reduces the latency of blockchain verification and guarantees the performance of the system.

2) Abnormal Node Identification Model: We set a minimum reputation value Γ_{min} to prevent tasks from being offloaded to abnormal nodes, which can be represented as follows:

$$\partial_{i,j}=0, \alpha_{j,j'}=0, \forall \quad \Gamma_j[t] \le \Gamma_{min} \tag{13}$$

In the subsequent algorithm design, users select a suitable UAV for offloading based on factors such as reputation score and queue backlog. The UAV-edge layer also collaboratively process tasks based on the reputation scores.

D. Task and Queue Model

We suppose that the tasks generated on CE devices have randomness and separability, measured in data bits, and each CE device and UAV maintains a buffer queue to handle tasks.

Let $l_i^c[t]$ denote the size of task processed locally by CE device i, then we have:

$$l_i^c[t] = \frac{f_i[t]\tau}{c_i} \leq \frac{f_i^{max}\tau}{c_i}, \tag{14}$$

where c_i is used to represent the number of CPU cycles required for CE device i to compute one bit of task, and $f_i[t]$ denotes the CPU frequency cycle of CE device i. Additionally, the maximum value of the CPU frequency cycle for CE device i is denoted as f_i^{max}, such that $f_i[t] \leq f_i^{max}$.

Due to the limited data transmission capacity of wireless channels, let $a_i[t] \in [0, a_i^{max}]$ represent the arriving data bits of the CE device in each time slot, and $l_i^o[t]$ as the offloading task bits. Therefore, there are constraints as follows:

$$l_i^o[t] \leq r_i^j[t]\tau. \tag{15}$$

The evolution of the task queue backlog [44] for CE device i, represented by $Q_i[t]$, is as follows:

$$Q_i[t+1] = max\{Q_i[t] - l_i^c[t] - l_i^o[t], 0\} + a_i[t]. \tag{16}$$

Similarly, $F_j[t]$ represents the CPU frequency of UAV j, with its peak value being F_j^{max}, such that $F_j[t] \leq F_j^{max}$. Using C_j to denote the number of CPU cycles required for UAV j to compute one bit of data, the size of the task processed locally is given by:

$$L_j^c[t] = \frac{F_j[t]\tau}{C_j} \leq \frac{F_j^{max}\tau}{C_j}, \tag{17}$$

the total task load received by UAV j in t-th is described as $A_j[t]$. The UAV j has a maximum payload limit of U_j, thus we have constraint:

$$0 \leq A_j[t] = \sum_{i=0}^{N-1} \partial_{i,j} l_i^o[t] + \sum_{j'=0, j' \neq j}^{M-1} \alpha_{j',j}[t] h_{j'}^o[t] \leq U_j, \tag{18}$$

where $\sum_{i=0}^{N-1} \partial_{i,j} l_i^o[t]$ represents the offloading amount received by UAV j from CE devices, and $\sum_{j'=0, j' \neq j}^{M-1} \alpha_{j',j}[t] h_{j'}^o[t]$ represents the offloading amount received by UAV j from other UAVs. The size of the task that needs to be offloaded is denoted by $L_j^o[t]$, and it satisfies the following constraint:

$$L_j^o[t] \leq R_j^{j'}[t]\tau. \tag{19}$$

Similarly, each UAV also maintains a task queue to store unprocessed tasks. The queue backlog is represented by $X_j[t]$, and the update of the task queue backlog is as follows:

$$X_j[t+1] = max\left\{Q_j[t] - L_j^c[t] - L_j^o[t], 0\right\} + A_j[t]. \tag{20}$$

To ensure that the task queue remains in a stable state over the long term, we introduce the following constraints:

$$\lim_{T\to\infty} \sum_{t=0}^{T-1} \frac{E\{Q_i[t]\}}{T} = 0, \tag{21}$$

$$\lim_{T\to\infty} \sum_{t=0}^{T-1} \frac{E\{X_j[t]\}}{T} = 0, \tag{22}$$

where $E\{Q_i[t]\}$ and $E\{X_j[t]\}$ represent the expected lengths of the task queues.

E. Energy Consumption Model

In the system model, we consider the energy consumption of both CE devices and UAVs. The energy consumption generated by CE devices can be divided into two main parts. One part is the energy consumption of local computation, which mainly comes from CPU computation and is related to the CPU hardware architecture [45]. It can be expressed as follows:

$$E_{CE,i}^{comp}[t] = \kappa \cdot \tau \cdot (f_i[t])^3, \tag{23}$$

where κ is the effective switched capacitance of the CPU. The other part is the energy consumption associated with task offloading, which arises when CE devices offloading tasks to UAVs, and it can be denoted:

$$E_{CE,i}^{offload}[t] = P_i \frac{l_i^o[t]}{r_i^j[t]}. \tag{24}$$

Combining the two parts, the total energy consumption generated by CE devices in time slot t is:

$$E_{CE}[t] = \sum_{i=0}^{N-1} \left(E_{CE,i}^{comp}[t] + E_{CE,i}^{offload}[t]\right). \tag{25}$$

Similarly, the energy consumption generated by UAVs is also divided into two parts: local computation energy consumption and task offloading energy consumption. Local computation energy consumption comes from CPU computation [46], while task offloading energy consumption is related to the energy consumption for transmission. Therefore, the calculations are as follows:

$$E_{UAV,j}^{comp}[t] = \gamma \cdot \tau \cdot \left(F_j[t]\right)^3, \tag{26}$$

$$E_{UAV,j}^{offload}[t] = \sum_{j'=0, j' \neq j}^{M-1} \alpha_{j,j'}[t] P_j' \frac{L_j^o[t]}{R_j^{j'}[t]}. \tag{27}$$

where γ is the effective switched capacitance of the CPU.

Hence, we can obtain the total energy consumption of UAVs as follows:

$$E_{UAV}[t] = \sum_{j=0}^{M-1} \left(E_{UAV,j}^{comp}[t] + E_{UAV,j}^{offload}[t]\right). \tag{28}$$

In the system model, we use energy consumption as the system cost, calculated as:

$$P[t] = E_{CE}[t] + E_{UAV}[t]. \tag{29}$$

F. Problem Formulation

Our objective is to achieve the minimum energy consumption of the system while ensuring the stability of the system's queue. We can formalize the problem as follows:

$$P1: \min_{\Omega[t]} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{P[t]\}, \tag{30a}$$

$$s.t. \quad 0 \leq f_i[t] \leq f_i^{max}, \forall i \in \mathcal{N}, \tag{30b}$$

$$0 \leq F_j[t] \leq F_j^{max}, \forall j \in \mathcal{M}, \tag{30c}$$

(1), (15), (18), (19), (21), (22).

where $\Omega[t] = \{f_i[t], l_i^o[t], F_j[t], L_j^o[t], \alpha_{j,j'}\}$ is the set of optimization variables, (30b) and (30c) represent the CPU frequency constraints for CE devices and UAVs respectively, (1) is the decision constraint for UAVs, (18) ensure that the tasks generated and received by UAVs do not exceed their load limits, while (15) and (19) ensure that the size of task offloaded falls within the range of the data that can be transmitted.

