scientific reports

Check for updates

OPEN

Blockchain-based heterogeneous resource configuration scheme in computing power network

Qiang Gao[1], Chunyu Liu[2], Lei Wang[3], Yiqing Liu[2] & Yueqiang Xu[3]✉

To provide users with on-demand computing and network resources while ensuring service quality, the concept of a computational power network has emerged. However, efficiently allocating these resources to achieve an optimal balance between system energy consumption and latency, as well as ensuring the security of the resource allocation process, remain significant challenges. To solve these challenges, this paper proposes a secure edge offloading optimization framework. Within this framework, we incorporate blockchain technology to record critical network information in blocks within the blockchain, thereby ensuring the security of resource allocation. Subsequently, we formulate a joint optimization problem aimed at optimizing network bandwidth, computational resources, and blockchain maintenance resources simultaneously to achieve the best possible balance between system latency and energy consumption. Given the complexity of the original problem, we decompose it into two subproblems and solve each using the Lagrange multiplier iterative algorithm. To validate the effectiveness of our proposed solution, we compare it with several baseline approaches, including those based on Particle Swarm Optimization (PSO) and reinforcement learning methods such as Proximal Policy Optimization (PPO). Simulation results demonstrate that our proposed scheme performs notably better in achieving an optimal balance between system energy consumption and latency.

Keywords Computing power network, Resource awareness, Blockchain, Resource optimization

Background

Amid the rapid advancement of digitalization, the widespread adoption of smart devices and the surging computational demands of emerging applications have posed unprecedented challenges to traditional local computing paradigms. These applications not only require processing massive amounts of data but also demand real-time responsiveness and ultra-low latency, placing immense pressure on the hardware capabilities of devices themselves. However, constrained by physical size, battery life, and other factors, mobile devices often struggle to deliver sufficient computational power to meet these high-performance requirements.

To solve this problem, computing power network has emerged as a pivotal technology to meet the growing demand for high-performance computing by efficiently integrating and allocating distributed computing resources. It leverages advanced networking technologies and intelligent scheduling algorithms to seamlessly connect widely distributed edge servers, forming a powerful virtual computing platform. This network not only enhances computing efficiency and resource utilization but also supports various application scenarios, such as big data analytics, artificial intelligence training, and augmented reality services, providing users with a smoother and more efficient experience. By transferring compute-intensive tasks from resource-constrained end devices to edge servers, computation offloading effectively alleviates the computational burden on mobile devices, enhances processing efficiency, and reduces energy consumption. This technology plays a critical role in advancing 5G and future communication networks, serving as a cornerstone for building an intelligently connected world.

Motivation

However, computation offloading in computing power networks still faces several key challenges. First, task processing demands substantial computational resources, significantly increasing energy consumption at edge nodes. While reducing computational resources is an effective way to lower energy usage, this approach

[1]School of Information Engineering, Xi'an University, Xi'an 710065, China. [2]Joint Laboratory for Space Information Network Frequency and Orbit Technology and Applications, China Satellite Network System Research Institute Co., Ltd, Beijing 100094, China. [3]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. ✉email: yueqiang@ustb.edu.cn

may adversely impact transmission latency. Consequently, achieving an optimal balance between latency and energy efficiency remains a critical challenge. Second, computation offloading requires transferring data to edge servers, exposing the system to considerable security risk, particularly in untrusted environments where malicious attackers may be present. Specifically, malicious nodes can disrupt resource allocation decisions, leading to inefficient resource utilization and degraded quality of experience (QoE) for users. While blockchain-based frameworks offer a promising solution by encapsulating critical information in immutable transactions to enable tamper-proof distributed fraud prevention, they introduce substantial computational overhead. This creates a dual challenge: maintaining both blockchain performance and efficient computational task processing simultaneously.

Our contributions

To solve these issues, we propose a secure task offloading architecture that incorporates blockchain technology to ensure offloading security. The key contributions of this work are as follows:

(1) We propose a secure edge computing offloading framework. In this framework, we integrate blockchain technology to record critical network information in blocks, thereby ensuring the security and transparency of resource allocation decisions.
(2) We formulate a joint optimization problem that simultaneously considers network bandwidth, computational resources, and blockchain maintenance requirements. Our objective is to achieve an optimal trade-off between system energy consumption and service latency.
(3) We address the complexity of the original problem by decomposing it into two subproblems. To solve each subproblem efficiently, we apply the Lagrange multiplier iteration method, enabling us to derive optimal solutions in an iterative manner.
(4) We evaluate the performance of our proposed approach through extensive simulations. We compare our method with several baseline strategies, including Particle Swarm Optimization (PSO) and reinforcement learning-based Proximal Policy Optimization (PPO). The results show that our scheme significantly outperforms existing methods in balancing energy consumption and system delay.

