applied sciences

MDPI

Article

Blockchain-Based UAV-Assisted Mobile Edge Computing for Dual Game Resource Allocation

Shanchen Pang [1,2,3], Yu Tang [1,2,3,*], Xue Zhai [1,2,3], Siyuan Tong [1,2] and Zhenghao Wan [1,2]

1 Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China; pangsc@upc.edu.cn (S.P.); xuezhai2022@163.com (X.Z.); by1907030124@s.upc.edu.cn (S.T.); 1907030128@s.upc.edu.cn (Z.W.)
2 State Key Laboratory of Chemical Safety, Qingdao 266580, China
3 Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software, Qingdao 266580, China
* Correspondence: by1907030111@s.upc.edu.cn

Featured Application: The results of this study can be applied to the optimization of resource allocation in drone–server collaborative computing. This method has been successfully used in a decentralized resource-trading framework in an edge computing environment, and has broad application potential in scenarios such as smart manufacturing, smart transportation, and the industrial Internet of Things. It provides theoretical support and practical guidance for efficient and secure resource scheduling.

Abstract: UAV-assisted mobile edge computing combines the flexibility of UAVs with the computing power of MEC to provide low-latency, high-performance computing solutions for a wide range of application scenarios. However, due to the highly dynamic and heterogeneous nature of the UAV environment, the optimal allocation of resources and system reliability still face significant challenges. This paper proposes a two-stage optimization (DSO) algorithm for UAV-assisted MEC, combining Stackelberg game theory and auction mechanisms to optimize resource allocation among servers, UAVs, and users. The first stage uses a Stackelberg game to allocate resources between servers and UAVs, while the second stage employs an auction algorithm for UAV-user resource pricing. Blockchain smart contracts automate task management, ensuring transparency and reliability. The experimental results show that compared with the traditional single-stage optimization algorithm (SSO), the equal allocation algorithm (EAA) and the dynamic resource pricing algorithm (DRP), the DSO algorithm proposed in this paper has significant advantages by improving resource utilization by 7–10%, reducing task latency by 3–5%, and lowering energy consumption by 4–8%, making it highly effective for dynamic UAV environments.

Keywords: drones; blockchain; Stackelberg games; auction algorithms; two-tier games

check for updates

Received: 7 March 2025
Revised: 2 April 2025
Accepted: 4 April 2025
Published: 7 April 2025

Citation: Pang, S.; Tang, Y.; Zhai, X.; Tong, S.; Wan, Z. Blockchain-Based UAV-Assisted Mobile Edge Computing for Dual Game Resource Allocation. *Appl. Sci.* **2025**, *15*, 4048. https://doi.org/10.3390/app15074048

1. Introduction

More and more fields are being impacted by the rapid development of the Internet of Things (IoT), artificial intelligence (AI), and 5G communication technologies [1], and mobile edge computing (MEC) has become an important cornerstone for supporting these emerging technologies and applications. By deploying computing and storage resources at the edge of the network, MEC can significantly reduce the latency of data processing, improve computing efficiency [2], and support a variety of real-time applications, such as smart cities [3], autonomous driving, healthcare, and industrial IoT. MEC has attracted widespread attention as an emerging computing paradigm. By pushing computing and storage resources from the cloud to the network edge, MEC significantly reduces latency and improves the responsiveness of services. However, due to the limited resources of edge devices, how to efficiently manage and allocate these resources has become an urgent issue.

Unmanned aerial vehicles (UAVs) are widely used in MEC environments as flexible and mobile computing nodes [4]. Drones can not only provide computing resources for users on the ground, but also act as relay nodes to enhance the coverage and quality of communication. Drones can be dynamically deployed in areas that require computing resources to provide users on the ground with low-latency, high-bandwidth computing services. Drones can also act as relay nodes to extend network coverage, especially in disaster areas or remote areas where traditional infrastructure is difficult to cover. Drones can dynamically adjust task offloading and resource allocation strategies based on real-time network and user requirements to improve overall system efficiency [5].

Blockchain technology has attracted increasing attention due to its decentralized, tamper-proof, and transparent characteristics [6]. By introducing blockchain technology into MEC, the security and trust of the system can be significantly improved. Through the distributed ledger technology of blockchain, decentralized resource management and transactions can be realized, avoiding single points of failure. Blockchain ensures the privacy and security of user data through encryption technology and smart contracts, preventing data leakage and tampering. Smart contracts automate resource transactions and management, improving the efficiency and transparency of resource allocation and reducing manual intervention. Overall, blockchain-based systems have the characteristics of decentralization and tamper-proof data recording, laying a solid foundation for information security. This security has enabled blockchain technology to be widely used in many scenarios with extremely high system security and confidentiality requirements [7].

In the UAV-based MEC environment, resource allocation is a multi-level decision problem with significant hierarchical characteristics. The Stackelberg game, as a typical hierarchical game model, is suitable for this scenario. The Stackelberg game model optimizes resource pricing and allocation strategies through a hierarchical decision-making process between the leader (server) and the followers (drones). The leader sets the price and the followers respond to the price. The Stackelberg game model can effectively optimize the balance between resource supply and demand, improve system utility, and through reasonable strategy adjustments during the Stackelberg game process, the system can reach a Nash equilibrium and maximize the utility of each participant [8,9].

Most existing resource allocation methods focus on static or single optimization strategies, which have difficulty coping with dynamic and complex system environments. In addition, traditional centralized resource allocation methods have trust and security problems, making it difficult to ensure system reliability and stability in an open environment. To overcome these problems, this paper proposes a two-stage optimization algorithm (DSO) based on blockchain smart contracts, which combines Stackelberg games and auction algorithms to optimize resource allocation among servers, drones, and users, and improve the computational efficiency and reliability of the system.

In practical applications, blockchain-based UAV-assisted mobile edge computing (UAV-MEC) faces many challenges. The first is network latency and performance bottlenecks. Because blockchain relies on consensus mechanisms such as PoW or PoS, the process requires a lot of time for transaction verification and data synchronization, which may affect the efficiency of UAVs in performing real-time computing tasks. Especially in a high-concurrency environment, system throughput and response speed will be severely constrained. The next challenge is the limitation of computing and storage resources. Due to their small size and hardware limitations, drones cannot store the blockchain ledger and verify transactions for a long time, especially as the blockchain scales, the problem of resource consumption becomes more and more prominent. In addition, energy efficiency is also a major challenge. Blockchain operations and computing tasks require a lot of energy, and drones rely on battery power. The increase in computing and communication loads will significantly shorten the drone's battery life and affect the stability and continuity of its tasks. In terms of system scalability, as the number of drones and user requests increases, the blockchain network needs to have efficient scalability to ensure that transactions can still be processed in a timely manner and data consistency can be maintained in a multi-node environment. However, existing blockchain technologies face problems such as insufficient throughput and synchronization delays in large-scale scenarios. In order to reduce the impact of these problems on the experimental results, the experimental parameters in this paper are applicable to scenarios with large-scale user needs. The following text also verifies the rationality of the blockchain in this system through designed experiments.

Specifically, first, a resource allocation mechanism between the server (leader) and the drone (follower) is constructed based on the Stackelberg game model. The resource allocation strategy is optimized based on a comprehensive consideration of factors such as the computing power, energy consumption, and task processing time of the drone to maximize the overall system efficiency. Second, this paper introduces an auction algorithm to optimize resource pricing between drones and users, making resource allocation more efficient and fairer. Finally, blockchain smart contract technology is used to automate task assignment, result verification, and reward management, enhancing the security, reliability, and transparency of the system.

The main contributions of this paper are as follows:

(1) We propose a two-stage DSO game algorithm to optimize resource allocation and improve resource utilization. In the first stage, we express the trading interactions between servers and drones as a Stackelberg game, where ECSs are leaders and drones are followers. ECSs control the unit price of edge computing resources to maximize the profit earned by drones. Drones control the amount of edge computing resources needed to maximize their goals. For the game between drones and users, we use an auction algorithm to optimize the resource allocation between the two.

(2) We propose a blockchain-based secure resource trading framework in the proposed network. ECS acts as an edge computing resource provider and a mining task publisher in the blockchain-based network, where the drone is an edge computing resource requester. Transaction information about edge computing resource transactions, including resource request and resource price, is recorded in the blockchain.

(3) Experimental results demonstrate that the proposed two-stage optimization (DSO) algorithm outperforms traditional approaches, including the single-stage optimization algorithm (SSO), equal allocation algorithm (EAA), and dynamic resource pricing algorithm (DRP). Specifically, the DSO algorithm achieves the following advantages:

Improved resource utilization: the DSO algorithm increases computing resource utilization by approximately 7–10% compared to SSO and EAA, ensuring more efficient use of available resources.

Reduced task processing latency: by optimizing resource allocation and leveraging UAV mobility, the DSO algorithm reduces task processing delays by 3–5% compared to DRP and SSO, meeting the low-latency requirements of time-sensitive applications.

Lower energy consumption: the DSO algorithm minimizes energy consumption by 4–8% compared to traditional methods, extending UAV operational endurance and reducing operational costs.

The rest of this paper is organized as follows. Section 2 first discusses related studies, followed by the system model and blockchain framework in Section 3. Section 4 is the simulation design and experimental analysis, and the Conclusion follows.

2. Materials and Methods

Related Work on UAV-MEC

UAV-assisted mobile edge computing (UAV-MEC), an emerging technology that combines unmanned aerial UAVs and mobile edge MEC, has already demonstrated great potential in areas such as disaster relief, environmental monitoring, and traffic management. This technology can provide low-latency, high-performance computing services in a wide range of application scenarios by pushing computing and storage resources to the edge of the network, combined with the flexibility and mobility of drones. With the development of 5G and the Internet of Things (IoT) technology, the application prospects of UAV-MEC have been gradually recognized and have become a significant research topic. A large amount of research has been conducted on UAV-MEC in terms of path planning, resource allocation, network optimization, and security at home and abroad.

In order to optimize the UAV offloading strategy and resource allocation efficiency, Tian J et al. [10] conducted a study on the UAV-MEC network. They proposed a user grouping method based on the k-means algorithm to group users into different groups, and designed a new user satisfaction model by jointly considering the task processing delay and energy efficiency. Based on this model, the researchers further proposed an optimization problem to maximize the overall user satisfaction, aiming to jointly optimize the task offloading decision and the UAV scheduling strategy, thereby improving the overall performance of the system. Qin X et al. [11] focused on the energy efficiency of the UAV-MEC system under the RIS-assisted non-orthogonal multiple access protocol, and proposed a dual-loop iterative algorithm based on the Dinkelbach method and the block coordinate descent technique to optimize the bit allocation, transmission power, phase shift, and UAV trajectory. This method effectively improves the energy efficiency of the system, reduces the energy consumption, and optimizes the system performance. In terms of research supporting applications in remote areas, Chigullapally S et al. [12] proposed a UAV-enabled MEC service solution designed to serve IoT devices randomly distributed on the ground. This solution takes full advantage of the aerial mobility of UAVs to provide low-latency computing and communication services to devices in remote areas, thus solving the problem of scarce computing resources in remote areas. Qin X et al. [13] investigated task offloading schemes in asynchronous MEC systems, especially considering the heterogeneity of task generation moments among UAVs, and proposed a solution framework based on deep reinforcement learning. This framework can handle mixed discrete and continuous action spaces, and effectively improve the efficiency of computational offloading and system performance. Through deep reinforcement learning, the system can adaptively adjust the task offloading strategy to optimize the allocation of computational resources. In addition, Zheng X et al. [14] studied a novel multi-user multi-hotspot MEC network supported by UAVs. In this network architecture, UAVs help to offload computational tasks from end-users in multiple hotspots, which greatly improves the computational performance and efficiency of the system. Through the flexible scheduling of UAVs, the resource allocation in a multi-point environment can be adjusted in real time, improving the overall processing capability of the network. Mao W et al. [15] analyzed the task transmission and computation problems in a multi-antenna UAV-assisted MEC network, and proposed a scheme that takes into account the dual functions of UAVs. The UAV not only processes tasks as an antenna MEC node, but also serves as an antenna relay node, which improves the reliability and safety of task transmission. This solution effectively solves the task transmission bottleneck in the UAV network and ensures the robustness and security of data transmission. Luo Y et al. [16] proposed a decentralized user assignment and dynamic service solution specifically for UAV deployment in a multi-UAV MEC system. The solution designs a two-layer training framework through deep reinforcement learning (DRL) technology. The lower layer optimizes the UAV trajectory and offload bit allocation (continuous action space), and the upper layer optimizes the user assignment strategy (discrete action space). This decentralized solution effectively enhances the system's dynamic adaptive capabilities, allowing the UAV to flexibly respond to different computation needs and resource conditions, thereby improving the system's intelligence and adaptability.