Problem $P1$ is a stochastic optimization problem, and as the number of CE devices increases, the unpredictability associated with factors such as task generation and wireless channel states also increases. Making decisions at each time slot without future information makes it difficult to maintain stability in the long-term queues. Additionally, the task offloading and the decision for UAVs are coupled, adding to the complexity of the problem. To address this issue, we introduce stochastic optimization techniques to transform this stochastic optimization problem into a deterministic one, providing a better solution for this complex scenario. It is worth noting that the problem $P1$ can be expressed as a MINLP problem that has been proven to be NP-hard and cannot be solved using conventional mathematical methods (Detailed proof can be found in Appendix A) [12].

IV. Blockchain-Based Multi-UAV Cooperative Task Offloading Algorithm Design

In this section, to solve the original dynamic stochastic problem $P1$, we utilize stochastic optimization techniques to convert it into a deterministic optimization problem within a single time slot. Subsequently, we design an energy-efficient online algorithm called BMCTO, which decouples the problem into five online subproblems and applies a series of optimization methods to solve these subproblems. This allows us to make decisions based solely on current information without the need for future system information.

A. Problem Transformation

The queues of all CE devices and UAVs in a time slot are represented by a Lyapunov function [44], defined as follows:

$$L(\Theta[t]) = \frac{1}{2}\left(\sum_{i=0}^{N-1} Q_i^2[t] + \sum_{j=0}^{M-1} X_j^2[t]\right), \tag{31}$$

where $\Theta[t] = \{Q_0[t], Q_1[t], Q_2[t], \ldots, Q_{N-1}[t], X_0[t], X_1[t], X_2[t], \ldots, X_{M-1}[t]\}$ represents the queue backlog vector. The initial queue backlog is zero. As the task queue backlogs change, $L(\Theta[t])$ also changes accordingly. However, excessive queue backlog can affect the stability of the system. To maintain system stability, we define the drift function as:

$$\Delta(\Theta[t]) = E\{L(\Theta[t+1]) - L(\Theta[t])|\Theta[t]\}. \tag{32}$$

To minimize the system cost while ensuring queue stability, we define the drift-plus-penalty function as:

$$\Delta(\Theta[t]) + VE\{P[t]|\Theta[t]\}, \tag{33}$$

where V is a parameter that balances queue stability and optimality of the cost function.

Although our goal is to minimize the energy cost of the system, the stability of the queue cannot be ignored during the optimization process. To comprehensively balance these two factors, we can transform the optimization problem into:

$$\min_{\Omega[t]} \ \Delta(\Theta[t]) + VE\{P[t]|\Theta[t]\}. \tag{34}$$

Theorem 1: For any task arrival rate $a_i[t]$, under the constraint condition (13), the following equation holds:

$$\begin{aligned}\Delta(\Theta[t]) + VE\{P[t]|\Theta[t]\} \le S + VE\{P[t]|\Theta[t]\} \\ + \sum_{i=0}^{N-1} Q_i[t]E\left\{a_i[t] - \left(l_i^c[t] + l_i^o[t]\right)|\Theta[t]\right\} \\ + \sum_{j=0}^{M-1} X_j[t]E\left\{A_j[t] - \left(L_j^c[t] + L_j^o[t]\right)|\Theta[t]\right\},\end{aligned} \tag{35}$$

where $S = \frac{1}{2}\{\sum_{i=0}^{N-1}[(a_i^{max}[t])^2 + (\frac{f_i^{max}\tau}{c_i} + r_i^{j-max}\tau)^2] + \sum_{j=0}^{M-1}[U_j^2 + (\frac{F_j^{max}\tau}{C_j} + R_j^{j'-max}\tau)^2]\}$ is a constant.

Proof of Theorem 1: From the mathematical knowledge that $max\{a-b,0\}^2 \le (a-b)^2$, we can square both sides of equation (16) to obtain:

$$\begin{aligned}Q_i^2[t+1] \le Q_i^2[t] + a_i^2[t] + \left(l_i^c[t] + l_i^o[t]\right)^2 - 2\cdot Q_i[t] \\ \cdot \left(l_i^c[t] + l_i^o[t]\right) + 2\cdot Q_i[t]\cdot a_i[t].\end{aligned} \tag{36}$$

After rearranging, we can get:

$$\begin{aligned}\frac{1}{2}\left(Q_i^2[t+1] - Q_i^2[t]\right) \le \frac{1}{2}\left[a_i^2[t] + \left(l_i^c[t] + l_i^o[t]\right)^2\right] \\ + Q_i[t]\left[a_i[t] - \left(l_i^c[t] + l_i^o[t]\right)\right].\end{aligned} \tag{37}$$

Then, the queue backlog of UAVs can be rearranged as:

$$\begin{aligned}\frac{1}{2}\left(X_j^2[t+1] - X_j^2[t]\right) \le \frac{1}{2}\left[A_j^2[t] + \left(L_j^c[t] + L_j^o[t]\right)^2\right] \\ + X_j[t]\left[A_j[t] - \left(L_j^c[t] + L_j^o[t]\right)\right].\end{aligned} \tag{38}$$

Substituting equations (37) and (38) into equation (33), we get:

$$\begin{aligned}\Delta(\Theta[t]) \le \frac{1}{2}\sum_{i=0}^{N-1}\left[a_i^2[t] + \left(l_i^c[t] + l_i^o[t]\right)^2\right] \\ + \frac{1}{2}\sum_{j=0}^{M-1}\left[A_j^2[t] + \left(L_j^c[t] + L_j^o[t]\right)^2\right] \\ + \sum_{i=0}^{N-1} Q_i[t]E\left\{a_i[t] - \left(l_i^c[t] + l_i^o[t]\right)|\Theta[t]\right\} \\ + \sum_{j=0}^{M-1} X_j[t]E\left\{A_j[t] - \left(L_j^c[t] + L_j^o[t]\right)|\Theta[t]\right\}.\end{aligned} \tag{39}$$

According to constraints (18), (30b), (30c), and $r_i^j[t] \le r_i^{j-max}$, $R_j^{j'}[t] \le R_j^{j'-max}$, we can get equation (35). ■

According to Theorem 1, it can be known that there exists an upper bound for equation (33), and our optimization

goal can be achieved by minimizing this upper bound. After ignoring the unnecessary constants and variables, problem $P1$ can be transformed into problem $P2$:

$$\begin{aligned} P2: \min_{\Omega[t]} \; & VE\{P[t]\} - \sum_{i=0}^{N-1} Q_i[t]\left(l_i^c[t] + l_i^o[t]\right) \\ & + \sum_{j=0}^{M-1} X_j[t]\left(A_j[t] - \left(L_j^c[t] + L_j^o[t]\right)\right), \\ s.t. \quad & 0 \le f_i[t] \le f_i^{max}, \forall i \in \mathcal{N}, \\ & 0 \le F_j[t] \le F_j^{max}, \forall j \in \mathcal{M}, \\ & (1), (15), (18), (19), (21), (22). \end{aligned} \tag{40}$$

B. Blockchain-Based Multi-UAV Cooperative Task Offloading Algorithm

To successfully solve problem $P2$, we design an energy-efficient algorithm that decomposes the optimization problem into five independent subproblems based on different decision variables for sequential resolution. It is important to note that while solving subproblem $P2.4$, the dynamic coupling between the decision of UAVs and the allocation of tasks and transmission rates is addressed by introducing a dynamic priority scoring algorithm. Furthermore, the solution to subproblem $P2.5$ relies on the optimal solution obtained from subproblem $P2.4$.