Related work

Alzubi et al.[1] Proposed a technique that utilized deep learning with a convolutional neural network to classify user data, alongside blockchain technology integrated with a cryptography-based federated learning module, to preserve the privacy of electronic health records. Alzubi et al.[2] proposed a blockchain and artificial intelligence-enabled secure medical data transmission model for IoT-based healthcare systems, which employed signcryption for secure data transfer and a modified discrete particle swarm optimization algorithm with a wavelet kernel extreme learning machine for disease detection. Alzubi et al.[3] proposed a blockchain-based security framework for MEC, ensuring secure and efficient data management with low overhead. Jafar et al.[4] provided an overview of the Internet of Things (IoT), highlighting its applications across various domains and discussing key challenges related to security, privacy, data management, and infrastructure, with the goal of offering a broad understanding of IoT's vision and potential. Zhao et al.[5] proposed a deep reinforcement learning-based resource allocation algorithm for ultra-reliable low-latency communication (URLLC) in 6G networks, achieving lower round-trip delay and improved performance compared to existing methods. Su et al.[6] proposed a resource allocation approach combining mean-field game theory and multi-agent reinforcement learning for UAV-assisted V2X communication networks, improving energy efficiency and outperforming existing methods under power and QoS constraints. Shang et al.[7] developed a deep reinforcement learning-based resource allocation scheme for integrated sensing and communication (ISAC) systems in vehicular networks, achieving higher spectral efficiency and communication rates compared to existing methods. Zhang et al.[8] investigated joint caching and resource allocation in cooperative MEC systems. The authors developed a hierarchical reinforcement learning framework that decomposed the problem into lower-layer and upper-layer optimizations, validated through simulations to enhance resource utilization and balance workloads among servers. Zhang et al.[9] explored energy-efficient task offloading in air-ground integrated MEC systems. The study designed a distributed online algorithm using stochastic optimization and game theory to address dynamic task arrivals and resource competition, proposing DGMS and TPA sub-algorithms to minimize energy consumption while ensuring queue stability under HAP-UAV collaboration[10] established a maritime MEC network with EH, formulated a stochastic optimization problem, and proposed JCORA algorithm to maximize long-term throughput. Liu et al.[11] conducted research on joint resource allocation and intrusion prevention system deployment for edge computing to enhance security and resource utilization. Liu et al.[12] presented a deep reinforcement learning-based method for joint service migration and resource allocation in edge IoT systems to adapt to dynamic environments. Zhang et al.[13] addressed the problem of joint terminal pairing and multi-dimensional resource allocation for cooperative computation in a WP-MEC system to improve cooperation efficiency. Chen et al.[14] devised an algorithm integrating deep reinforcement learning and Lyapunov optimization for joint trajectory optimization and resource allocation in UAV-MEC systems to minimize MU energy consumption. Zhang et al.[15] investigated a mobility-aware and double auction-based algorithm for joint task offloading and resource allocation in MEC to improve success rate and system revenue. Li et al.[16] investigated mobility-aware computation offloading and resource allocation for NOMA MEC in vehicular networks to minimize average task completion latency. Xu et al.[17] solved the task scheduling and resource allocation problem in self-powered sensor systems by formulating and optimizing a stochastic problem, reducing energy consumption. Zhang et al.[18] formulated an algorithm featuring partial offloading and resource allocation for MEC-assisted vehicular networks to minimize the weighted sum of delays and energy consumption. Qin et al.[19] introduced a new MEC architecture and proposed decentralized optimization schemes for task offloading and resource allocation, improving uplink throughput and reducing

delay. Zhang et al.[20] proposed a UAV-MEC-blockchain integrated scheme that optimized IoT resource allocation in remote areas, reducing energy consumption and latency while improving coverage and security, with ADMM ensuring stability. Dong et al.[21] developed a blockchain-based distributed secure resource allocation scheme using convex optimization consensus, enhancing security and enabling differentiated resource sharing for large-scale IoT access. Ma and Li[22] created a blockchain-powered MEC video streaming framework that optimized resource allocation using game theory to balance service quality and profits while ensuring security through blockchain. Wang et al.[23] established a blockchain-based secure data resource allocation model that optimized scheduling configuration, significantly reducing allocation load, resource consumption and latency while improving scheduling efficiency. Jiang et al.[24] introduced a blockchain-sharding-based secure sharing mechanism that employed a dual-layer PPO algorithm to optimize resource allocation, enhancing transaction throughput in digital twin edge networks while mitigating mapping error impacts. Du et al.[25] constructed a blockchain-driven MEC joint optimization algorithm that reduced IoT energy consumption via DDPG adaptive decision-making while enhancing trustworthiness and QoS, with simulations demonstrating superiority over conventional approaches. Yang et al.[26] formulated an MEC-blockchain integrated industrial internet optimization method that employed deep reinforcement learning to reduce energy consumption and computational overhead while dynamically optimizing offloading decisions and block size, thereby improving system performance. Liu et al.[27] constructed a blockchain-fog computing integrated task offloading scheme that optimized mobile terminals' cryptocurrency earnings based on node resources and latency, employing simulated annealing for solution, with simulations validating effectiveness.

Table 1 provides a summary of the differences between our proposed approach and key existing works. Compared with existing works, the main contributions of this paper can be summarized as follows:

(1) We propose a secure edge offloading architecture that incorporates blockchain to ensure the security of data processing.
(2) We present a joint optimization of bandwidth resources, computing resources, and blockchain maintenance resources, achieving an optimal balance of energy consumption and latency within the system.
(3) We design an efficient algorithm that decomposes the original problem and achieves the optimal solution of the optimization problem based on convex optimization theory.