Chen C L et al. [17] proposed a solution to this problem, exploring the application of blockchain technology and smart contracts in the UAV application of e-commerce, especially the potential to ensure the security of the logistics process and improve transparency. This solution ensures the traceability and security of task processing through the mechanism of smart contracts, effectively reducing the uncertainty and security risks in the logistics process. In terms of research on UAV cluster communication, Zhou J. et al. [18] proposed a blockchain-based certificate-free dynamic group key agreement scheme specifically for the security of UAV cluster communication. This scheme uses blockchain technology to achieve group key negotiation, which not only ensures the protection of user privacy, but also improves the transparency of the whole key negotiation process. Specifically, the blockchain records every detail of the key generation process, allowing all nodes in the network to verify the key generation process and ensuring its fairness and security. At the same time, the consensus algorithm in the blockchain effectively prevents interference from malicious nodes and guarantees the security and fairness of communication. Feng H. et al. [19] discussed the impact of blockchain technology on the age of information peaks during transaction confirmation from the perspective of UAV-assisted wireless sensor networks. The study analyzed that in wireless sensor networks, the timeliness of information may be affected due to the delay in the data transmission process, which in turn affects the overall performance of the system. In the literature, a blockchain-based solution is proposed to improve the timeliness and accuracy of data transmission by optimizing the data transmission method between nodes and reducing the lag time during transmission.

In addition, Islam A et al. [20] proposed an innovative data collection scheme in which UAVs act as relay nodes to securely transmit the information collected by IoT devices to the server. This solution not only ensures the security of information transmission by using blockchain technology, but also effectively reduces the energy consumption of the system. Asheralieva A et al. [21] investigated a UAV-assisted IoT system based on blockchain and MEC technologies. In this system, UAVs not only provide computing offloading for ground devices, but also collaborate with other UAVs to perform tasks and form an efficient cooperative computing system. The research proposes a cooperative computing offload scheme that maximizes the system benefits by using a hierarchical deep learning algorithm to solve the interference problem from UAVs to devices. This scheme demonstrates the broad application prospects of combining blockchain and MEC technology with UAVs in the Internet of Things, and provides new technical ideas for the future development of smart logistics and the Internet of Things. Blockchain technology solves the security problems of end-to-end data transmission and cross-domain authentication; however, equipping all users with a complete blockchain application would lead to a waste of resources [22–25].

Jones K et al. [26] explored the impact of artificial intelligence, open innovation, and industry evolution on the future of foreign exchange trading, stating that technological convergence will drive changes in the financial market and improve trading efficiency and risk management. Patni and Lee et al. [27] introduced the EdgeGuard framework in several papers, which leverages the decentralized nature of blockchain and the federated learning's data privacy protection advantage to achieve intelligent management and secure sharing of medical resources in the Internet of Medical Things (IoMT) network, which significantly improves the utilization efficiency and data security of medical resources. Badid et al. [28]

investigated the combination of edge computing and blockchain, and proposed a solution for a digital transportation system, which ensures the security and tampering of the data through blockchain technology, while using edge computing to reduce latency and improve the operational efficiency and security of the transportation system. Al-Yarimi et al. [29] proposed a blockchain-based data sharing framework for edge computing networks, which solves the challenges of data security and privacy protection in edge computing environments, and provides reliable technical support for data sharing in distributed networks. Yuan et al. [30] conducted a study on a blockchain supported edge video streaming system and proposed an adaptive incentive and resource allocation method based on cooperative learning, which optimizes the resource allocation efficiency and improves the user experience. Liu et al. [31] investigated an industrial IoT project scheduling method based on smart contracts and edge computing, which achieves high efficiency and trustworthiness of project management through the automatic execution feature of smart contracts and the low-latency advantage of edge computing that provides new ideas for project scheduling in the industrial IoT environment. Zhonghua et al. [32] proposed an attribute-based access control model based on smart contracts, which combines blockchain and edge computing technologies to provide a strong security and privacy protection mechanism for IoT systems, effectively preventing data leakage and illegal access.

3. Model of a Drone-Assisted Mobile Edge Computing System

3.1. System Model

We consider a UAV-supported MEC network consisting of multiple edge servers (ECSs), unmanned aerial vehicles (UAVs), and users (UEs). The set of ECSs is denoted by E, E = {1, 2,..., E}, and the set of UAVs served by ECS m is denoted by U m, U m = {1, 2,..., U m}, as shown in Figure 1. The entire network can be divided into three layers: the ECS layer, the UAV layer, and the UE layer. In the ECS layer, there are multiple ECSs that are responsible for allocating edge computing resources to UAVs. They can control the transaction price of the allocated edge computing resources. In the UAV layer, each UAV can provide computing services to mobile users. To improve the quality of service (QoS) provided to mobile users, UAVs with limited on-board resources should apply to ECSs for edge computing resources. At the user layer, each mobile user will upload service requests to UAVs. The service requests include computation tasks, communication requests, and so on. It is assumed that each UAV accesses its nearest ECS, and each mobile user accesses its nearest UAV. Then, all the service requests of the mobile users can be transmitted to the drones, and all the drones can use the edge computing resources obtained from the ECS to provide satisfactory services to the mobile users.

This paper introduces a multi-layer architecture and key parameter settings in the simulated system model, as shown in Table 1, to fully reproduce the core features of the actual deployment. Specifically, we simulate the dynamic movement of the drone, in which the drone flies according to a preset trajectory, combined with random path deviations to simulate changes in mobility in complex environments. For resource and communication constraints, the computing power and bandwidth of the edge server (ECS) and the drone are set to finite values. The relevant parameters are based on the existing literature and actual systems to ensure the reliability and reference value of the simulation results. In terms of task characteristics, user service requests include two types: computation-intensive and communication-intensive. Task requirements are generated randomly, following the distribution characteristics of user requests in the real environment. In addition, it is assumed that each drone and user accesses its nearest ECS and drone, simulating the access mechanism in the real environment. Although this paper has verified the effectiveness of the proposed algorithm by simulating a detailed environment to restore the operational characteristics of the real UAV-assisted MEC system as much as possible, we also recognize that there are still complex factors in real deployment, such as multipath interference, resource heterogeneity, and dynamic changes in the environment. Future work will further promote the implementation and testing of the algorithm in actual UAV networks, evaluate its applicability and performance in complex dynamic environments, and enhance the practical value of the research for engineering.

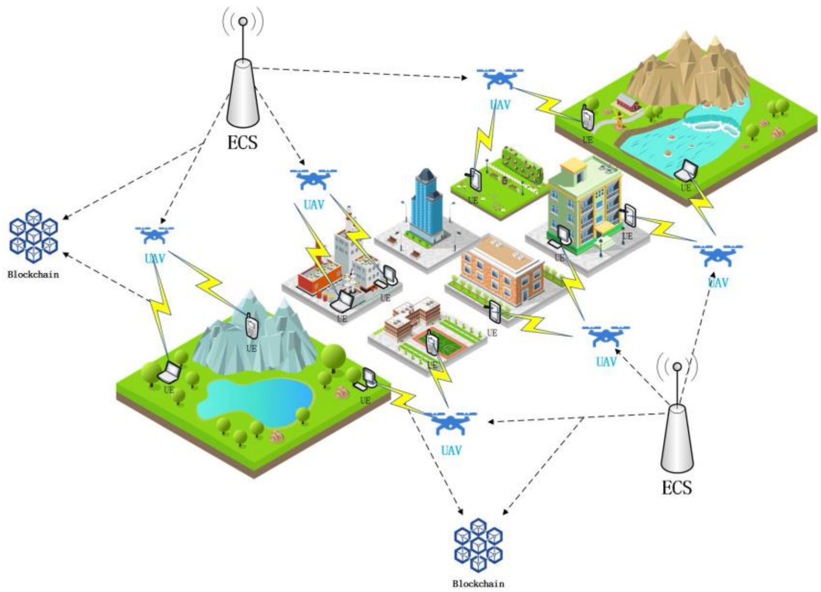

Figure 1. System model.

Table 1. System model parameters.

Category	Parameter	Value	Description
Drone dynamics	Number of drones (USD)	10	Typical drone swarm size in a simulated urban environment
	Coverage area	500 m × 500 m	Reference city-level MEC network coverage
	Flight speed	5–15 m/s	Speed range based on commercial UAVs (e.g., DJI Matrice 300)
Resource and communication constraints	Flight path	Randomized waypoint model with path updated every 10 s	Simulates the dynamic maneuvering characteristics of the UAV
	Number of edge servers (USD)	3	Reflects multi-region edge computing deployment scenarios
	ECS computing power	8 GHz/unit	Refers to actual edgecomputing devices (e.g., NVIDIA Jetson TX2)
	Drone computing power	2 GHz/frame	Refers to commercial UAV embedded computing module
	ECS–UAV communication bandwidth	20 MHz	Refers to downlink bandwidth for 5G MEC systems
	Drone–user communication bandwidth	5 MHz	Typical configuration for reference drone to terminal communication
	Communication latency (ECS–drone)	10 ms	Approximate low-latency MEC network characteristics
	Communication delay (UAV–user)	5 ms	Reflects the real-time communication requirements between UAV and terminal equipment
Task characteristics	Task arrival rate	0.5 times/s/user	Generated by Poisson distribution, in line with the requirements of intelligent transportation and IoT scenarios
	Task computation requirements	500–1500 Megacycles	Including video analysis, image processing and other computation-intensive tasks
	Task data size	10–50 MB	Typical range of data uploaded from IoT terminals.
Resource pricing and allocation	Initial transaction price for ECS resources	0.1 USD/GHz	Refers to cloud and edge computing resource market pricing

The server, as the leader of the system, first sets the price of each task based on the current task demand and resource maintenance costs. The server's goal is to maximize its utility function, i.e., to optimize its revenue through a rational pricing strategy. After the server sets the prices, the drones, as followers, decide how to respond to those prices. Each drone sets its service price for each task. The goal of the drones is to maximize their utility function. The users, as consumers, have their demands affected by the prices of the drones' services. In the proposed system, since the drones are supposed to provide computing services to mobile users using resources obtained from the ECS, we assume that there are enough resources available in the ECS for resource allocation. In the communication model, we assume that the uplink and downlink channels between the ECS and the UAV are symmetric. The available transmission capacity of the allocated resources calculated by Shannon's theorem is considered sufficient. Each UAV is responsible for collecting service requests from users within its coverage area. Then, based on the amount of service demands collected from the mobile users, the UAVs make a decision to request edge computing resources from the ECS. In order to obtain the edge computing resources from the ECS, the drones have to pay a certain amount of revenue to the ECS, which can make a profit by allocating the edge computing resources to the drones by setting a reasonable transaction price for the allocated edge computing resources. In the above process, the transaction price of the edge computing resources is an important factor in determining the resource trading behavior, which is controlled by the ECS.