1) Local Computation Resource Allocation of CE device: We decompose the subproblem $P2.1$ based on the decision variable $f_i[t]$ as follows::

$$\begin{aligned} P2.1: \min_{f_i[t]} \; & \sum_{i=0}^{N-1} \left(-\frac{Q_i[t]\tau}{c_i} f_i[t] + V\kappa \cdot \tau \cdot f_i^3[t] \right), \\ s.t. \quad & 0 \le f_i[t] \le f_i^{max}, \forall i \in \mathcal{N}, (21). \end{aligned} \tag{41}$$

It is obvious that this subproblem is a simple convex optimization problem. We can transform the problem into finding the minimum value of a polynomial function within a given domain. Since equation (41) is a polynomial function of $f_i[t]$ over a continuous domain, its optimal value $f_i^*[t]$ can be obtained at the boundary point or critical point. We can calculate the first-order derivative of the equation and set it equal to zero to find the critical point. Additionally, we need to consider CE devices' task backlog. When the size of the task that CE device can still compute locally is $Q_i[t]$, the CPU frequency at this time is $\frac{Q_i[t]c_i}{\tau}$. Therefore, we can obtain the optimal value $f_i^*[t]$ as:

$$f_i^*[t] = \min\left\{ \sqrt{\frac{Q_i[t]}{3V\kappa c_i}}, \frac{Q_i[t]c_i}{\tau}, f_i^{max} \right\}. \tag{42}$$

Analyzing equation (42), we can see that $f_i^*[t]$ is directly proportional to $Q_i[t]$. This result can be explained by the fact that as $Q_i[t]$ increases, more CPU resources are needed to prevent excessive queue backlog. We substitute $f_i^*[t]$ into equation (14) to calculate the local computation load of CE devices.

2) Offloading Resource Allocation of CE Device: To obtain the size of task offloaded by CE device i, we solve subproblem $P2.2$:

$$\begin{aligned} P2.2: \min_{l_i^o[t]} \; & \sum_{i=0}^{N-1} \left(-Q_i[t] l_i^o[t] + \frac{VP_i}{r_i^j[t]} l_i^o[t] \right) + \sum_{j=0}^{M-1} X_j[t] \\ & \cdot \sum_{i=0}^{N-1} \partial_{i,j} l_i^o[t], \\ s.t. \quad & (15), (21). \end{aligned} \tag{43}$$

Since this is a linear programming problem, with all parameters fixed except the objective variable $l_i^o[t]$, we can determine the offloading resource allocation for CE device as:

$$l_i^o[t]^* = \begin{cases} \min\left\{ r_i^j[t]\tau, Q_i(t) \right\}, & \frac{VP_i}{r_i^j[t]} - Q_i[t] + \\ & \sum_{j=0}^{M-1} X_j[t] \sum_{i=0}^{N-1} \partial_{i,j} \le 0, \\ 0, & otherwise. \end{cases} \tag{44}$$

3) Local Computation Resource Allocation of UAV: For the solution of local resource allocation for UAV, we simplify the problem to solving subproblem $P2.3$:

$$\begin{aligned} P2.3: \min_{F_j[t]} \; & \sum_{j=0}^{M-1} \left(-\frac{X_j[t]\tau}{C_j} F_j[t] + V \cdot \gamma \cdot \tau \cdot F_j^3[t] \right), \\ s.t. \quad & 0 \le F_j[t] \le F_j^{max}, (22). \end{aligned} \tag{45}$$

As subproblem $P2.3$ is a simple convex optimization problem, and equation (45) is a convex function concerning $F_j[t]$, the solution method is similar to subproblem $P2.1$. After finding the critical points of equation (45), we compare them with the CPU frequency handling the backlog of tasks for UAV at that moment to obtain the optimal value $F_j^*[t]$ as:

$$F_j^*[t] = \min\left\{ \sqrt{\frac{X_j[t]}{3V\gamma C_j}}, \frac{X_j[t]C_j}{\tau}, F_j^{max} \right\}. \tag{46}$$

4) Offloading Resource Allocation of UAV: Subproblem $P2.4$ is as follows:

$$\begin{aligned} P2.4: \min_{\alpha_{j,j'}, L_j^o[t]} \; & V \sum_{j=0}^{M-1} \sum_{j'=0, j' \ne j}^{M-1} \alpha_{j,j'}[t] P_j' \frac{L_j^o[t]}{R_j^{j'}[t]} \\ & - \sum_{j=0}^{M-1} X_j[t] \sum_{j'=0, j' \ne j}^{M-1} \alpha_{j,j'}[t] L_j^o[t], \\ s.t. \quad & (1), (19), (22). \end{aligned} \tag{47}$$

In equation (47), we can see that the size of offloaded tasks is highly coupled with the decision of the UAV. It is worth noting that the UAV's decision is also coupled with the transmission rate. This makes problem solving challenging. To solve problem $P2.4$, we need to perform a decoupling operation on problem $P2.4$, i.e., determine the UAV's decision and then optimize the offloading task size.

Since the uncertainty of the system, the execution efficiency of each UAV varies. Therefore, how to select the UAV with the minimum cost in each time slot is the most critical issue in the UAV offloading process. Therefore, we adopt a dynamic

priority scoring algorithm, which evaluates a score for each UAV in every time slot. When a UAV needs to offload tasks, it selects the UAV with the highest score from the cluster of UAVs within the communication range as the offloading target. The UAV's score is determined by considering the waiting delay of tasks in the UAV's queue, the reputation value, and the offloading distance. The size of the waiting delay reflects the backlog of tasks in the UAV's queue. The smaller the offloading distance of the UAV, the smaller the transmission delay, and it can offload more tasks under the same transmission rate.