Problem description and modeling

Problem description

As shown in Fig. 1, each node in the computing network has computing and communication capabilities, and each node can serve a certain number of users. Users access the edge nodes by wireless means, and the computing tasks generated by them need to be offloaded to the edge nodes and retrograded. The edge nodes in the network need to provide bandwidth resources and computing resources for each user to ensure the service requirements of users. As a top-level resource management architecture, blockchain can be deployed on edge nodes, and the operation of blockchain can be realized by using the resource capabilities of edge nodes. At the same time, resource security management is also realized by using encryption technology and distributed formula technology, so as to ensure the security of resource allocation.

Problem modeling

When modeling, Define $N = \{1, \ldots, \mathrm{n}, \ldots \mathrm{N}\}$ as the set of edge nodes. The set of users accessing node N can be defined as $U_n = \{1, \ldots, \mathrm{u_n}, \ldots \mathrm{U_n}\}$. The task attributes of users in the network can be represented by a triple, namely $D_{u_n} = \{I_{u_n}, O_{u_n}, T_{u_n}\}$, where $I_{u_n}, O_{u_n}, T_{u_n}$ represent the task size (unit: bit), computational complexity (unit: cycle/bit), and maximum tolerable delay (unit: s) of user u_n, respectively. Users need to send their own tasks to the edge node side, and its communication rate $R_{u_n,n}$ can be expressed as Eq. (1):

References	Bandwidth resource allocation	Computing resource allocation	Blockchain resource allocation	Performance trade-off between energy and latency	User QoE assurance	Blockchain performance assurance
[1,26]	×	√	√	×	√	√
[2]	×	×	√	×	√	√
[3]	×	×	√	×	×	√
[4]	×	×	×	×	×	×
[5,7,8,11]	√	√	×	√	×	×
[6,9,10,12–19]	√	√	×	√	√	×
[20]	√	√	√	×	√	√
[21]	√	×	√	×	×	√
[22,24,25,27]	√	√	√	×	√	√
[23]	√	√	√	×	×	√
Our work	√	√	√	√	√	√

Table 1. Our contributes compared with existing works.

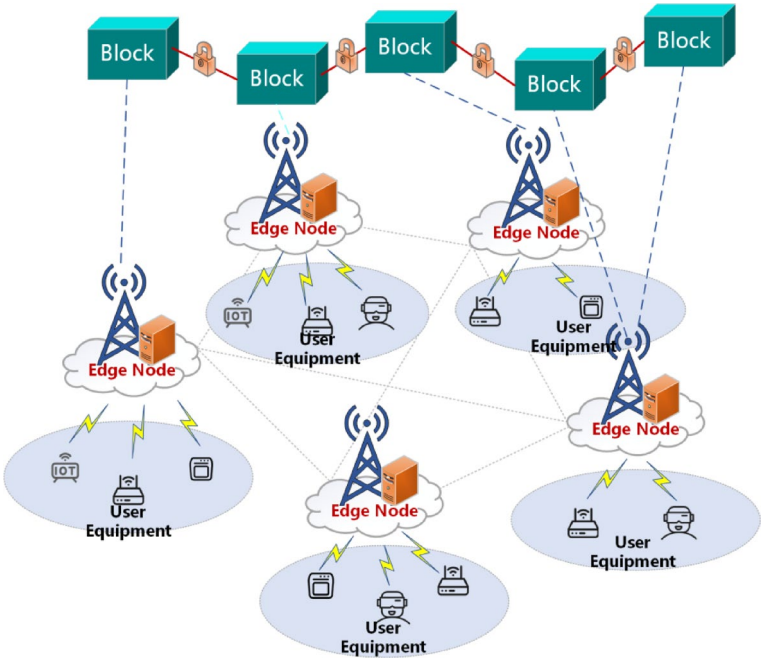

Fig. 1. The system architecture diagram.

$$R_{u_n,n} = b_{u_n,n} \cdot \log\left(1 + \frac{p_{u_n,n} \cdot \varphi_{u_n,n}}{b_{u_n} \cdot N_0}\right) \tag{1}$$

N_0 is the noise density.

Subsequently, the data offloading delay $T^{TRA}_{u_n,n}$ of the node can be obtained as Eq. (2):

$$T^{TRA}_{u_n,n} = \frac{I_{u_n}}{R_{u_n,n}} \tag{2}$$

After the edge node receives the user's data, it needs to allocate computing resources. Let $f_{u_{n,n}}$ be the computing resources allocated by node n to user u_n. Then the time required to compute the offloaded data of user u_n can be expressed as Eq. (3):

$$T^{COM}_{u_{n,n}} = \frac{I_{u_n} \cdot O_{u_n}}{f^{\mathrm{PRO}}_{u_{n,n}}} \tag{3}$$