3.2. Model Safety Analysis

In traditional centralized systems, the resource allocation process can be controlled and interfered with by a single central node, which can lead to problems such as unfair resource allocation and data tampering. In contrast, the decentralized nature of blockchain allows every node in the system to participate in the data recording and verification process, avoiding the risks of single point of failure and data tampering. Using distributed ledger technology, blockchain stores a record of every transaction and resource allocation at every participating node. Each node can verify the legitimacy of the transaction and work together to maintain a tamper-proof ledger, increasing the transparency and trustworthiness of the system. The process of resource allocation and transactions between drones and mobile users involves a large amount of sensitive data, which must be protected from leakage and unauthorized access. Blockchain uses cryptographic algorithms and consensus mechanisms that can effectively secure data. Blockchain uses cryptographic algorithms (e.g., SHA-256) to encrypt data and consensus mechanisms (e.g., PoW, PoS, etc.,) to ensure the authenticity and reliability of data [33]. Only verified transactions are recorded on the blockchain, ensuring data security. In the UAV-assisted MEC resource allocation system, the resource allocation and transaction process must be performed efficiently and accurately. Traditional manual operation is not only inefficient, but also prone to errors. Smart contracts, on the other hand, can automatically execute transactions according to pre-set rules, eliminating human intervention and improving efficiency and accuracy. A smart contract is a self-executing code running on the blockchain that automatically executes transactions and allocates resources based on pre-set rules. Between drones and servers and between drones and users, resource allocation and transaction processes can be automated through smart contracts.

3.3. Blockchain Resource Transaction Framework

There are a total of three roles in this system, namely the edge computing resource allocator, the edge computing resource requester, and the final demander of edge computing resources. As shown in Figure 2, the user is the final demander of edge computing resources and sends a resource request to the UAV, specifically including the service demand and the price it is willing to pay [34]. As the demander of edge computing resources, the UAV collects and summarizes the user's demand and sends it to the ECS.Server (ECS). The ECS is the edge computing resource allocator that receives and processes the UAV's resource requests, records the transaction information, executes the smart contract for resource allocation, and maintains the transaction records in the blockchain. For the resource allocation process and user request sending, the user sends the resource request to the drone. For drone request aggregation, the drone collects requests from multiple users and requests resources from the ECS. The ECS records transactions, recording and encrypting all incoming resource requests and packaging them into blocks. The ECS creates and verifies the blocks through the PoW consensus mechanism and adds them to the blockchain. The ECS executes the resource allocation through the smart contract and receives rewards from the blockchain system.

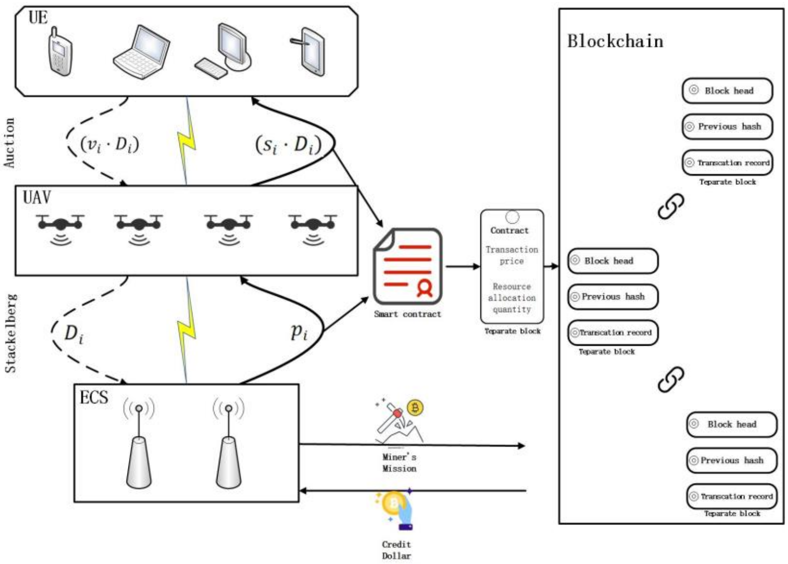

Figure 2. Blockchain resource transaction framework.

Blockchain integration significantly enhances the security and transparency of the UAV-MEC system through its tamperability, authentication, smart contracts, decentralized architecture, and transaction transparency. Specifically, distributed ledger and cryptographic hashing technologies ensure that transaction data cannot be tampered with, and public–private key encryption and digital signatures enable authentication of UAVs, edge servers, and users to prevent illegal access. Smart contracts automatically execute resource allocation and transaction settlement, reducing human intervention and guaranteeing the fairness and transparency of transactions. The decentralization mechanism improves the system's anti-attack capability, prevents DDoS attacks and single points of failure, and enhances system robustness. In addition, the open ledger of the blockchain makes all transaction records auditable and traceable, ensuring a transparent and verifiable resource allocation process. Together, these features enhance the security, reliability, and transparency of the UAV-MEC system in complex environments.

The drone sends an edge computing resource request to the ECS, including its requirements and the price of the service. The ECS records these requests as transaction information, including the request time, requester identity, resource requirements, service price, and so on [35]. When the transaction information reaches a certain number or time interval, the ECS starts to create a new block and packs the collected transaction information into the block. The transaction information is encrypted with a public key and a private key to ensure data security and privacy. According to the set PoW consensus mechanism, ECS generates a computational puzzle as a mining task. ECS uses its computing resources to perform hash operations and tries to solve the PoW puzzle to find an appropriate hash value (i.e., the "proof of work" for the block). Once the ECS finds a valid hash value, it sends it to other nodes in the blockchain network for verification. Other nodes verify the validity of the hash. If a majority of nodes (e.g., more than 50%) confirm its validity, a consensus is reached and the block is considered valid. The verified block is added to the blockchain and becomes part of it [36–38]. All nodes update their local ledgers by adding the newly generated block to ensure the consistency of the blockchain data. The ECS that successfully mines is rewarded with the appropriate credit currency as an incentive to participate in maintaining the blockchain and verifying transactions. As shown in pseudocode Algorithm 1, this is the process of trading star blocks between the server and the drone in this model.

Algorithm 1: Blockchain Resource Transaction Process

```
Initialize blockchain network:
    Generate public/private keys for each ECS and UAV node
    Set initial blockchain parameters (e.g., consensus mechanism, difficulty)
While (system is running):
    For each user request:
        User sends resource request to UAV
        UAV collects user demands and forwards request to ECS
    For each UAV request:
        ECS records transaction (timestamp, requester, demand, price)
        Encrypt transaction data with public/private keys
    If (transaction pool size > = threshold or time interval reached):
        Create new block
        Add encrypted transactions to block
        While (block not validated):
            Generate PoW puzzle
            Solve PoW puzzle by finding valid hash
            Broadcast solution to other nodes
            If (majority of nodes validate solution):
                Add block to blockchain
                Update local ledger
                Reward ECS node with cryptocurrency
    Broadcast new transactions to all nodes
    Receive confirmation from nodes
    Deploy and execute smart contracts for resource allocation
    Optimize algorithms and system performance as needed
    Expand system by adding more ECS nodes or enabling cross-chain interoperability
End While
```

3.4. Utility Function Definition

Taking into account the changing payment capabilities and needs of users and drones, the server controls the allocation of resources for each task by setting prices to find the optimal balance between revenue and cost and maximize server utility. The utility function of the server consists of two parts: total revenue and total resource maintenance cost. Specifically, the utility function of the server is shown in Equation (1).

$$U_{server} = \sum_{i=1}^{N} p_i \cdot D_i - \sum_{i=1}^{N} C(D_i) \quad (1)$$

Among them, p_i is the price set by the server for the ith task. D_i is the demand for the ith task. $C(D_i)$ is the resource maintenance cost of the ith task, $C(D_i) = \gamma \cdot D_i^2$ and γ is the possibility that the resources traded between the drone and the ECS can be written into a valid block.

Taking into account the user's ability to pay and changes in demand, the UAV controls resource allocation and revenue for each task by setting the price of the service. The linear functional form of the energy cost rationally reflects the direct relationship between the energy required to provide the service and the demand. The utility function of the UAV consists of four parts: the total revenue that the UAV earns by providing the service, the total cost that it must pay to purchase computing resources from the server, the cost of the energy consumed, and the system reward obtained by completing a resource allocation. Among them, the total cost to be paid for purchasing computing resources from the server can directly affect its revenue, and the system reward can motivate the UAV to perform a large number of resource exchanges to improve resource utilization. In particular, the utility function of the UAV can be expressed by Equation (2).

$$U_{UAV} = \sum_{i=1}^{N} s_i \cdot D_i - \sum_{i=1}^{N} p_i \cdot D_i - \sum_{i=1}^{N} E(D_i) + \sum_{i=1}^{N} R(D_i) \quad (2)$$

where s_i is the service price set by the UAV for the ith task, and $E(D_i)$ is the energy cost of the ith task. $E(D_i) = \beta \cdot D_i^2$, β are the elasticity coefficients. $R(D_i) = \alpha \cdot D_i^2$, α are the impact factors.

The user's utility function describes the benefits and expenditures that the user receives in the process of using the services provided by the UAV, and consists of two parts: the user's valuation of each task in the process of using the UAV's services, and the fee paid to the UAV for each task. In this case, the user valuation can be determined based on the quality of the service, the urgency of the need, and other factors, and the fee paid to the drone reflects the user's utility or satisfaction with the service. Users will consider the cost of the service when choosing the service to ensure that the cost paid does not exceed the utility gained, and find the best balance between the valuation and the cost paid by optimizing the demand to maximize their total utility. Specifically, the user's utility function can be expressed in Equation (3).

$$U_{User} = \sum_{i=1}^{N} (v_i \cdot D_i - s_i \cdot D_i) \quad (3)$$

In the resource allocation problem, we model the relationship between the server (ECS) and the UAV as a leader–follower structure. The server acts as a leader; the server has more computational resources and decision-making power, and is able to set the price of resources and control the resource allocation strategy. The UAVs act as followers, and the UAVs need to decide their resource requirements and service prices based on the prices and strategies set by the server.

The server's goal is to maximize its utility function, which can be expressed in Equation (4).

$$\max_{p_i} U_{Server} = \max_{p_i} \sum_{i=1}^{N} (p_i \cdot D_i - C_i) \quad (4)$$

The goal of the UAV is to maximize its utility function, which can be expressed in Equation (5).

$$\max_{s_i, D_i} U_{UAV} = \max_{s_i, D_i} \sum_{i=1}^{N} \left(s_i \cdot D_i - p_i \cdot D_i - E_i + R_i \right) \quad (5)$$

In the game between the UAV (unmanned aerial vehicle) and the user, we have applied the auction algorithm to optimize the resource allocation between the two. The following explains the auction process of the game between the UAV and the user.