We provide the calculation method for scoring factors:

$$D_j^{score}[t] = \frac{D^{max} - D_j^{j'}[t]}{D^{max}} \times 100, \tag{48}$$

$$T_j^{score}[t] = \frac{\tau - T_j^{wait}[t]}{\tau} \times 100, \tag{49}$$

$$\Gamma_j^{score}[t] = \Gamma_j[t] \times 100, \tag{50}$$

where $T_j^{wait}[t]$ is the waiting delay, which can be expressed as the queue backlog before task arrival in each time slot divided by the local computation rate, that is, $T_j^{wait}[t] = \frac{X_j[t]C_j}{F_j^{max}[t]}$. $D_j^{score}[t]$ represents the offloading distance score, and $T_j^{score}[t]$ is the waiting delay score. $\Gamma_j[t]$ is the same as the meaning in (12)

Based on the scoring factors, here is the formula for calculating the priority score:

$$Score_j[t] = w_1 D_j^{score}[t] + w_2 T_j^{score}[t] + w_3 \Gamma_j^{score}[t], \tag{51}$$

where w_1, w_2, and w_3 represent weighting coefficients. The weight values here are dynamically variable to adapt to the dynamism of the system and the diversity of task types. The weight values of the scoring factors are influenced by the type of task, such as the transmission efficiency, urgency, security, and privacy of the task. We can obtain the optimal value of weights through convex optimization methods.

By using offloading decisions as optimization variables, we effectively reduce the coupling between components in the system, making the system more resilient to individual component failures and ensuring the system can continue to operate stably even with partial component failures. To address the UAV decision problem, we provide Algorithm 1.

According to Algorithm 1, we can obtain the decision for UAV j and determine the target UAV j'. Furthermore, according to the constraint $\sum_{j'=0}^{M-1} \alpha_{j,j'}[t] = 1$, we can decouple $L_j^o[t]$ and $R_j^{j'}[t]$ from the decision, transforming problem $P2.4$ into:

$$P2.4.1: \min_{L_j^o[t]} \sum_{j=0}^{M-1} \left(-X_j[t] L_j^o[t] + \frac{VP'_j}{R_j^{j'}[t]} L_j^o[t] \right), \quad s.t. \quad (19), (22). \tag{52}$$

The problem can be regarded as a linear programming problem, and the results are obtained according to the constraints of the decision variables as follows:

$$L_j^o[t]^* = \begin{cases} \min\left\{ R_j^{j'}[t]\tau, X_j[t] \right\}, & \frac{VP'_j}{R_j^{j'}[t]} - X_j[t] \le 0, \\ 0, & otherwise. \end{cases} \tag{53}$$

Algorithm 1: Dynamic Priority Scoring Algorithm

Input: $\mathcal{M}, \tau, C_j[t], F_j^{max}[t], D^{max}$, the positions of UAVs, the queue backlog $X_j[t]$ of UAV
Initialization:
$X_j[t] \leftarrow 0$, $(F_j^{max}[t], D^{max})$ ←Fixed value, $\Gamma_j^{score}[t] \leftarrow 0$, $C_j[t] \leftarrow$ random()
End Initialization
for *each UAV* $j \in \mathcal{M}$ **do**
  Calculate the value of D_j^{score}, $T_j^{score}[t]$, and $\Gamma_j^{score}[t]$
  Calculate $Score_j[t]$ according to the process (6)
  Update the queues $X_j[t]$ and the positions of UAVs
for *each UAV* $j \in \mathcal{M}$ **do**
  Select the UAV $j'(j' \neq j)$ with the highest score
  Assign the decision $\alpha_{j,j'}[t]$ a value of 0 or 1
Update the value of D_j^{score}, $T_j^{score}[t]$, and $\Gamma_j^{score}[t]$.
Output: The decision $\alpha_{j,j'}[t]$ for UAVs

5) Offloading Resources From Collaborative UAVs:

$$P2.5: \min_{h_{j'}^o[t]} \sum_{j=0}^{M-1} \left(X_j[t] \cdot \left(\sum_{j'=0, j' \neq j}^{M-1} \alpha_{j',j}[t] h_{j'}^o[t] \right) \right), \quad s.t. \quad (1), (19), (22). \tag{54}$$

We find that the offloading task size and decision are also coupled. According to Algorithm 1, we determine the decision. It is worth noting that at this point, the decision is for UAV j' to offload to UAV j, i.e., $\sum_{j'=0, j' \neq j}^{M-1} \alpha_{j',j}[t] > 1$, but this does not affect the execution result. We substitute the optimal offloading task size $L_{j'}^o[t]^*$, into the equation (54) to obtain the optimal solution:

$$\sum_{j=0}^{M-1} \left(X_j[t] \cdot \left(\sum_{j'=0, j' \neq j}^{M-1} \alpha_{j',j}[t] L_{j'}^o[t]^* \right) \right). \tag{55}$$

Algorithm 2 provides the detailed steps of the BMCTO algorithm.

We give the complexity analysis of the proposed algorithm. Algorithm 1 solves the continuous problem of each time slot in two steps: First, by traversing M UAVs to calculate the priority score for each one and update their queue backlog and location information. The algorithm complexity of this part is $O(M)$. Second, we iteratively solve the decision for each UAV, and the algorithm complexity is $O(M^2)$. In summary, the complexity of Algorithm 1 is $O(M^2)$. For Algorithm 2, it is necessary to iterate over T time slots, and calculate the optimal CPU frequency and the optimal offloading task size by traversing N CE devices and M UAVs. Thus, the complexity of the proposed BMCTO algorithm is $O(T(N + M^2))$.

Fig. 2 illustrates the detailed solving process of the BMCTO algorithm. The algorithm is divided into three parts and executed in parallel. Obtain the optimal CPU frequency and offloading task size for the CE devices layer and UAVs layer by solving subproblems. At the same time, calculate the reputation value of UAVs in the blockchain and identify abnormal nodes. Provide decisions for UAVs based on dynamic priority scoring algorithm.