Therefore, the task processing delay $\mathrm{T}^{\mathrm{TOT}}_{u_n}$ of the user u_n can be calculated and expressed as Eq. (4):

$$T^{TOT}_{u_{n,n}} = T^{COM}_{u_{n,n}} + T^{TRA}_{u_{n,n}} \tag{4}$$

During the task processing, the energy consumption generated by the node mainly includes transmission energy consumption and computing energy consumption. Let $E^{TRA}_{u_{n,n}}$ and $E^{COM}_{u_{n,n}}$ be the transmission energy consumption and computing energy consumption during the task processing respectively. These two parts can be expressed by the following formula as Eq. (5):

$$\begin{cases} E^{TRA}_{u_{n,n}} = p_{u_n,n} \cdot T^{TRA}_{u_{n,n}} \\ E^{COM}_{u_{n,n}} = \alpha \cdot f^{\mathrm{PRO}^3}_{\mathrm{n}} \cdot T^{COM}_{u_n} \end{cases} \tag{5}$$

In the above formula (5), α is the energy consumption parameter of the edge node, which is mainly related to the CPU performance of the node. Therefore, the total energy consumption of task processing can be expressed as Eq. (6):

$$E^{TOT}_{u_{n,n}} = E^{TRA}_{u_{n,n}} + E^{COM}_{u_{n,n}} \tag{6}$$

In the network, malicious nodes may tamper with critical information, thereby affecting the quality of computation offloading. To ensure the security of computation offloading, this paper introduces blockchain technology to record key information in the network (such as resource allocation decisions) onto the blockchain.

The operation of blockchain consumes certain computational and communication resources. To improve system efficiency while maintaining security and a reasonable level of decentralization, this paper adopts the Delegated Proof-of-Stake (DPoS) consensus mechanism. Under this mechanism, nodes in the network elect a group of trusted nodes as block producers through a voting process. These block producers are responsible for transaction validation and block generation. They perform their duties in a predetermined order and are subject to supervision by other nodes. If a block producer behaves maliciously or fails to meet performance requirements, it can be replaced through a community vote, thereby ensuring the overall stability and reliability of the system. Referring to reference[28], the maintenance of the blockchain is mainly divided into three steps: 1) block generation; 2) block distribution; and 3) distributed consensus. Below, we will provide a detailed introduction to each step.

(1) *Block generation*: Referring to reference[28], edge nodes in the network select block-producing nodes through the improved DPoS consensus. This is an efficient consensus scheme based on an election mechanism, which can improve consensus efficiency and conserve network resources. The block-producing nodes are responsible for packaging transaction information in the network. The packaging process requires a certain amount of computing power resources. Let D_n^{BLO} be the transaction information packaged by the block generation node, then the time consumed for block generation can be expressed as

$$T_n^{GEN} = \frac{\varphi \cdot D_n^{BLO}}{f_n^{GEN}}, \tag{7}$$

where φ is the block processing density, f_n^{GEN} is the allocated computing resources. The energy consumption generated during the block generation process can be expressed as

$$E_n^{GEN} = \alpha \cdot f_n^{GEN3} \cdot T_{u_{n,n}}^{GEN}. \tag{8}$$

(2) *Block broadcasting*: After the block generation node completes the block, it will broadcast the block in the network. Each node in the network is connected via high-speed fiber optics. Let $R_{n,n'}^{TRA}$ be the wired transmission rate between nodes n and n′, then the block broadcasting delay can be expressed as

$$T_n^{BRO} = \frac{D_n^{BLO}}{\min\limits_{n \neq n'} R_{n,n'}^{TRA}}. \tag{9}$$

(3) *Block verification*: After receiving the block, each node in the network needs to verify its legitimacy. The verification process also requires a certain amount of computing resources. Let f_n^{VER} be the verification resources invested by node n, and ϵ be the number of CPU cycles required in the block verification process (unit: cycles), then the delay required for all nodes in the network to complete block verification can be expressed as

$$T_n^{VER} = \frac{\int}{\min\limits_{n \neq n'} f_n^{VER}}. \tag{10}$$

Similarly, the energy consumption generated during the block verification process can be expressed as

$$E_n^{VER} = \alpha \cdot f_n^{VER3} \cdot T_n^{VER}. \tag{11}$$

In summary, the time latency required for the blockchain to complete information recording and achieve finality can be expressed as

$$T_n^{BLO} = T_n^{GEN} + T_n^{BRO} + T_n^{VER} \tag{12}$$

The total energy consumption of the blockchain can be expressed as

$$E_n^{BLO} = E_n^{GEN} + E_n^{VER} \tag{13}$$

Considering the energy consumption levels of wireless transmission and wired transmission, the energy consumption in the wired transmission process is ignored in the above formula.

Optimization problem construction

Edge nodes configure parameters $\delta = \left\{b_{u_n,n}, f_{u_n,n}^{PRO}, f_n^{GEN}\right\}$ to ensure the quality of service for users and the efficiency of resource awareness, while also ensuring that their own energy consumption is minimized. The quality of service for users can be measured by the task processing delay.