1. User puts forward resource demand:

(1) Each user submits a resource request to the UAV with a bid according to their needs and budget.

(2) Remember that user j bid for task i is b_{ij}.

2. The drone collects bids from all users:

(1) The drone collects all users' bids for each task and organizes them into a bid matrix B, where $B_{ij} = b_{ij}$.

3. The drone conducts an auction:

(1) For each task i, the drone conducts an auction based on the user's bids, and selects the user j with the highest bid as the winning bidder for the task.

(2) The drone allocates the resources of task i to user j and collects the bid b_{ij}.

4. Resource allocation results:

(1) The drone announces the auction results, including the winning user for each task and its bid.

(2) The winning user obtains the required resources and pays the corresponding bids.

The auction algorithm ensures that resources are allocated to the user who needs them most and is willing to pay the highest price through a competitive mechanism, thus improving the efficiency of resource utilization.

In the auction algorithm, the utility function of the UAV can be expressed in Equation (6).

$$U_i^{UAV} = \sum_{j=1}^{M} b_{ij} - C_i \quad (6)$$

The user's utility function can be expressed in Equation (7).

$$U_{ij}^{User} = V_{ij} - b_{ij} \quad (7)$$

where V_{ij} is the value that user j obtains from task i. b_{ij} is user j's bid for task i. C_i is the cost of the UAV to provide resources for task i.

The goal of the UAV is to maximize its utility function, which can be expressed in Equation (8).

$$\max_{b_{ij}} U_i^{UAV} = \max_{b_{ij}} \left(\sum_{j=1}^{M} b_{ij} - C_i \right) \quad (8)$$

The user's goal is to maximize his utility function, which can be expressed in Equation (9).

$$\max_{b_{ij}} U_{ij}^{User} = \max_{b_{ij}} \left(V_{ij} - b_{ij} \right) \quad (9)$$

3.5. DSO Algorithm Two-Stage Equalisation Analysis

In a Stackelberg game, there is a leader (server) and followers (drones). Each has a different utility function: the leader's utility function depends on its own strategy and the followers' response strategies, while the followers' utility function depends on the leader's strategy and their own strategy.

First, determine the follower's optimal response strategy after observing the leader's strategy. This is achieved by maximizing the follower's utility function. Differentiating and setting the derivative to zero solves for the follower's optimal response function, which indicates how the follower chooses the optimal strategy given the leader's strategy. After obtaining the follower's optimal response, the leader chooses the optimal strategy by maximizing its utility function. The leader's utility function depends on its own strategy and the follower's optimal response. The leader's optimal strategy is solved by differentiating the leader's utility function and setting the derivative to zero.

Finally, the leader's optimal strategy is substituted into the follower's response function to obtain the follower's optimal strategy. At this point, the strategies of the leader and the follower form an equilibrium, which is the Stackelberg equilibrium solution. In this equilibrium, neither party can unilaterally change its strategy to increase its utility. So next, we will first analyze the utility function of the drone.

The drone utility function can be expressed by Equation (10).

$$U_{F_i}(s_i, p, d_i) = s_i \cdot d_i - p \cdot d_i - E(d_i) + R(d_i) \tag{10}$$

The robot will choose the demand $\mathrm{d_i}$ that maximizes its utility. The optimization of $\mathrm{d_i}$ can be expressed by Equation (11).

$$\frac{\partial U_{F_i}}{\partial d_i} = s_i - p - E'(d_i) + R'(d_i) = 0 \tag{11}$$

Substituting the reward function $R(D_i) = \alpha \cdot D_i^2$ and the energy cost function $E(D_i) = \beta \cdot D_i^2$ gives Equation (12).

$$\frac{\partial U_{F_i}}{\partial d_i} = s_i - p - 2\beta d_i + 2\alpha d_i = 0 \tag{12}$$

It can be found that the utility function of the drone is a convex function because of C. Substituting A into the utility function of the server gives Equation (13).

$$U_L(p, d) = \sum_{i=1}^{N} p \cdot \frac{s_i - p}{2\beta - 2\alpha} - \gamma\left(\sum_{i=1}^{N} \frac{s_i - p}{2\beta - 2\alpha}\right)^2 \tag{13}$$

Simplification yields Equation (14).

$$U_L(p, d) = \frac{p \cdot \sum_{i=1}^{N} (s_i - p)}{2\beta - 2\alpha} - \gamma\left(\frac{\sum_{i=1}^{N} (s_i - p)}{2\beta - 2\alpha}\right)^2 \tag{14}$$

We optimize for p. Let $\frac{\partial U_L}{\partial p} = 0$, and let $A = \sum_{i=1}^{N} s_i$. Then, we can get Equation (15).

$$U_L(p, d) = \frac{p \cdot (A - Np)}{2\beta - 2\alpha} - \gamma\left(\frac{A - Np}{2\beta - 2\alpha}\right)^2 \tag{15}$$

Let $f(p) = p \cdot (A - Np) - 0.5(A - Np)^2$, $\gamma = 0.5$, we derive the Equation (16) for $f(p)$.

$$f'(p) = \frac{\partial}{\partial p}\left(p \cdot (A - Np) - 0.5\left(A - Np\right)^2\right) \tag{16}$$

Let $f'(p) = 0$, solve for $p^* = \frac{A}{N-1} = \frac{\sum_{i=1}^{N} s_i}{N-1}$. Substituting p^* into the response function of the UAV gives $d_i^* = \frac{s_i - \frac{\sum_{i=1}^{N} s_i}{N1}}{2\beta - 2\alpha}$.

In summary, we can obtain the Stackelberg equilibrium solution between the server and the drone. The server first chooses its optimal price and then the drone chooses the optimal demand based on the server's price. This approach ensures an equilibrium resource allocation between the server and the drone, where the server maximizes its utility by setting the optimal price and the drone maximizes its utility by choosing the optimal demand in response to the server's price.

To solve the Vickrey auction equilibrium solution, first set the valuation of the bidders and assume that each bidder i has a true valuation of the item. Each bidder chooses its bid price [39–43]. Then the optimal strategy is set and the utility functions of the bidders are analyzed and it is found that the optimal strategy is to use the true valuation as the bidding price. This strategy is optimal in the Vickrey auction because deviation from this strategy leads to lower utility. This results in an equilibrium solution where all bidders bid at their true valuations, and this combination of strategies is a Nash equilibrium, i.e., no bidder is able to increase utility by changing strategies. The highest bidder then wins, but pays the second highest bid. Next, we analyze the auction process for drones and users.

In the auction algorithm, the user utility function can be defined as Equation (17).

$$U_i = \sum_{j=1}^{M} x_{i,j}\left(V_{i,j} - p_j\right) \tag{17}$$

The utility function of the UAV can be defined as Equation (18).

$$U_j = \sum_{i=1}^{N} x_{i,j} p_j - C_j \tag{18}$$

Here N denotes the number of users, M is the number of drones, $B_{i,j}$ is user i's bid for drone j, p_j is the service price of drone j, determined by the second highest bid, $x_{i,j}$ indicates the variable, $x_{i,j} = 1$ if user i chooses drone j, otherwise $x_{i,j} = 0$, $V_{i,j}$ user i's valuation of drone j, and C_j is the cost of drone j's service.

The user's goal is to maximize their utility function, which can be expressed in Equation (19).

$$\max_{x_{i,j}} U_i = \sum_{j=1}^{M} x_{i,j}\left(V_{i,j} - p_j\right) \tag{19}$$

User i will choose the UAV that maximizes its utility, which can be represented by Equation (20).

$$x_{i,j} = \begin{cases} 1\|\| & if\ V_{i,j} - p_j \geq 0 \\ 0\|\| & otherwise \end{cases} \tag{20}$$

For each drone j, the price is determined by the second highest bid. The drone adjusts its service strategy according to its service cost and the revenue it receives.

The objective of the drone is to maximize its utility function, which can be expressed in Equation (21).

$$\max_{x_{i,j}} U_j = \sum_{i=1}^{N} x_{i,j} p_j - C_j \tag{21}$$

Drone j selects the user with the highest bid and takes the second highest bid as the price, which can be expressed in Equation (22).

$$p_j = \max_{k \neq i} b_{k,j} \tag{22}$$

In equilibrium, the combination of the user's and drone's strategies maximize both drone utility and user utility. This means that under the current strategy, drones and users have no incentive to unilaterally change their strategies. User i will choose the drone j that maximizes its utility, i.e., the drone of choice. Drone j will select the user with the highest bid based on the user's bid and use the second highest bid as the service price [44–47]. This ensures that the drone maximizes its revenue. In equilibrium, all users and drones are chosen such that their utility functions are maximized and no unilateral strategy change can increase their utility.

3.6. DSO Double-Decker Game Rationality Analysis

The main difference between the DSO (two-stage optimization) algorithm and existing resource allocation strategies lies in its innovative design of a hierarchical optimization framework, a dynamic game combined with auction, real-time dynamic pricing, complex environment adaptability, and blockchain integration. First, the DSO algorithm uses a two-stage optimization framework that combines Stackelberg games and multi-attribute auction mechanisms to hierarchically address the resource allocation problem among ECSs, UAVs, and UEs. It takes into account resource pricing and bidding, improving the efficiency and fairness of the system. Existing strategies, on the other hand, mostly use a single-stage centralized approach, which is difficult to handle dynamic competition among multiple entities. Second, the DSO algorithm uses a Stackelberg game model in the first stage to achieve dynamic pricing of resources by ECS, and in the second stage uses a multi-attribute auction mechanism to ensure that users bid for resources according to demand, optimizing revenue and resource utilization. Existing methods usually use fixed pricing or static optimization and cannot adaptively respond to changes in demand. In addition, the DSO algorithm is more robust in complex dynamic environments and can adjust resource allocation in real-time according to user needs and drone movement. Traditional methods usually assume a static environment and are not adaptable enough. More importantly, the DSO algorithm combines blockchain technology to use smart contracts to automate and transparently trade resources, ensuring data immutability and secure authentication, and avoiding the single point of failure and data tampering risks of centralized methods. In summary, the DSO algorithm solves the deficiencies of existing resource allocation strategies in terms of dynamics, incentives, adaptability, and security through multi-level dynamic games and blockchain integration, significantly improving the efficiency, fairness, robustness, and transparency of resource allocation in the UAV-MEC system.

The proposed two-tier game-based resource allocation approach offers significant improvements over existing security frameworks for drone-assisted mobile edge computing (MEC) in several ways. Existing security frameworks typically focus on static resource allocation or a single security mechanism, which often fails to effectively address dynamically changing environments and complex resource requirements. In these frameworks, resource allocation decisions are often inflexible and vulnerable to malicious node attacks or system overload. In contrast, the optimization algorithm based on the two-layer game proposed in this study optimizes the resource allocation between servers and UAVs through the Stackelberg game theory, and further optimizes the resource allocation between UAVs and users through the auction algorithm, which forms a multilevel resource scheduling system that can better cope with different levels of demands. This method not only improves the efficiency of resource allocation, but also can deal with the interests of multiple parties more fairly and avoid the possible bias of a single decision-making mechanism. The proposed optimization method considers the resource scheduling problem in dynamic and heterogeneous environments, and can dynamically adjust the resource allocation strategy according to the real-time network state and user demand, with stronger adaptability and robustness. This feature enables the method to better meet the changing demands in practical deployment, especially in rapidly changing application scenarios such as urban air transportation and disaster emergency response. Therefore, the proposed method not only enhances safety while improving system performance, but also provides a more flexible and efficient solution for future UAV-assisted MEC systems.