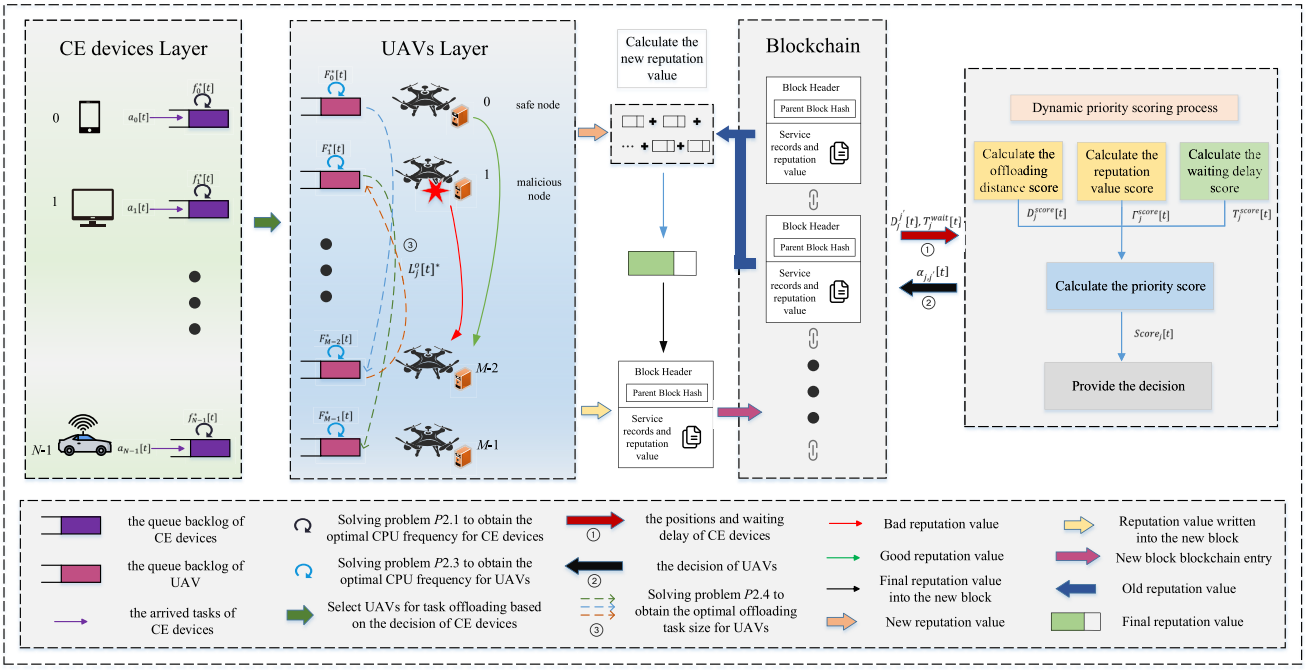

Fig. 2. Schematic diagram of BMCTO algorithm execution in CE devices layer, UAVs layer and blockchain.

Algorithm 2: Blockchain-Based Cooperative Task Offloading and Resource Allocation Algorithm

Input: $\mathcal{N}, \mathcal{M}, \kappa, \gamma$
Output: $f_i[t], l_i^o[t], F_j[t], L_j^o[t], \alpha_{j,j'}$
for *each* $t \in \mathcal{T}$ **do**
  Obtain $a_i[t]$
  for *each CE device* $i \in \mathcal{N}$ **do**
    Calculate $f_i[t]$ according to formula (42)
    Calculate $l_i^o[t]$ according to formula (44)
  Obtain decision $\alpha_{j,j'}[t]$ by Algorithm 1
  for *each UAV* $j \in \mathcal{M}$ **do**
    Substitute $\alpha_{j,j'}[t]$ into formulas (47) and (54)
    Calculate $F_j[t]$ according to formula (46)
    Calculate $L_j^o[t]$ according to formula (53)
  Update $Q_i[t+1], X_j[t+1], \alpha_{j,j'}[t+1]$

V. Performance Analysis of BMCTO Algorithm

To further explore the effectiveness and applicability of the BMCTO algorithm, we conduct a theoretical analysis of the algorithm. The analysis proves that the algorithm can maintain the gap between system energy consumption and queue backlog and the optimal solution within a limited range, and can be controlled by adjusting relevant parameters. Let $\overline{L}$ represent the average queue backlog, i.e.:

$$\overline{L} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \left(\sum_{i=0}^{N-1} E\{Q_i[t]\} + \sum_{j=0}^{M-1} E\{X_j[t]\} \right). \quad (56)$$

Lemma 1: For any information set $\lambda \in \Lambda$ within each time slot, there exists an optimal offloading policy μ^* such that the following formula holds:

$$\begin{aligned} &E\left\{P^{\mu^*}[t]\right\} = P^*[\lambda], \\ &E\{a_i[t]\} \le E\left\{l_i^{c,\mu^*}[t] + l_i^{o,\mu^*}[t]\right\}, \\ &E\{A_j[t]\} \le E\left\{L_j^{c,\mu^*}[t] + L_j^{o,\mu^*}[t]\right\}, \end{aligned} \quad (57)$$

Proof of Lemma 1: Lemma 1 can be proven using the Caratheodorys theorem [44], and we will not delve into the proof of Lemma 1 in detail here. ■

We know that there are constraints $0 \le a_i[t] \le a_i^{max}[t]$ and $0 \le A_j[t] \le U_j$. For the system energy cost $P[t]$, let its upper bound be $\hat{P}$ and lower bound be $\check{P}$. Through Lemma 1, we can derive Theorem 2, obtaining the gap between system energy consumption and queue backlog and the optimal solution.

Theorem 2: If there exists an $\epsilon \ge 0$ and the information set satisfies $\lambda + \epsilon \in \Lambda$, then the average queue backlog satisfies the following inequality:

$$\overline{L} \le \frac{S + V\left(\hat{P} - \check{P}\right)}{\epsilon}. \quad (58)$$

The energy consumption gap between the BMCTO algorithm and the optimal value is within a specific range:

$$P^{BMCTO} \le P^* + \frac{S}{V}, \quad (59)$$

where S is a constant.

Proof of Theorem 2: Please refer to Appendix B. ■

VI. Experiment Evaluation

To comprehensively evaluate the performance of the BMCTO algorithm, this section conducts a detailed analysis through simulation experiments. We analyze how changes in

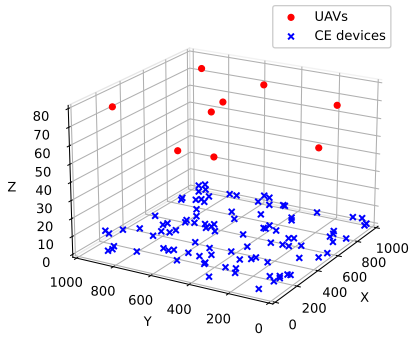

Fig. 3. Spatial distribution of UAVs and CE devices.

TABLE II
PARAMETER SETTINGS

Parameters	Value
Maximum CPU frequency of UAV	20 GHz
Maximum CPU frequency of CE device	1 GHz
Effective switched capacitance κ, γ	10^{-27}
Channel noise power σ	10^{-13}W
Bandwidth of UAV	2 MHz
Bandwidth of CE device	1 MHz
CPU cycles needed to process 1 bit data of UAV	500 cycles/bit
CPU cycles needed to process 1 bit data of CE device	1000 cycles/bit
Carrier frequency f_c	0.1 GHz
Link loss ξ_i^{LoS}, ξ_i^{NLoS}, ζ_j^{LoS}, ζ_j^{NLoS}	0.1, 21, 0.1, 21
Environment parameter ϕ, ϱ,$\varkappa$, φ	4.88, 0.43, 4.88, 0.43
Transmission power of UAV	0.05W
Transmission power of CE device	0.01W

different parameters, such as the penalty coefficient V, the number of CE devices, the task arrival rate and the number of UAVs affect the energy consumption and queue backlog of the system. In addition, by comparing the BMCTO algorithm with other algorithms, we validate the effectiveness of the BMCTO algorithm.

A. Experiment Settings

The experiments consider the deployment of 9 UAVs in a 1000 m $\times$ 1000 m square area to serve multiple CE devices. We also evenly distribute the vertical distance of UAVs from the ground at [60, 80]m. Each UAV is uniformly distributed in the area in a hovering manner, while the CE devices are randomly distributed. The maximum actual distance between UAVs is 1415m, and the task arrival rate for each CE device is set at $[0, 2] \times 10^6$ bits. Fig. 3 shows the positions of CE devices and UAVs, in which the blue cross is the positions of the CE devices and the red dot is the positions of UAVs. We set the time slot sum to 2000. Next, the speed of light and carrier frequency are set the value is 10^8 m/s and 10^8 HZ. Table II presents the settings of the experimental parameters [47].