In order to improve the efficiency of resource awareness, it is necessary to increase the investment in computing resources to reduce the delay of the blockchain. The improvement of blockchain performance will

cause additional energy consumption. Therefore, how to achieve a trade-off between the two is a problem. For this reason, this paper sets the optimization objective as Eq. (14):

$$\begin{aligned}
&\boldsymbol{P0}: \min_{\delta} \sum_{u_n \in \boldsymbol{U_n}} \sum_{\boldsymbol{n} \in \boldsymbol{N}} \boldsymbol{w}_1 \cdot \left(\boldsymbol{T}_{\boldsymbol{u_n,n}}^{\boldsymbol{TOT}}\right) + \boldsymbol{w}_2 \cdot \left(\boldsymbol{E}_{\boldsymbol{n}}^{\boldsymbol{BLO}} + \boldsymbol{E}_{\boldsymbol{u_{n,n}}}^{\boldsymbol{TOT}}\right) \\
&\boldsymbol{s.t.}\ \boldsymbol{C1}: \sum_{u_{n \in U_n}} b_{u_n,n} \leq B_n^{MAX} \\
&\quad \boldsymbol{C2}: f_n^{VER} + f_n^{GEN} + \sum_{u_{n \in U_n}} f_n^{PRO} \leq F_n^{MAX} \\
&\quad \boldsymbol{C3}: T_n^{GEN} + T_n^{BRO} + T_n^{VER} \leq T_{MAX}^{BLO} \\
&\quad \boldsymbol{C4}: T_{u_{n,n}}^{COM} + T_{u_{n,n}}^{TRA} \leq T_{MAX}^{BRO}
\end{aligned} \tag{14}$$

In the above formula, C1 indicates that the bandwidth resources allocated by each edge node to the user cannot exceed its maximum resource capacity. C2 indicates that the maximum transmit power on the user side cannot exceed its maximum resource capacity. C3 indicates that the computing resources configured by the edge node cannot exceed the sum of its total resources. C4 indicates that the delay of the blockchain system should not be greater than a certain security threshold. C5 indicates that the task processing delay of the user should not be greater than its maximum tolerable delay.

After optimization, the remaining resources of the edge node can be expressed as Eq. (15):

$$\begin{cases}
B_n^{REM} = B_n^{MAX} - b_{u_n,n}^* \\
F_n^{REM} = F_n^{MAX} - f_n^{VER*} + f_n^{GEN*} + \sum_{u_{n \in U_n}} f_n^{PRO*}
\end{cases} \tag{15}$$

In the above formula, $b_{u_n,n}^*, f_n^{VER*}, f_n^{GEN*}, f_{\mathrm{n}}^{\mathrm{PRO}^*}$ are the optimal bandwidth resource configuration, the optimal block verification resource configuration, the optimal block generation resource configuration, and the optimal data processing resource configuration respectively.

The remaining resources of the node can be used as the transaction information for the next stage and packed into the block to achieve the secure sharing of resources between nodes.

Problem solving

The original problem is a multi-variable coupled joint optimization problem. For such joint optimization problems with variable coupling, it is common practice to decompose the problem into smaller subproblems. As shown in Fig. 2, we decompose the original problem into two subproblems: the bandwidth resource allocation subproblem and the computing resource allocation subproblem. Subsequently, we provide mathematical proofs demonstrating that both subproblems are convex optimization problems.

Proposition 1 The decomposed two subproblems are both convex optimization problems, and each has an optimal solution.

Proof Refer to Appendix A.

Then, we introduce the Lagrangian multiplier iteration algorithm, which is a classic approach for solving constrained optimization problems. Its core idea is to transform a constrained optimization problem into an unconstrained one by introducing Lagrangian multipliers.

As shown in Fig. 3, the Lagrange multiplier iteration algorithm mainly consists of three major steps. First, a Lagrangian function is constructed, combining the objective function with all constraints, where each constraint is associated with a Lagrangian multiplier. Next, mathematical derivations of the Lagrangian function are performed to analyze the relationships between variables and multipliers, identifying the necessary conditions for achieving the optimal solution. This process typically relies on theoretical tools such as the Karush–Kuhn–Tucker (KKT) conditions. Finally, in practical implementations, an iterative approach is employed to gradually approach the optimal solution. The process begins with initial guesses for the variables and multipliers, which are then updated alternately: the variables are first adjusted to minimize the objective function, followed by an update of the multipliers to better enforce the constraints.This process continues until a convergence criterion is met, such as when the changes in variables or multipliers become sufficiently small. This method not only effectively handles equality constraints but also addresses inequality constraints through appropriate transformations, making it a vital tool for solving complex optimization problems.