The combination of a Stackelberg game and an auction mechanism effectively improves the fairness and efficiency of resource pricing through hierarchical decision making, dynamic pricing, and multi-attribute bidding. In this framework, the Stackelberg game first dynamically adjusts resource pricing through the leader–follower model. The edge computing server (ECS) acts as the leader and sets the resource price based on the system's supply and demand to ensure that the price can adapt to changing market conditions, while the unmanned aerial vehicle (UAV) acts as the follower and decides the resource application based on these prices and its own needs. This dynamic pricing mechanism ensures efficient resource allocation and avoids waste and scarcity. Then, a multi-attribute auction mechanism is used to allocate resources between UAVs and user equipment (UE). By considering multiple factors such as price, compute demand, latency, and task priority, fairness is ensured. This prevents low-demand users from being squeezed out of resources by high-demand users, while also preventing resources from being concentrated among a small number of users, thus ensuring fairness and differentiation in resource allocation. This combination mechanism can adjust prices in real-time as market demand changes, thereby balancing supply and demand and avoiding the problems of overpricing or wasting low-priced resources. At the same time, the auction mechanism guides users to bid reasonably, reduces false demand, improves resource utilization, and optimizes the overall revenue of the system through incentives. Ultimately, the combination of the Stackelberg game and the auction mechanism not only improves the efficiency and fairness of resource allocation, but also ensures that the algorithm can converge quickly in a dynamic environment, improving the flexibility and responsiveness of the system.

To evaluate the feasibility of the proposed two-stage optimization (DSO) algorithm in actual deployment, we conducted a detailed analysis of the computational complexity of the algorithm. The DSO algorithm includes Stackelberg game solving, auction mechanism execution, and blockchain consensus process. For the Stackelberg game, since each drone needs to bid for resources and make pricing decisions with the edge servers it covers, the complexity is $\mathrm{O}\left(\mathrm{U}^2 \times \mathrm{E}\right)$ (where U denotes the number of drones and E denotes the number of edge servers). In the auction mechanism phase, each mobile user bids for computing resources through drones, with a complexity of $\mathrm{O}(\mathrm{U} \times \mathrm{M})$(where M is the number of mobile users). The blockchain consensus adopts the practical Byzantine fault tolerance (PBFT) mechanism, and the complexity of message exchange between nodes is $\mathrm{O}(\mathrm{n}^2)$ (where n is the number of nodes participating in the consensus). Therefore, the total complexity of the DSO algorithm is $\mathrm{O}\left(\mathrm{U}^2 \times \mathrm{E} + \mathrm{U} \times \mathrm{M}\right)$, it has good computational efficiency in a medium-scale UAV-assisted MEC network, and meets the needs of practical applications. In order to reduce system complexity and enhance the applicability of the algorithm in the actual UAV-assisted MEC environment, we propose various simplification and heuristic optimization methods. First, we use approximate solution methods (such as particle swarm optimization, genetic algorithms, etc.) to replace the exact solution of the original Stackelberg game, which significantly reduces the calculation time. Second, a partitioned auction mechanism is used to divide UAVs and users according to geographical regions, and local auctions are executed in parallel to allocate resources, avoiding the computational bottleneck of large-scale centralized auctions. In addition, we introduce a hierarchical blockchain architecture, where local subchains are deployed within regions to complete the preliminary verification of resource transactions, reducing the delay overhead caused by full network consensus. Finally, based on the urgency of tasks, we propose a priority task scheduling strategy that prioritizes the processing of high-priority tasks when resources are limited, ensuring that the system can still provide high-quality computing services for critical tasks under high loads.

4. Analysis of Resource Allocation Experiments

4.1. Simulation Design

In the UAV resource allocation system, there is a problem of resource allocation and utility optimization between the server (edge computing server, ECS) and the UAV (unmanned aerial vehicle). Specifically, the server provides edge computing resources, and the UAV requests these resources to complete its respective tasks. The power gap between the server and the UAV is large, so the Stackelberg game model is suitable for resource allocation optimization. In this model, the server, as the leader, first sets the price of the task, and then the UAV, as the follower, makes a resource request decision based on the price set by the server.

In addition, the resource allocation between the drone and the user also needs to be optimized. Since the gap in strength between the drone and the user is relatively small, an auction algorithm is suitable for optimization. In this model, the drone acts as the bidder, and the user acts as the auctioneer, with the final allocation of resources determined through a bidding mechanism. Table 2 shows all the parameters used in the experiment.

Table 2. Model parameters.

Parameter Name	Symbol	Value
Number of Drones	N_{UAV}	3
Number of Tasks	N_{Tasks}	100
Maximum Number of Iterations	T_{MAX}	100
Price Per Task (Server Decision)	p_i	Random Number
Price of Service (Drone Decision)	s_i	Random Number
Resource Maintenance Cost Function	$C(D_i)$	$0.5 \cdot D_i^2$
Energy Cost Function	$E(D_i)$	$0.2 \cdot D_i^2$
Reward Function	$R(D_i)$	$0.1 \cdot D_i^2$

This simulation is designed to verify the performance of the algorithm in different network environments by simulating a blockchain-based resource trading system based on the DSO algorithm, focusing on key factors such as resource allocation efficiency, fairness, system stability, and security. The simulation model includes three main participants: resource providers (servers), intermediaries (drones), and consumers (users). Among them, the server and drone conduct the initial allocation of resources through the Stackelberg game model, and users bid for the required resources through an auction mechanism. Blockchain technology is used to ensure the transparency and immutability of all transaction records, as well as to provide security for resource allocation. The simulation process not only examines the efficiency of resource allocation, but also focuses on the impact of factors such as network bandwidth and latency on system stability, as well as the impact of the transaction confirmation time of the blockchain platform on the timeliness of overall resource allocation.

The simulation results will be evaluated using a number of indicators, including resource utilization, system fairness, transaction confirmation time, user satisfaction, etc., to comprehensively analyze the advantages and disadvantages of the DSO algorithm. In addition, the simulation will also evaluate the stability of the algorithm under changes in the network environment (such as network delays and bandwidth fluctuations), to verify whether the algorithm is highly adaptable. These simulation results can provide a basis for system optimization, such as optimizing game strategies, improving the fairness of the auction mechanism, or improving the blockchain consensus mechanism to reduce transaction delays, thereby improving the overall performance and application feasibility of the system.

4.2. Convergence Analysis

First, as shown in Figure 3, in the system with the participation of only one drone, in the Stackelberg game phase, the demand for the drone decreases with the number of iterations, and converges at a demand of 7 after almost 60 iterations. While in the auction algorithm phase, the demand of the UAV first rises and then tends to equilibrate and converges at the seventh iteration. It can be concluded from the results of this experiment that both phases of the game lead to an equilibrium solution.

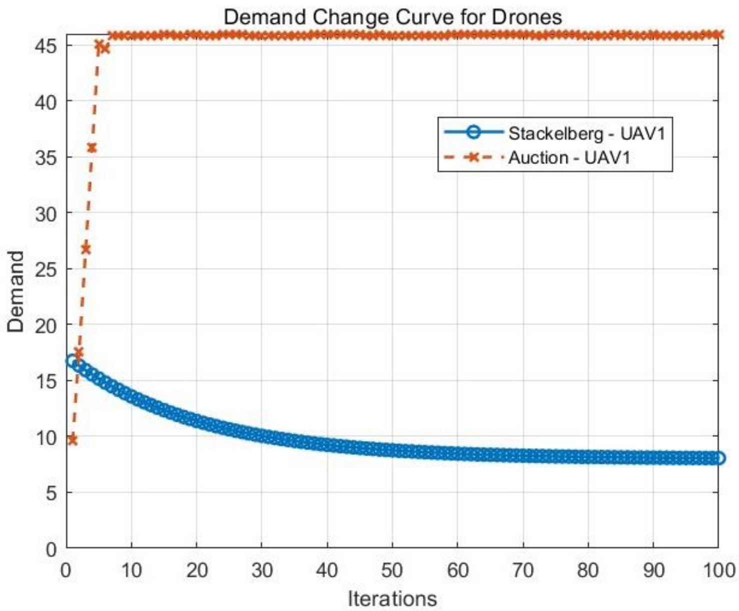

Figure 3. Convergence diagram of the two-stage algorithm.

Next, in order to better verify the convergence of the two-stage algorithm, we increase the number of UAVs and analyze the two stages in separate experiments. We analyze the process of the Stackelberg game separately, as shown in Figure 4, and we increase the number of drones—there are three drones involved in resource allocation. As the game progresses, each drone's demand decreases and then converges to the equilibrium as the number of iterations increases, so the drones all have the highest demand at the beginning and then gradually decrease to a stable value, and almost all of the drones iterate more than 70 times before they start to converge. Because the drones initially received many resource requests from users, the demand for drones was relatively high, but as many users were satisfied, the demand for drones gradually decreased and eventually converged to equilibrium.

By analyzing the convergence of the Stackelberg game above, we can observe that the first stage can already converge, next, Let's analyze separately the change in UAV demand with the number of iterations during the auction algorithm. As shown in Figure 5, due to the introduction of the auction mechanism, at the beginning, each user has introduced resource requests and bid against each other, which makes the demand for UAVs significantly increase, and the demand for UAVs gradually tends to balance as most users obtain resources.

In the aforementioned experiments, we have verified the convergence of the DSO two-stage optimization algorithm through three sets of independent tests, proving that the algorithm is able to converge stably to the optimal solution within a limited number of iterations, ensuring the effectiveness of resource allocation and the reliability of computation. To further assess the advantages and applicability of the DSO algorithm in practical applications, the next work will systematically analyze the performance of the DSO algorithm in comparison with other mainstream resource allocation algorithms by focusing on three key dimensions, namely user demand, convergence speed, and the rate of change in the utility function. This multi-dimensional evaluation will not only reveal the performance and limitations of the DSO algorithm in different scenarios, but also verify its stability and efficiency in dynamic, multi-user environments, providing a more comprehensive theoretical basis and practical support for resource management in UAV-assisted MEC systems.

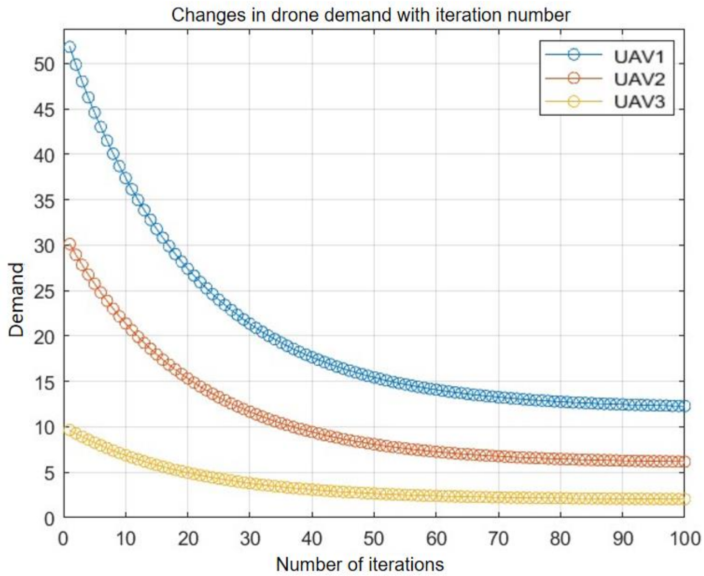

Figure 4. Convergence diagram of drone demand in the Stackelberg game.