B. Parameter Analysis

1) Impact of Parameter V: In our study, V is a parameter that balances queue stability and optimality of the cost function. The size of V can be used to adjust the optimization strength of energy consumption or queue. We have given the relationship between the different values of parameter V and the energy consumption and queue backlog of the system. As

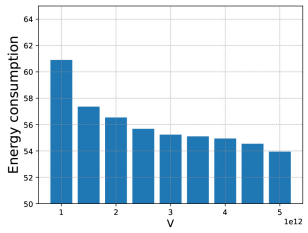

Fig. 4. Energy consumption with different values of parameter V.

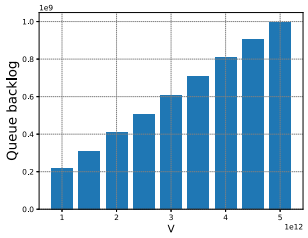

Fig. 5. Queue backlog with different values of parameter V.

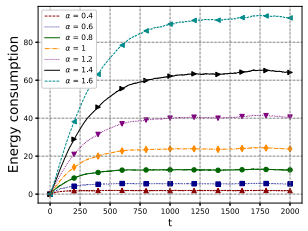

Fig. 6. Energy consumption with different task arrival rates.

depicted in Fig. 4, with the continuous increase of parameter V, the energy consumption consistently decreases and tends to plateau. This trend aligns with the results from Equation (59), as the larger parameter V, the smaller the gap between the energy consumption and the optimal value, prompting the BMCTO algorithm to dynamically adjust the allocation of resources to approach the optimal energy consumption. Fig. 5 reveals that the queue backlog steadily increases as parameter V rises, a trend consistent with equation (58). The queue backlog in the experiments represents the cumulative backlog of both UAVs and CE devices. Due to their limited computational and transmission capabilities, when the backlog reaches a certain threshold, it may impact system stability. Therefore, adjusting parameter V is necessary to maintain stable queues. Through the parameter analysis experiment, we select an appropriate parameter $V = 3 \times 10^{12}$ for subsequent experiments.

2) Impact of Task Arrival Rate: In this experiment, the task arrival rate is set as $\alpha \cdot a_i[t]$, with values of α: 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, and 1.6. Overall, as time progresses, the queue backlog and energy consumption of the system are able to converge rapidly and stabilize under varying task arrival rates. This indicates that the BMCTO algorithm can promptly adapt to different task arrival rates, demonstrating its strong adaptability. Fig. 6 shows that energy consumption rises as α increases. As more tasks arrive in the system, more tasks need to be executed, resulting in greater energy consumption for task processing. Fig. 7 indicates that queue backlog increases with α. With the processing and transmission capabilities of

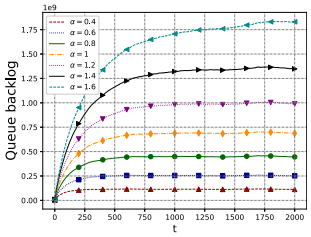

Fig. 7. Queue backlog with different task arrival rates.

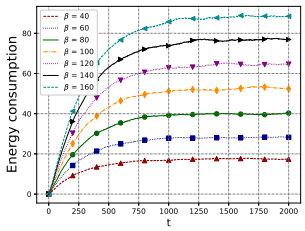

Fig. 8. Energy consumption with different numbers of CE devices.

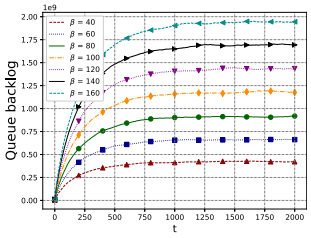

Fig. 9. Queue backlog with different numbers of CE devices.

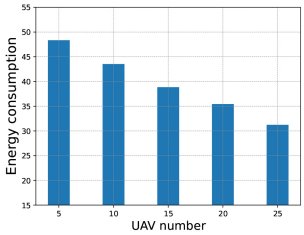

Fig. 10. Energy consumption with different numbers of UAV.

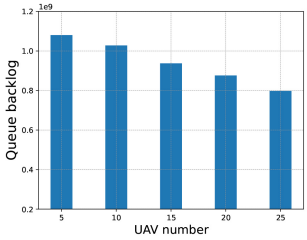

Fig. 11. Queue backlog with different numbers of UAV.

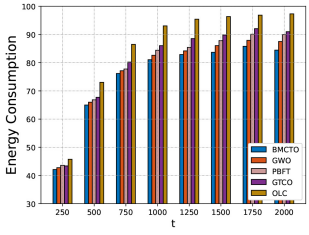

Fig. 12. Energy consumption with different algorithms.

CE devices and UAVs remaining unchanged, they are unable to process the increasing number of tasks promptly, resulting in a continuous accumulation of tasks in the queue and, hence, increased queue backlog.

3) Impact of Number of CE Device: We denote the quantities of CE devices as β, where β takes values of 40, 60, 80, 100, 120, 140, and 160. From Fig. 8, it can be seen that the energy consumption of the system is directly proportional to the number of CE devices, showing an upward trend. This is because the system energy consumption includes the portion generated by CE device processing tasks, which explains the trend of curve changes in Fig. 8. Fig. 9 shows the variation in queue backlog under different numbers of CE devices. Since the queue backlog in the experiment is the sum of the queue backlogs of both UAVs and CE devices, when an excessive number of CE devices access the system, the number of tasks will rise. This results in larger queue backlogs for CE devices and an increase in the tasks unloaded from CE devices to UAVs. This, in turn, causes a corresponding increase in UAV queue backlog, and ultimately, a rise in the overall queue backlog.

4) Impact of Number of UAV: Fig. 10 and Fig. 11 demonstrate the impact of different numbers of UAVs on energy consumption and queue backlog. As depicted in Fig. 10, energy consumption continuously decreases as the number of UAVs increases. This trend of change is reasonable. With the same number of CE devices connected to the system, the greater the number of UAVs, the better the system's ability to handle tasks and the less energy it consumes. The change in queue backlog in Fig. 11 also follows this trend. The ability of the system to process tasks is improved, and the generated tasks can be processed promptly, resulting in a decrease in queue backlog.

C. Comparative Experiments

In this subsection, to evaluate the performance and effectiveness of the BMCTO algorithm, we compare it with other algorithms.