When the optimal computing resources are given, sub—problem one can be expressed as Eq. (16):

$$\boldsymbol{P1}: \min_{b_{u_n,n}} w_1 \cdot \left(T_{u_n}^{TOT} + T_n^{BLO}\right) + w_2 \cdot \left(E_n^{BLO} + E_{u_{n,n}}^{TOT}\right) \quad s.t.\ \ C1,C3,C4. \tag{16}$$

For sub—problem one, first give the Lagrangian expression of this problem as Eq. (17):

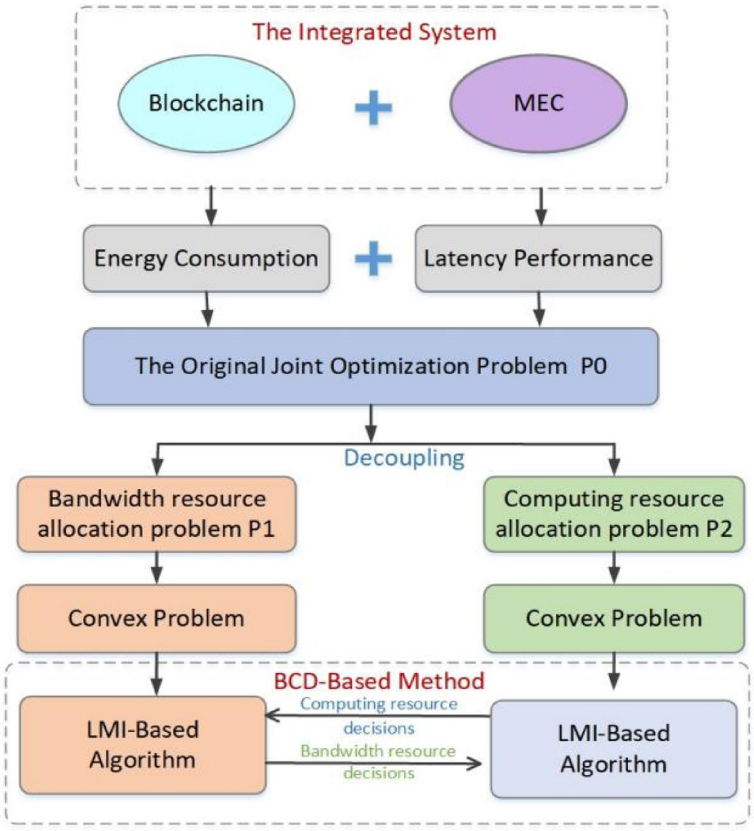

Fig. 2. The framework of the proposed algorithm.

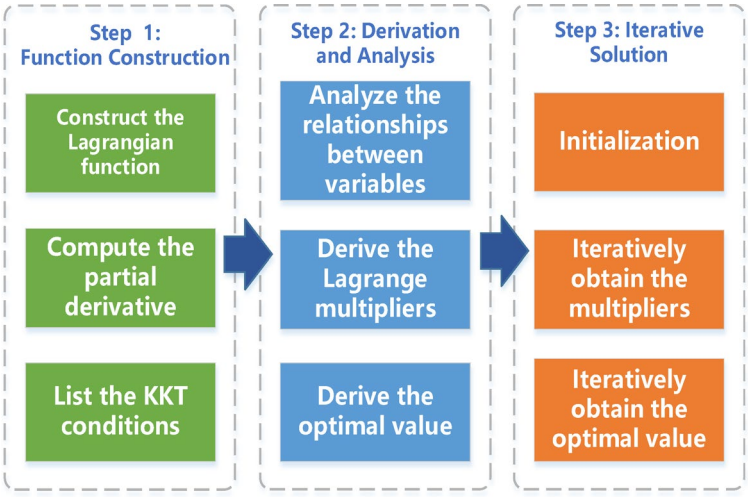

Fig. 3. The framework of Lagrangian multiplier iteration algorithm.

$$L = \sum_{n\in N}\sum_{u_n\in U_n} w_1 \cdot \frac{I_{u_n}\cdot O_{u_n}}{f_{u_n,n}^{PRO}} + w_1 \cdot \frac{I_{u_n}}{R_{u_n,n}} + w_2 \cdot E_n^{GEN} + w_2 \cdot E_n^{VER} + w_2 \cdot p_{u_n,n} \cdot T_{u_n,n}^{TRA}$$
$$+w_2 \cdot \alpha \cdot f_n^{PRO^3} \cdot T_{u_n}^{COM} + \alpha_n \cdot \left(\sum_{u_n\in U_n} b_{u_n,n} - B_n^{MAX}\right) + \beta_n \cdot \left(T_n^{GEN} + T_n^{BRO} + T_n^{VER} - T_{MAX}^{BRO}\right) \quad (17)$$
$$+\delta_{u_n,n} \cdot \left(T_{u_n,n}^{COM} + T_{u_n,n}^{TRA} - T_{MAX}^{PRO}\right)$$

In the above formula, $\alpha_n, \delta_{u_n,n}, \gamma_n$ is the Lagrange multiplier.