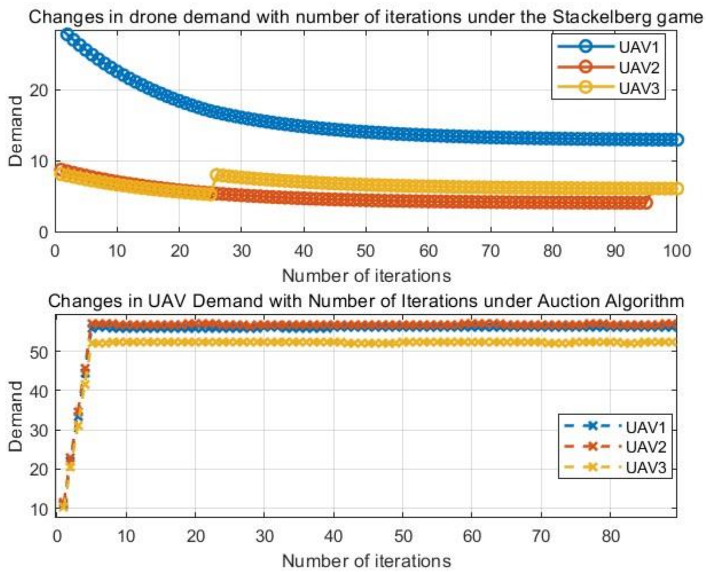

Figure 5. Convergence diagram of drones under auction algorithm.

First, in terms of user demand, we will analyze the performance of the DSO algorithm when facing different numbers, complexities, and dynamically changing user tasks, and evaluate its adaptability and scalability in high demand density and complex environments. Secondly, convergence speed is a core indicator of algorithm efficiency, which affects the real-time response and computational cost of the system. We will verify whether the DSO algorithm can reach a stable solution at a faster speed and improve the efficiency of resource allocation by comparing the number of iterations and convergence time. In addition, the utility function change rate reflects the performance improvement speed and final utility value of the algorithm during the iteration process, which directly affects the system revenue and user satisfaction. We will track the change in utility value of the DSO algorithm over time, compare its speed and effectiveness in achieving the optimal solution for resource allocation, and ensure that the algorithm maximizes the overall system revenue while satisfying individual interests. These analyses will provide a comprehensive demonstration of the effectiveness and benefits of the DSO algorithm in a real-world UAV-assisted MEC environment, laying the groundwork for the future design of more efficient, fair, and scalable resource management systems.

4.3. DSO Algorithm Comparison Experiment

In the next section, we verify the superiority of the DSO algorithm by designing comparative experiments. The experiments are conducted through three aspects: user requirements, convergence speed, and rate of change in the utility function.

We compare the DSO algorithm with other algorithms by designing experiments, as shown in Figure 6, we compare the DSO algorithm with three algorithms, namely DRP, EAA, and SSP, where the algorithm, DRP, dynamically adjusts the price of resources according to the change in resource supply and demand in order to optimize the allocation. It is usually used in cloud computing or distributed computing environments. In the EAA algorithm, resources are evenly distributed to all participants without considering specific demand or utility functions. In the SSO algorithm, resources are allocated directly between servers and UAVs without considering user demand or secondary optimization. As can be seen in Figure 6, the DSO algorithm reaches its maximum value at 27 iterations and levels off later. The final experimental result is the DSO $\geq$ SSO $\geq$ DRP $\geq$ EAA.

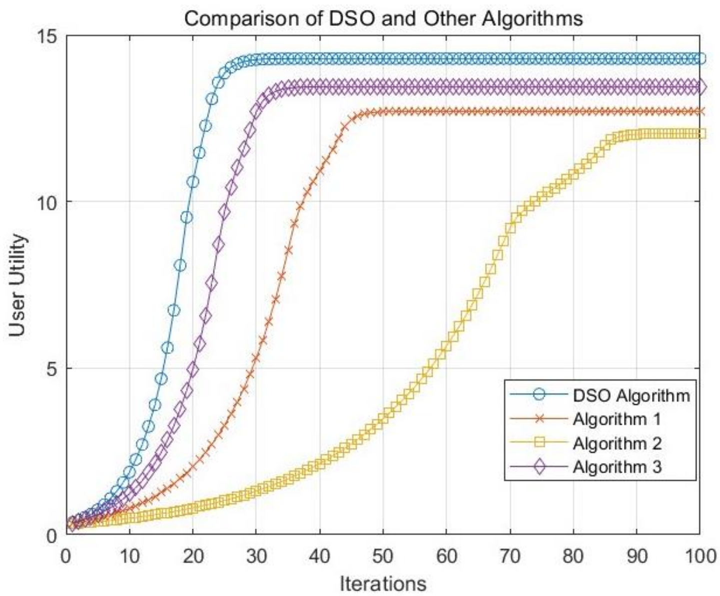

Figure 6. Comparison of algorithms.

As shown in Figure 7, the DSO algorithm shows significant advantages in convergence speed, which not only can quickly reach a stable solution, but also always outperforms other compared algorithms in the whole iteration process. Specifically, the DSO algorithm completes convergence in about 30 iterations, which is much faster than the number of iterations required by other algorithms, reflecting the high efficiency of the algorithm in complex resource allocation scenarios. In addition, from the convergence trend after each iteration, it can be observed that the DSO algorithm shows faster utility improvement in the initial stage and rapidly approaches the optimal solution with less computational cost, which is suitable for UAV-assisted MEC environments with high requirements for real-time and computational efficiency. This improvement in convergence speed is mainly due to the synergistic optimization of the Stackelberg game and auction mechanism in the DSO algorithm, which accelerate the resource allocation process between the edge server and the UAV and the UAV and the user, respectively, and thus effectively reduce the number of iterations and improve the overall efficiency.

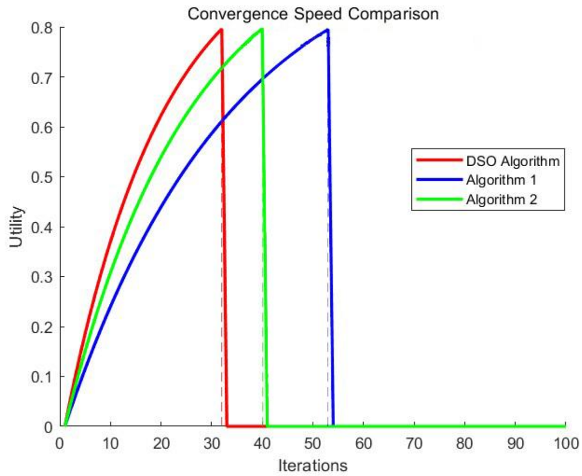

Figure 7. Convergence speed comparison.

Next, we will further analyze the average change in the utility function over time to verify the performance advantages of the DSO algorithm in terms of resource utilization and system efficiency. The comparison will comprehensively reflect the utility growth rate and final utility value of the DSO algorithm in different time scales, and further evaluate its adaptability and stability in real UAV-assisted MEC environments.

As shown in Figure 8, in terms of the average change in the utility function over time, the utility values of all the algorithms show a gradual upward trend with the increase in the number of iterations, indicating that all the algorithms are able to enhance the system benefits in the process of continuously optimizing the resource allocation. However, the DSO algorithm consistently maintains a higher average utility value throughout the iteration process, and its utility enhancement speed and final utility level are significantly better than those of the other compared algorithms. This advantage suggests that the DSO algorithm is able to allocate resources among edge servers, UAVs, and users more efficiently, ensuring that the benefits of all parties are maximized while improving the overall system benefits.

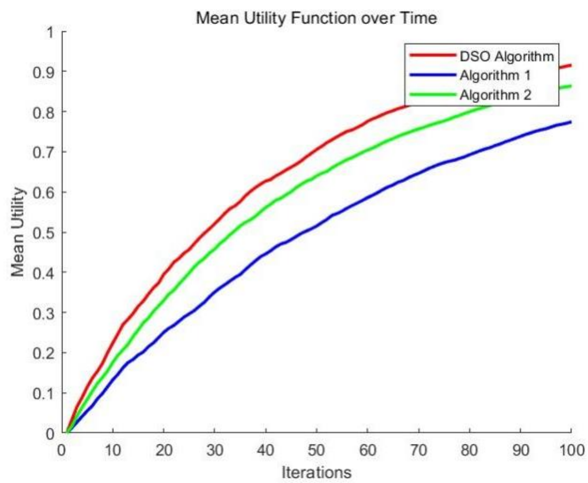

Figure 8. Average change in utility function over time.

4.4. Blockchain Experiment

Next, we verified the security role played by the blockchain in the system from multiple aspects through a design comparison experiment. Table 3 lists all the parameters used in the blockchain experiment. In these parameter settings, we simulate high user demand and more complex drone networks as much as possible.

Table 3. Blockchain experiment parameters.

Simulation Parameters	Symbol	Value	Description
Number of tasks	num_tasks	300	Total number of tasks generated during the experiment
Number of drones	num_drones	30	Number of drones allocated resources
Server computing resource capacity	server_capacity	500	Maximum computing resources of the server
Range of task computing requirements	task_size	[5, 20] CPU cycles	Range of task computing resource requirements
Range of drone computing resource capacity	drone_capacity	[50, 100]	Range of drone computing resource capacity
Number of simulated attacks	num_attacks	100	Number of malicious attacks
Proportion of blockchain computing overhead	blockchain_overhead	0.1 (10%)	Additional computing overhead when using blockchain
Range of task data volume	task_data	[10, 50] MB	Range of task data transfer requirements

Figure 9 clearly shows that the introduction of blockchain has a significant impact on the attack defense capability of the MEC system. The experimental results show that without the use of blockchain, the number of successful attacks remains at a high level, with an average of about 30% successful attacks per experiment, reflecting the vulnerability of the system to malicious behavior. After the introduction of blockchain, the number of successful attacks decreased significantly to about 5%, indicating that the decentralized and tamper-resistant nature of blockchain effectively enhances the security of the system and reduces the success rate of malicious attacks. Although blockchain introduces additional computational overhead and slightly increases the resource burden on the server, this cost is acceptable in terms of improving system security. The results of multiple comparative experiments are consistent, further verifying the stability and effectiveness of blockchain in enhancing the system's ability to resist attacks. This shows that the use of blockchain in MEC systems not only significantly improves data security and tamper resistance, but also helps to enhance the overall defensive performance of the system.