- *Only Local Computing (OLC):* Tasks generated are computed only on the CE devices without any offloading operation.
- *GTCO:* The algorithm is extended from [48] to our model, where in each time slot, both CE devices and UAVs maximize offloading to achieve optimal resource allocation.
- *Grey Wolf Optimizer (GWO):* This algorithm [49] is applied to our model, selecting the energy consumption function of UAVs as the fitness function for each gray wolf. Through several iterations, it provides task scheduling results (the decisions of UAVs).
- *Practical Byzantine Fault Tolerance (PBFT):* In scenarios with up to 2k+1 faulty or malicious nodes [50], the PBFT protocol ensures that UAVs offload tasks to a consensus-achieving cluster of UAVs while excluding nodes that fail to reach consensus.

In Fig. 12 and Fig. 13, we plot the variation curves of system energy consumption and queue backlog for five

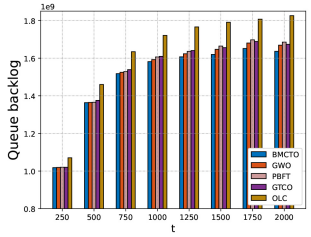

Fig. 13. Queue backlog with different algorithms.

algorithms. As shown in Fig. 12, the BMCTO algorithm is the most energy-efficient among the five algorithms. For the GTCO algorithm, as it selects task offloading in each time slot, pursuing immediate optimization, it overlooks the offloading energy consumption. In the long run, the energy consumption generated by offloading will increase the total energy consumption. For the OLC algorithm, due to the much lower computing power of CE devices compared to UAVs, processing tasks solely on CE devices will result in massive energy consumption. Meanwhile, when the number of tasks that can be processed approaches the maximum limit of CE devices, the energy consumption of the OLC algorithm tends to stabilize. The GWO algorithm demonstrates rapid convergence and low energy consumption in its early stages. However, as time progresses and the number of tasks within the system increases, GWO may encounter local optimal solutions that lead to increased energy consumption. PBFT requires multiple rounds of information exchange to achieve consensus, which increases communication overhead among UAVs and results in higher system energy consumption. The BMCTO algorithm dynamically adjusts the offloading strategy according to the current system state, resulting in relatively lower energy consumption.

Fig. 13 shows the queue backlog with different algorithms. Under the same experimental conditions, as the number of tasks in the system continues to increase, CE devices and UAVs are unable to process them in time, leading to a continuous increase of the queue backlog. Due to the GWO algorithm's tendency to converge toward an optimal solution (minimizing energy consumption) with each iteration, certain UAVs may experience excessive task loads and significant queue backlogs. Owing to the fault tolerance characteristics of PBFT, nodes that do not reach consensus are excluded from selection as offloading targets, resulting in an imbalance in load distribution among UAVs and further increasing queue backlogs. Since BMCTO can dynamically adjust resource allocation, it effectively reduces queue backlog, whereas the other algorithms do not significantly reduce it. By comprehensively analyzing the trends in Fig. 12 and Fig. 13, the BMCTO algorithm can stabilize the queue backlog while reducing system energy consumption.

VII. Conclusion

In this paper, we investigated the problem of Multi-UAV cooperative task offloading in a blockchain-enabled MEC system for CE, aiming to jointly optimize the CPU cycle frequencies and offloading resources of UAVs and CE devices, and the decision of UAVs while minimizing the energy consumption cost and ensuring system performance. Subsequently, we transformed the problem into five subproblems using stochastic optimization techniques and solved them using convex optimization and linear programming. Furthermore, considering load, balancing, and offloading security, we designed a dynamic priority scoring strategy to achieve cooperative offloading among UAVs while reducing abnormal nodes occupying resources. Finally, the BMCTO algorithm was proposed, and its performance was evaluated, proving that the gap between the BMCTO algorithm and the optimal solution was constant. Simulation results demonstrated that the BMCTO algorithm could perform well in reducing system energy consumption while maintaining queue stability. In the future, we will study the problem of Multi-UAV assisted MEC task offloading equipped with energy harvesting CE devices.

References

[1] Z. Zhang, F. Zhang, Z. Xiong, K. Zhang, and D. Chen, "LsiA3CS: Deep-reinforcement-learning-based cloud–edge collaborative task scheduling in large-scale IIoT," *IEEE Internet Things J.*, vol. 11, no. 13, pp. 23917–23930, Jul. 2024.

[2] J. Huang, B. Ma, Y. Wu, Y. Chen, and X. Shen, "A hierarchical incentive mechanism for federated learning," *IEEE Trans. Mobile Comput.*, early access, Jul. 4, 2024, doi: 10.1109/TMC.2024.3423399.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[4] X. Zhou et al., "Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3191–3211, Oct. 2023.

[5] Y. Li, J. Shen, P. Vijayakumar, C.-F. Lai, A. Sivaraman, and P. K. Sharma, "Next-generation consumer electronics data auditing scheme toward cloud–edge distributed and resilient machine learning," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2244–2256, Feb. 2024.

[6] H. Ma, R. Li, X. Zhang, Z. Zhou, and X. Chen, "Reliability-aware online scheduling for DNN inference tasks in mobile-edge computing," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11453–11464, Jul. 2023.

[7] J. Huang et al., "Incentive mechanism design of federated learning for recommendation systems in MEC," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2596–2607, Feb. 2024.

[8] Y. Chen, J. Hu, J. Zhao, and G. Min, "QoS-aware computation offloading in leo satellite edge computing for IoT: A game-theoretical approach," *Chin. J. Electron.*, vol. 33, no. 4, pp. 875–885, Jul. 2024.

[9] T.-X. Zheng, X. Chen, Y. Wen, N. Zhang, D. W. K. Ng, and N. Al-Dhahir, "Secure offloading in NOMA-enabled multi-access edge computing networks," *IEEE Trans. Commun.*, vol. 72, no. 4, pp. 2152–2165, Apr. 2024.

[10] N. Huang et al., "Mobile edge computing aided integrated sensing and communication with short-packet transmissions," *IEEE Trans. Wireless Commun.*, vol. 23, no. 7, pp. 7759–774, Jul. 2024.

[11] J. Huang, F. Liu, and J. Zhang, "Multi-dimensional QoS evaluation and optimization of mobile edge computing for IoT: A survey," *Chin. J. Electron.*, vol. 33, no. 4, pp. 859–874, Jul. 2024.

[12] L. Tan, Z. Kuang, L. Zhao, and A. Liu, "Energy-efficient joint task offloading and resource allocation in OFDMA-based collaborative edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1960–1972, Mar. 2022.

[13] P. Consul, I. Budhiraja, D. Garg, N. Kumar, R. Singh, and A. S. Almogren, "A hybrid task offloading and resource allocation approach for digital twin-empowered UAV-assisted MEC network using federated reinforcement learning for future wireless network," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3120–3130, Feb. 2024.

[14] Y. Chen, J. Zhao, Y. Wu, J. Huang, and X. Shen, "Multi-user task offloading in UAV-assisted LEO satellite edge computing: A game-theoretic approach," *IEEE Trans. Mobile Comput.*, early access, Sep. 24, 2024, doi: 10.1109/TMC.2024.3465591.

[15] L. Xing and B. W. Johnson, "Reliability theory and practice for unmanned aerial vehicles," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3548–3566, Feb. 2023.