Take the first-order derivative of L with respect to $b_{u_n,n}$, and the following formula can be obtained as Eq. (18):

$$\frac{\partial L}{\partial b_{u_n,n}} = \mu_{u_n,n} \cdot \frac{\frac{p_{u_n,n}\cdot\varphi_{u_n,n}}{\ln 2\cdot\left(b_{u_n}\cdot N_0 + p_{u_n,n}\cdot\varphi_{u_n,n}\right)} - \vartheta_{u_n,n}}{\left(b_{u_n,n}\cdot\vartheta_{u_n,n}\right)^2} + \alpha_n \quad (18)$$

In the above Eq. (18), $\mu_{u_n,n}$ and $\vartheta_{u_n,n}$ can be respectively expressed as Eq. (19):

$$\begin{cases} \mu_{u_n,n} = log\left(1 + \dfrac{p_{u_n,n}\cdot\varphi_{u_n,n}}{b_{u_n}\cdot N_0}\right) \\ \vartheta_{u_n,n} = I_{u_n}\cdot\left(w_1 + w_2\cdot p_{u_n,n} + \delta_{u_n,n}\right) \end{cases} \quad (19)$$

To obtain the optimal $b_{u_n,n}$, the Newton's method can be used, and its update formula can be expressed as Eq. (20):

$$b_{u_n,n}[l+1] = b_{u_n,n}[l] - \frac{W\left(b_{u_n,n}[l]\right)}{W\left(b_{u_n,n}[l]\right)'} \quad (20)$$

In the above equations, $W\left(b_{u_n,n}[l]\right)'$ and $W\left(b_{u_n,n}[l]\right)$ can be expressed as Eq. (21):

$$\left(b_{u_n,n}[l]\right) = \mu_{u_n,n} \cdot \frac{\frac{p_{u_n,n}\cdot\varphi_{u_n,n}}{\ln 2\cdot\left(b_{u_n}\cdot N_0 + p_{u_n,n}\cdot\varphi_{u_n,n}\right)} - \vartheta_{u_n,n}}{\left(b_{u_n,n}\cdot\vartheta_{u_n,n}\right)^2} + \alpha_n \quad (21)$$

$$LW\left(b_{u_n,n}[l]\right)' = \frac{\mu_{u_n,n}}{\left(b_{u_n,n}\cdot\vartheta_{u_n,n}\right)^3} \cdot \frac{\vartheta_{u_n,n}\cdot\left(p_{u_n,n}\cdot\varphi_{u_n,n}\right)^2}{\left(\ln 2\cdot\left(b_{u_n}\cdot N_0 + p_{u_n,n}\cdot\varphi_{u_n,n}\right)\right)^2}$$
$$+\frac{2\cdot\mu_{u_n,n}}{\left(b_{u_n,n}\cdot\vartheta_{u_n,n}\right)^3}\left(\frac{p_{u_n,n}\cdot\varphi_{u_n,n}}{\ln 2\cdot\left(b_{u_n}\cdot N_0 + p_{u_n,n}\cdot\varphi_{u_n,n}\right)} - \vartheta_{u_n,n}\right)^2 \quad (22)$$

Subsequently, the Lagrange multipliers in the above solution process are updated. The update process is shown in Eq. (23):

$$\begin{cases} \alpha_n^{i+1} = \alpha_n^i + S_n^1 \cdot \left(\sum_{u_n\in U_n} b_{u_n,n} - B_n^{MAX}\right) \\ \beta_n^{i+1} = \beta_n^i + S_n^2 \cdot \left(\left(T_n^{GEN} + T_n^{BRO} + T_n^{VER} - T_{MAX}^{BLO}\right)\right) \\ \delta_n^{i+1} = \delta_n^i + S_n^3 \cdot \left(T_{u_n,n}^{COM} + T_{u_n,n}^{TRA} - T_{MAX}^{PRO}\right) \end{cases} \quad (23)$$

In the above equation, S_n^1, S_n^2, S_n^3 is the step size of the iteration.

When the optimal bandwidth resources are given, sub-problem two can be expressed as Eq. (24):

$$\boldsymbol{P2}: \min_{f_{u_n,n}^{PRO} f_n^{GEN}} w_1 \cdot \left(T_{u_n}^{TOT} + T_n^{BLO}\right) + w_2 \cdot \left(E_n^{BLO} + E_{u_n,n}^{TOT}\right) \quad s.t.\ C2, C3, C4. \quad (24)$$

Similarly, the Lagrangian equation of the sub-problem 2 is given by

$$
\begin{aligned}
L = & \sum_{n\in N}\sum_{u_n\in U_n} w_1\cdot\frac{I_{u_n}\cdot O_{u_n}}{f_{u_n,n}^{PRO}} + w_1\cdot\frac{I_{u_n}}{R_{u_n,n}} + w_2\cdot E_n^{GEN} + w_2\cdot E_n^{VER} \\
& + w_2\cdot p_{u_n,n}\cdot T_{u_n,n}^{TRA} + w_2\cdot\alpha\cdot f_n^{PRO^3}\cdot T_{u_n}^{COM} \\
& + \gamma_n\cdot\left(\left(f_n^{VER} + f_n^{GEN} + \sum_{u_n\in U_n} f_{u_n,n}^{PRO}\right) - F_n^{MAX}\right) \\
& + \beta_n\cdot\left(T_n^{GEN} + T_n^{BRO} + T_n^{VER} - T_{MAX}^{BRO}\right) \\
& + \delta_n\cdot\left(T_{u_n,n}^{COM} + T_{u_n,n}^{TRA} - T_{MAX}^{BRO}\right)
\end{aligned}
\tag{25}
$$