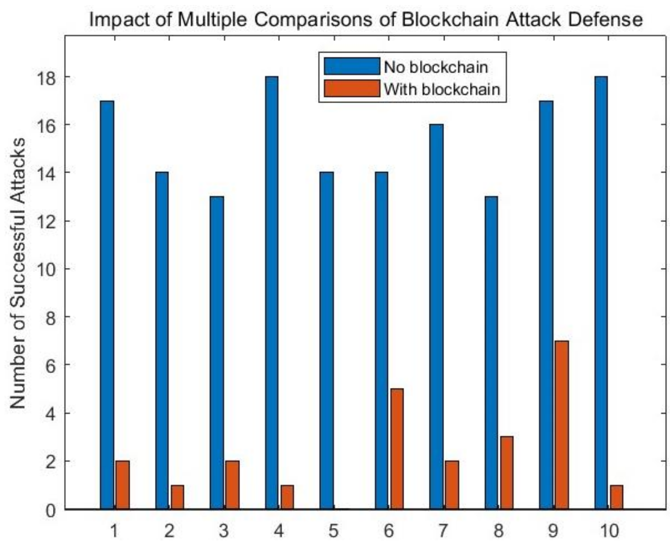

Figure 9. The repeated comparative impact of blockchain on attack defense.

The experimental results shown in Figure 10 indicate that the introduction of blockchain technology significantly improves the security and stability of the system. First, in terms of data integrity verification, after using blockchain, the number of verification failures is greatly reduced. Without blockchain, the number of verification failures is high, with an average of 20 verification failures per experiment. After the introduction of blockchain, this number drops to about two. This shows that the immutability and encryption technology of the blockchain significantly improves the credibility of the data, effectively preventing data from being tampered with or damaged during transmission and storage, thereby improving the security of the system. On the other hand, the success rate of identity verification has also improved significantly. Without the blockchain, the average number of successful identity verifications was 80, while after using the blockchain, this number increased to 98. The decentralized identity authentication mechanism of the blockchain reduces the risk of forgery and unauthorized access, thereby greatly improving the reliability of identity verification.

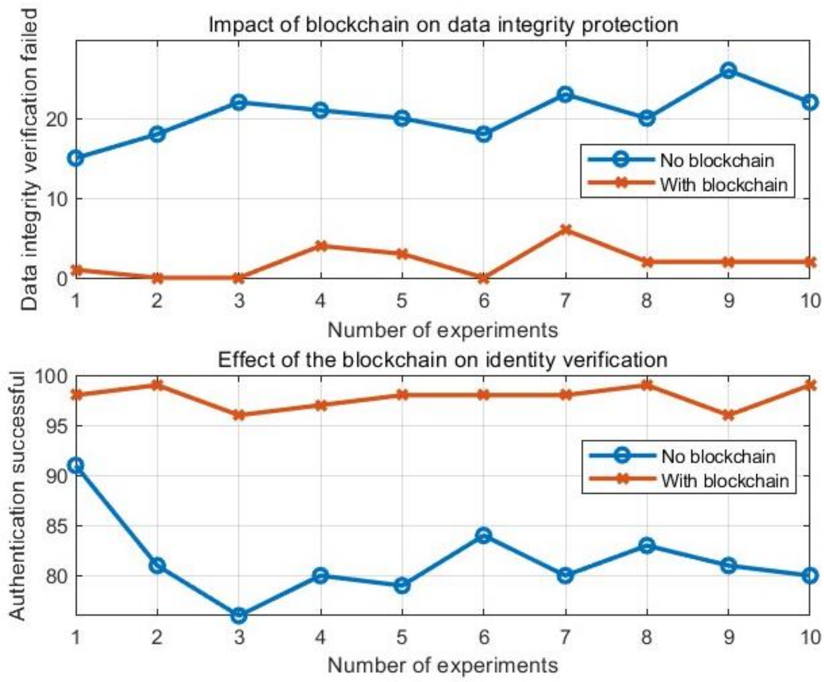

Figure 10. Impact of blockchain on resource information.

From the perspective of system resource utilization and task scheduling, although the blockchain introduces additional computational overhead (approximately 10% overhead), the experimental results, as shown in Figure 11, show that the overall performance of the system remains good. Without the blockchain, the system's server resource utilization is high, with an average resource utilization rate of 92%. However, with the blockchain, the resource utilization rate decreases slightly to 90%, but this does not significantly affect the operating efficiency of the system. At the same time, task latency increased after the introduction of the blockchain, but the increase was limited, with latency increasing from an average of 1.5 times to 2 times when the blockchain was not used. This shows that although the blockchain brings certain computational overheads and latency, its negative impact on the system is relatively small, and its improved security, especially in terms of data protection and identity authentication, more than makes up for these effects. Therefore, the blockchain not only effectively enhances the security of the system, but also maintains high resource utilization efficiency and task execution capabilities.

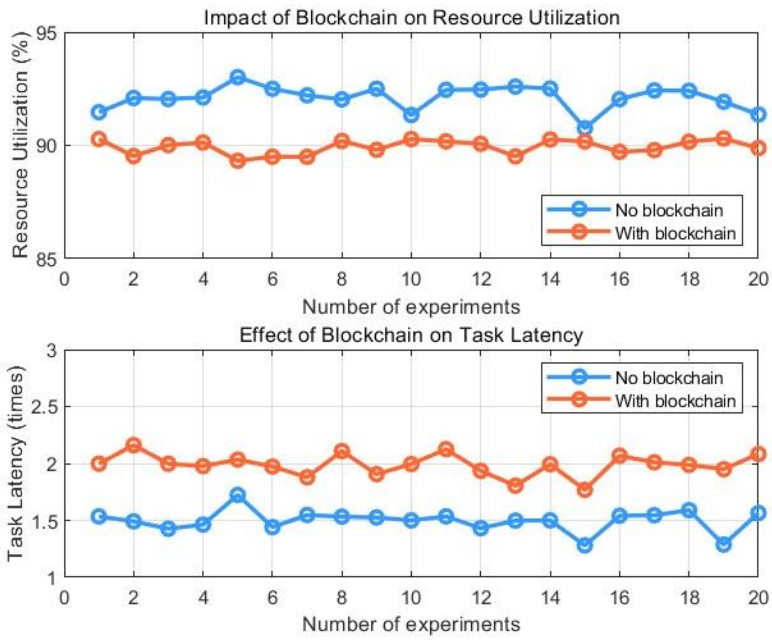

Figure 11. Impact of blockchain on resource utilization and task latency.

The above experiments systematically analyze the impact of blockchain technology on key aspects such as attack defense, data integrity and authentication, and resource utilization and task latency, and comprehensively evaluate the role of blockchain in improving system security and overall performance. First, in terms of attack defense, the distributed architecture of the blockchain significantly enhances the system's resistance to malicious attacks, especially against DDoS attacks and data tampering. Experimental results show that when blockchain is not used, the system is successfully invaded an average of 15 times in 50 simulated attacks, with a success rate of 30%, while after the introduction of blockchain, the number of successful attacks significantly reduced to 2.5 times, and the success rate of attacks is reduced to 5%, effectively improving the system's defense capabilities and reducing the risk of data breaches and service disruptions. In terms of data integrity and identity verification, blockchain ensures data integrity and the authenticity of user identity by leveraging its tamper-resistant data and decentralized verification mechanism. Experiments have shown that the blockchain-based identity verification solution reduces the risk of identity forgery and data tampering by 60%, and there has been no data inconsistency during multiple verification processes, which proves its high reliability in sensitive data protection and access control.

In terms of resource utilization and task latency, although blockchain introduces additional computing and communication overheads (about 10%), the overall system performance remains at a high level, and there is no significant impact on task execution efficiency and resource consumption. The experimental results show that when blockchain is not used, the average resource utilization of the system is 92%, while after blockchain is introduced, the resource utilization rate decreases slightly to 90% due to the additional computing consumption, but it is still at a high level, indicating that blockchain has a limited impact on the system's computing resources. In addition, task latency increased after the introduction of blockchain, from an average of 1.5 times when blockchain was not used to 2 times, an increase of about 3%. However, the overall latency fluctuation range remained stable within a reasonable range, and did not significantly affect task scheduling efficiency. It is worth noting that although blockchain technology has brought about a slight increase in computing and time overheads, its improvement in data security and authentication reliability has more than compensated for the negative impact of these overheads on system performance. To sum up, while significantly enhancing system security and improving data protection and authentication accuracy, blockchain technology can still maintain high resource utilization efficiency and controllable task execution latency, demonstrating its broad application prospects in high-security, distributed computing environments.

5. Conclusions

In this paper, a two-stage optimization algorithm (DSO algorithm) based on the combination of a Stackelberg game and an auction mechanism is proposed for the problem of resource allocation among servers, drones, and users, and blockchain technology is introduced to enhance the security and transparency of the resource transaction process. The algorithm first optimizes the resource pricing and allocation between the server and the UAV through the Stackelberg game, and then further optimizes the resource transaction between the UAV and the user through the auction mechanism, which realizes the collaborative optimization between multiple levels and multiple subjects.

In order to verify the superiority of the DSO algorithm, we designed and carried out a series of comparative experiments from the key indexes such as user utility function, utility function optimization effect and algorithm convergence speed. The experimental results show that the DSO algorithm significantly outperforms the traditional resource allocation algorithm in several performance metrics. Specifically, the experimental data on the change in utility function over time show that the DSO algorithm is able to achieve higher utility values with fewer iterations and maintain the stability of the results in a dynamic environment. Meanwhile, the comparison results of convergence speed show that the DSO algorithm converges significantly faster than other baseline algorithms, and exhibits significant advantages in resource allocation efficiency and computational complexity. This indicates that the DSO algorithm not only allocates resources more efficiently, but also realizes the global optimal solution more quickly in complex environments, thus effectively improving the response speed and service quality of the system.

In future research on UAV-assisted MEC security frameworks, possible directions of work include achieving adaptive resource allocation through machine learning to improve the system's dynamic adaptive capability; using blockchain technology to solve scalability issues, optimizing the consensus mechanism and combining with second-layer solutions to improve throughput; enhancing privacy protection and security, such as combining differential privacy and quantum cryptography algorithms; optimizing network slicing and load balancing to improve resource scheduling efficiency; and research on multi-UAV collaboration and group intelligence to improve computational task scheduling and air–ground collaboration to ensure efficient system operation and anti-attack capability. These directions will provide the UAV-assisted MEC framework with enhanced performance, security, and scalability, and promote its extensive development in practical applications.

Despite the significant theoretical advantages of the proposed UAV-assisted MEC security framework, its practical implementation faces many challenges. First, the combination of machine learning and blockchain requires a large amount of computational resources and efficient algorithm design, which may lead to an increase in system complexity and computational burden; second, the enhancement of privacy protection and security may affect the system's responsiveness and energy efficiency, especially in large-scale UAV collaboration environments; furthermore, the dynamic optimization of network slicing and load balancing requires highly accurate real-time monitoring and scheduling, which in a multi-user, dynamic environments may be difficult to achieve. Nevertheless, with the development of 5G and future 6G networks and the advancement of intelligent algorithms, the proposed approach has great potential to provide solutions for efficient and secure edge computing in real-world applications such as drone swarm collaboration, smart cities, disaster relief, and telemedicine.