[16] D. Diao, B. Wang, K. Cao, R. Dong, and T. Cheng, "Enhancing reliability and security of UAV-enabled NOMA communications with power allocation and aerial jamming," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8662–8674, Aug. 2022.

[17] X. Hou, Z. Ren, J. Wang, S. Zheng, and H. Zhang, "Latency and reliability oriented collaborative optimization for multi-UAV aided mobile edge computing system," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2020, pp. 150–156.

[18] C. Xu et al., "Energy consumption and time-delay optimization of dependency-aware tasks offloading for industry 5.0 applications," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1590–1600, Feb. 2024.

[19] Q. Qi, T. Shi, K. Qin, and G. Luo, "Completion time optimization in UAV-relaying-assisted MEC networks with moving users," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1246–1258, Feb. 2024.

[20] J. Huang, M. Zhang, J. Wan, Y. Chen, and N. Zhang, "Joint data caching and computation offloading in UAV-assisted Internet of Vehicles via federated deep reinforcement learning," *IEEE Trans. Veh. Technol.*, early access, Jul. 18, 2024, doi: 10.1109/TVT.2024.3429507.

[21] Y. Chen, K. Li, Y. Wu, J. Huang, and L. Zhao, "Energy efficient task offloading and resource allocation in air-ground integrated MEC systems: A distributed online approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8129–8142, Aug. 2024.

[22] D. Wang, Y. Jia, M. Dong, K. Ota, and L. Liang, "Blockchain-integrated UAV-assisted mobile edge computing: Trajectory planning and resource allocation," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 1263–1275, Jan. 2024.

[23] Y. Zhang, Y. Gan, C. Li, C. Deng, and Y. Luo, "Primary node selection based on node reputation evaluation for PBFT in UAV-assisted MEC environment," *Wireless Netw.*, vol. 29, no. 8, pp. 3515–3539, 2023.

[24] S. Datta and S. Namasudra, "Blockchain-based smart contract model for securing healthcare transactions by using consumer electronics and mobile-edge computing," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 4026–4036, Feb. 2024.

[25] X. Zhou et al., "Federated distillation and blockchain empowered secure knowledge sharing for Internet of Medical Things," *Inf. Sci.*, vol. 662, Mar. 2024, Art. no. 120217. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025524001300

[26] X. Zhou et al., "Hierarchical federated learning with social context clustering-based participant selection for Internet of Medical Things applications," *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 4, pp. 1742–1751, Aug. 2023.

[27] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for Internet of Things at the edge," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2737–2746, Apr. 2020.

[28] Z. Yang, S. Bi, and Y.-J. A. Zhang, "Dynamic offloading and trajectory control for UAV-enabled mobile edge computing system with energy harvesting devices," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10515–10528, Dec. 2022.

[29] X. Hu, K.-K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, Oct. 2019.

[30] A. Samy, I. A. Elgendy, H. Yu, W. Zhang, and H. Zhang, "Secure task offloading in blockchain-enabled mobile edge computing with deep reinforcement learning," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 4872–4887, Dec. 2022.

[31] M. Hosseinpour and M. H. Yaghmaee, "Quality-of-experience-aware computation offloading in MEC-enabled blockchain-based IoT networks," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14483–14493, Apr. 2024.

[32] D. Wang, Y. Jia, L. Liang, K. Ota, and M. Dong, "Resource allocation in blockchain integration of UAV-enabled MEC networks: A Stackelberg differential game approach," *IEEE Trans. Services Comput.*, early access, Jun. 24, 2024, doi: 10.1109/TSC.2024.3418330.

[33] T. Xiao, C. Chen, Q. Pei, and H. H. Song, "Consortium blockchain-based computation offloading using mobile edge platoon cloud in Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17769–17783, Oct. 2022.

[34] X. He, R. Jin, and H. Dai, "Multi-hop task offloading with on-the-fly computation for multi-UAV remote edge computing," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 1332–1344, Feb. 2022.

[35] X. Qi, J. Chong, Q. Zhang, and Z. Yang, "Collaborative computation offloading in the multi-UAV fleeted mobile edge computing network via connected dominating set," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10832–10848, Oct. 2022.

[36] Q. Wang, A. Gao, and Y. Hu, "Joint power and QoE optimization scheme for multi-UAV assisted offloading in mobile computing," *IEEE Access*, vol. 9, pp. 21206–21217, 2021.

[37] K. Zeng, X. Li, and T. Shen, "Energy-stabilized computing offloading algorithm for UAVs with energy harvesting," *IEEE Internet Things J.*, vol. 11, no. 4, pp. 6020–6031, Feb. 2024.

[38] B. Rana and Y. Singh, "Internet of Things and UAV: An interoperability perspective," in *Unmanned Aerial Vehicles for Internet of Things (IoT): Concepts, Techniques, and Applications*. Hoboken, NJ, USA: Wiley, 2021, pp. 105–127.

[39] S. S. Albouq, A. A. Abi Sen, N. Almashf, M. Yamin, A. Alshanqiti, and N. M. Bahbouh, "A survey of interoperability challenges and solutions for dealing with them in IoT environment," *IEEE Access*, vol. 10, pp. 36416–36428, 2022.

[40] Y. Yang, Y. Chen, K. Li, and J. Huang, "Carbon-aware dynamic task offloading in NOMA-enabled mobile edge computing for IoT," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15723–15734, May 2024.

[41] Y. Chen, J. Xu, Y. Wu, J. Gao, and L. Zhao, "Dynamic task offloading and resource allocation for noma-aided mobile edge computing: An energy efficient design," *IEEE Trans. Services Comput.*, vol. 17, no. 4, pp. 1492–1503, Jul./Aug. 2024.

[42] S. Zhu, L. Gui, N. Cheng, Q. Zhang, F. Sun, and X. Lang, "UAV-enabled computation migration for complex missions: A reinforcement learning approach," *IET Commun.*, vol. 14, no. 15, pp. 2472–2480, 2020.

[43] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[44] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan Claypool, 2010.

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

[46] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.

[47] X. Zhou, X. Yang, J. Ma, and K. I.-K. Wang, "Energy-efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14988–14997, Aug. 2022.

[48] Z. Luo and A. Huang, "Joint game theory and greedy optimization scheme of computation offloading for UAV-aided network," in *Proc. 31st Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, 2021, pp. 198–203.

[49] A. Satouf, A. Hamidoglu, O. M. Gul, and A. Kuusik, "Grey wolf optimizer-based task scheduling for IoT-based applications in the edge computing," in *Proc. 8th Int. Conf. Fog Mobile Edge Comput. (FMEC)*, 2023, pp. 52–57.

[50] Z. Chen, O. M. Gul, and B. Kantarci, "Practical Byzantine fault tolerance based robustness for mobile crowdsensing," *Distrib. Ledger Technol. Res. Pract.*, vol. 2, no. 2, pp. 1–24, 2023.