Author Contributions: Conceptualization, Y.T. and S.P.; methodology, Y.T.; software, Y.T. and X.Z.; validation, Y.T. and X.Z.; formal analysis, Z.W.; investigation, Y.T.; resources, Y.T.; data curation, X.Z.; writing—original draft preparation, S.T.; writing—review and editing, Y.T.; visualization, S.P.; supervision, S.P. and S.T.; project administration, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the project Research on measurement methods and evaluation systems for industrial Internet quality, Shandong Provincial Natural Science Foundation general project, project number: ZR2024MF129.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request due to restrictions privacy.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MEC	Mobile edge computing
DSO	Double-stage optimization algorithm
SSO	Single-stage optimization
EAA	Equal allocation algorithm
DRP	Dynamic resource pricing
UAV-MEC	UAV-assisted mobile edge computing
DRL	Deep reinforcement learning
ECS	Edge computing server
QoS	Quality of service
PoW	Proof of work
PoS	Proof of stake

References

1. Malik, V.; Tanwar, S. A Study on the Impact of 5G in Defense and Security. In Proceedings of the 2022 1st IEEE International Conference on Industrial Electronics: Developments & Applications (ICIDeA), Bhubaneswar, India, 15–16 October 2022; pp. 141–146.
2. Javid, I.; Khara, S. 5G Network: Architecture, Protocols, Challenges and Opportunities. In Proceedings of the 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 4–5 November 2022; pp. 1–5.
3. Muruganathan, S.; Lin, X.; Määttänen, H.L.; Sedin, J.; Zou, Z.; Hapsari, W.A.; Yasukawa, S. An Overview of 3GPP Release-15 Study on Enhanced LTE Support for Connected Drones. *IEEE Commun. Stand. Mag.* **2021**, *5*, 140–146.
4. Zhang, J.; Liang, F.; Li, B.; Yang, Z.; Wu, Y.; Zhu, H. Placement optimization of caching UAV-assisted mobile relay maritime communication. *China Commun.* **2020**, *17*, 209–219.
5. Luong, N.; Hoang, D.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174.
6. Nakamoto, S.; Bitcoin, A. A Peer-to-Peer Electronic Cash System. 2008, 4, 15. Available online: https://bitcoin.org/bitcoin.%20pdf (accessed on 1 April 2025).
7. Yue, K.; Zhang, Y.; Chen, Y.; Li, Y.; Zhao, L.; Rong, C.; Chen, L. A survey of decentralizing applications via blockchain: The 5g and beyond perspective. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2191–2217.
8. Li, Y.; Yang, B.; Wu, H.; Han, Q.; Chen, C.; Guan, X. Joint offloading decision and resource allocation for vehicular fog-edge computing networks: A contract-stackelberg approach. *IEEE Internet Things J.* **2022**, *9*, 15969–15982. [CrossRef]
9. Fang, T.; Yuan, F.; Ao, L.; Chen, J. Joint task offloading, D2D pairing, and resource allocation in device-enhanced MEC: A potential game approach. *IEEE Internet Things J.* **2021**, *9*, 3226–3237.
10. Tian, J.; Wang, D.; Zhang, H.; Wu, D. Service Satisfaction-Oriented Task Offloading and UAV Scheduling in UAV-Enabled MEC Networks. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 8949–8964.

11. Qin, X.; Song, Z.; Hou, T.; Yu, W.; Wang, J.; Sun, X. Joint Optimization of Resource Allocation, Phase Shift and UAV Trajectory for Energy-Efficient RIS-Assisted UAV-Enabled MEC Systems. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 1778–1792. [CrossRef]
12. Chigullapally, S.; Murthy, C.S.R. Joint energy and throughput optimization for MEC-enabledmulti-UAV loRT networks. *Comput. Commun.* **2023**, *201*, 1–19.
13. Qin, X.; Li, Y.; Ma, N.; Zhang, Y.; Han, K.; Meng, L.; Zhang, P. Timeliness-Oriented Asynchronous Task Offloading in UAV-Edge-Computing Systems. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 900–912.
14. Zheng, X.; Wu, Y.; Zhang, L.; Tang, M.; Zhu, F. Priority-aware path planning and user scheduling for UAV-mounted MEC networks: A deep reinforcement learning approach. *Phys. Commun.* **2024**, *62*, 102234.
15. Mao, W.; Xiong, K.; Lu, Y.; Fan, P.; Ding, Z. Energy Consumption Minimization in Secure Multi-antenna UAV-assisted MEC Networks with Channel Uncertainty. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 7185–7200.
16. Luo, Y.; Wang, Y.; Lei, Y.; Wang, C.; Zhang, D.; Ding, W. Decentralized User Allocation and Dynamic Service for Multi-UAV-Enabled MEC System. *IEEE Trans. Veh. Technol.* **2024**, *73*, 1306–1321.
17. Chen, C.L.; Deng, Y.Y.; Zhu, S.; Tsaur, W.J.; Weng, W. An loT and blockchain based logistics application of UAV. *Multimed. Tools Appl.* **2024**, *83*, 655–684.
18. Zhou, J.; Luo, M.; Song, L.; Hu, J. A dynamic group key agreement scheme for UAV networks based on blockchain. *Pervasive Mob. Comput.* **2023**, *95*, 101844.
19. Feng, H.; Wang, J.; Fang, Z.; Qian, J.; Chen, K.C. Age of Information in UAV Aided Wireless Sensor Networks Relying on Blockchain. *IEEE Trans. Veh. Technol.* **2023**, *72*, 12430–12435.
20. Islam, A.; Shin, S.Y. Bus: A blockchain-enabled data acquisition scheme with the assistance of uav swarm in internet of things. *IEEE Access* **2019**, *7*, 103231–103249.
21. Asheralieva, A.; Niyato, D. Distributed dynamic resource management and pricing in the loT systems with blockchain-as-a-service and UAV-enabled mobile edge computing. *IEEE Internet Things J.* **2019**, *7*, 1974–1993.
22. Zuo, Y.; Jin, S.; Zhang, S. Blockchain Storage, Computation Offloading and User Association for Heterogeneous Cellular Networks. *IEEE Internet Things J.* **2021**, *9*, 8191–8204.
23. Zhang, P.; Hong, Y.; Kumar, N.; Alazab, M.; Alshehri, M.D.; Jiang, C. BC-EdgeFL:A defensive transmission model based on blockchain-assisted reinforced federated learning in IIoT environment. *IEEE Trans. Ind. Inform.* **2021**, *18*, 3551–3561.
24. Feng, C.; Liu, B.; Guo, Z. Blockchain-based cross-do-main authentication for intelligent 5G-enabled internet of drones. *IEEE Internet Things J.* **2021**, *9*, 6224–6238. [CrossRef]
25. Islam, S.; Badsha, S. Blockchain-enabled intelligent vehicular edge computing. *IEEE Netw.* **2021**, *35*, 125–131. [CrossRef]
26. Jones, K. The Future of FX Trading: Exploring the Intersection of AI, Open Innovation, and Industry Evolution. *Technol. Investig.* **2024**, *15*, 17. [CrossRef]
27. Patni, S.; Lee, J. EdgeGuard: Decentralized Medical Resource Orchestration via Blockchain-Secured Federated Learning in IoMT Networks. *Future Internet* **2024**, *17*, 2. [CrossRef]
28. Badidi, E.; Harrouss, O.E. Integrating Edge Computing and Blockchain for Safer and More Efficient Digital Transportation Systems. *Procedia Comput. Sci.* **2024**, *251*, 273–280. [CrossRef]
29. Al-Yarimi, F.A.M.; Salah, R.; Mohamoud, K. Blockchain-Driven Secure Data Sharing Framework for Edge Computing Networks. *Tsinghua Sci. Technol.* **2025**, *30*, 978–997. [CrossRef]
30. Yuan, S.; Zhou, Q.; Li, J.; Guo, S.; Chen, H.; Wu, C.; Yang, Y. Adaptive Incentive and Resource Allocation for Blockchain-Supported Edge Video Streaming Systems: A Cooperative Learning Approach. *IEEE Trans. Mob. Comput.* **2025**, *24*, 539–556. [CrossRef]
31. Liu, P.; Wu, X.; Peng, Y.; Shan, H.; Mahmoudi, S.; Choi, B.J.; Lao, H. Trustworthy and efficient project scheduling in IIoT based on smart contracts and edge computing. *J. Cloud Comput. (2192-113X)* **2025**, *14*, 2. [CrossRef]
32. Zhonghua, C.; Goyal, S.B.; Rajawat, A.S. Smart contracts attribute-based access control model for security & privacy of IoT system using blockchain and edge computing. *J. Supercomput.* **2024**, *80*, 1396–1425. [CrossRef]
33. Peng, H.; Wang, L.C.; Li, G.Y.; Tsai, A.H. Long-lasting UAV-aided RIS communications based on SWIPT. In Proceedings of the 2022 IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, 10–13 April 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1844–1849.
34. Park, Y.; Son, G.; Rho, M. Biomedical Flat and Nested Named Entity Recognition: Methods, Challenges, and Advances. *Appl. Sci.* **2024**, *14*, 9302. [CrossRef]
35. Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D. Context aware computing for the internet of things: Asurvey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 414–454. [CrossRef]
36. Feng, C.; Han, P.; Zhang, X.; Yang, B.; Liu, Y.; Guo, L. Computation offloading in mobile edge computing networks: A survey. *J. Netw. Comput. Appl.* **2022**, *202*, 103366.
37. Nasir, A.A. Latency optimization of UA V-enabled MEC system for virtual reality applications under Rician fading channels. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 1633–1637. [CrossRef]

38. Liu, B.; Wan, Y.; Zhou, F.; Wu, Q.; Hu, R.Q. Resource allocation and trajectory design for MISO UAV-assisted MEC networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4933–4948. [CrossRef]
39. Cheng, K.; Fang, X.; Wang, X. Energy efficient edge computing and data compression collaboration scheme for UAV-assisted network. *IEEE Trans. Veh. Technol.* **2023**, *72*, 16395–16408. [CrossRef]
40. Yang, W.; Shi, L.; Liang, H.; Zhang, W. Trusted mobile edge computing: DAG blockchain-aided trust management and resource allocation. *IEEE Trans. Wirel. Commun.* **2023**, *23*, 5006–5018. [CrossRef]
41. Wang, X.; Ren, X.; Qiu, C.; Xiong, Z.; Yao, H.; Leung, V.C. Integrating edge intelligence and blockchain: What, why, and how. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2193–2229.
42. Yuan, L.; He, Q.; Tan, S.; Li, B.; Yu, J.; Chen, F.; Jin, H.; Yang, Y. Coopedge: A decentralized blockchain-based platform for cooperative edge computing. In Proceedings of the Web Conference 2021, New York, NY, USA, 19–23 April 2021; pp. 2245–2257.
43. Zhang, S.; Gu, H.; Chi, K.; Huang, L.; Yu, K.; Mumtaz, S. Drl-based partial offloading for maximizing sum computation rate of wireless powered mobile edge computing network. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 10934–10948.
44. Li, N.; Iosifidis, A.; Zhang, Q. Graph reinforcement learning-based CNN Inference offloading in dynamic edge computing. In Proceedings of the Globecom 2022-2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 982–987.
45. Peng, H.; Wang, L.C. Energy harvesting reconfigurable intelligent surface for UAV based on robust deep reinforcement learning. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 6826–6838. [CrossRef]
46. Yu, S.; Chen, X.; Zhou, Z.; Gong, X.; Wu, D. When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network. *IEEE Internet Things J.* **2020**, *8*, 2238–2251.
47. Chai, F.; Zhang, Q.; Yao, H.; Xin, X.; Gao, R.; Guizani, M. Joint multi-task offloading and resource allocation for Mobile Edge Computing Systems in Satellite IoT. *IEEE Trans. Veh. Technol.* **2023**, *72*, 7783–7795. [CrossRef]