In order to obtain the optimal computing resources, the first-order derivative of L with respect to $f_{u_n,n}^{PRO}$ and f_n^{GEN} is can be expressed as

$$
\frac{\partial L}{\partial f_{u_n,n}^{PRO}} = -\frac{\left(w_1\cdot I_{u_n}\cdot O_{u_n} + \delta_n\cdot I_{u_n}\cdot O_{u_n}\right)}{f_{u_n,n}^{PRO^2}} + 2\cdot w_2\cdot\alpha\cdot f_n^{PRO}\cdot I_{u_n}\cdot O_{u_n} + \gamma_n \tag{26}
$$

$$
\frac{\partial L}{\partial f_n^{GEN}} = 2\cdot w_2\cdot\alpha\cdot f_n^{GEN}\cdot\varphi\cdot D_n^{BLO} \quad + \gamma_n - \frac{\beta_n\cdot\varphi\cdot D_n^{BLO}}{f_n^{GEN^2}} \tag{27}
$$

According to the Karush–Kuhn–Tucker (KKT) conditions, we can obtain $\frac{\partial L}{\partial f_{u_n,n}^{PRO}} = 0$ and $\frac{\partial L}{\partial f_n^{GEN}} = 0$. Hence, we have the following conclusions

$$
\begin{cases}
-\dfrac{\left(w_1\cdot I_{u_n}\cdot O_{u_n} + \delta_n\cdot I_{u_n}\cdot O_{u_n}\right)}{f_{u_n,n}^{PRO^2}} + 2\cdot w_2\cdot\alpha\cdot f_n^{PRO}\cdot I_{u_n}\cdot O_{u_n} + \gamma_n = 0 \\
\dfrac{\partial L}{\partial f_n^{GEN}} = 2\cdot w_2\cdot\alpha\cdot f_n^{GEN}\cdot\varphi\cdot D_n^{BLO} \quad + \gamma_n - \dfrac{\beta_n\cdot\varphi\cdot D_n^{BLO}}{f_n^{GEN^2}} = 0
\end{cases}
\tag{28}
$$

Based on the KKT conditions, we can derive that $\gamma_n\cdot\left(\left(f_n^{VER} + f_n^{GEN} + \sum_{u_n\in U_n} f_{u_n,n}^{PRO}\right) - F_n^{MAX}\right) = 0$.

In this paper, we assume that the resources of edge nodes are sufficient. Hence, we can obtain that $\left(\left(f_n^{VER} + f_n^{GEN} + \sum_{u_n\in U_n} f_{u_n,n}^{PRO}\right) - F_n^{MAX}\right) > 0$ and $\gamma_n = 0$. Based on Eq. (28), the optimal computing resource allocation decision can be shown as

$$
f_{u_n,n}^{PRO*} = \sqrt[3]{\frac{(w_1+\delta_n)}{2\cdot w_2\cdot\alpha}} \tag{29}
$$

$$
f_n^{GEN*} = \sqrt[3]{\frac{\beta_n}{2\cdot w_2\cdot\alpha}} \tag{30}
$$

Subsequently, the Lagrange multipliers in the above solution process are updated, and the update process is shown in Eq. (31):

$$
\begin{cases}
\alpha_n^{i+1} = \alpha_n^i + C_n^1\cdot\left(f_n^{VER} + f_n^{GEN} + \sum_{u_n\in U_n} f_{u_n,n}^{PRO} - F_n^{MAX}\right) \\
\beta_n^{i+1} = \beta_n^i + C_n^2\cdot\left(\left(T_n^{GEN} + T_n^{BRO} + T_n^{VER} - T_{MAX}^{BLO}\right)\right) \\
\delta_n^{i+1} = \delta_n^i + C_n^3\cdot\left(T_{u_n,n}^{COM} + T_{u_n,n}^{TRA} - T_{MAX}^{PRO}\right)
\end{cases}
\tag{31}
$$

In the above equations, C_n^1, C_n^2, C_n^3 is the iteration step size. Based on the above analysis.

Results

Simulation settings

In this section, a comprehensive simulation verification of the proposed scheme will be carried out. The simulation software we use is PyCharm, and the programming language is PYTHON. Five edge nodes are set in the network, with each node capable of serving up to 3 users. The number of edge nodes and users in the network can be expanded according to the simulation requirements. In this scenario, the user locations are fixed. Other simulation parameters are detailed in Table 2.

Parameter	Value
Computing power of the edge node: F_n^{MAX}	50 GHz
Bandwidth capacity of the edge node: B_n^{MAX}	40 GHz
User data processing density: O_{u_n}	500 cycle/bit
User task size: I_{u_n}	[0.1,1] Mbit
Block data processing density: φ	700 cycle/bit
Computational resources required for block verification: ϵ	1 GHz
Number of cycles required for block verification: f_n^{VER}	10^7 cycles
Maximum tolerable delay for users: T_{MAX}^{BRO}	200 ms
Blockchain delay threshold: T_{MAX}^{BLO}	200 ms
Block size: D_n^{BLO}	1 Mbit
Energy consumption coefficient: α	10^{-27}
Matching coefficient: w_1, w_2	10^2, 10^4
Noise density: N_0	-174 dBm/Hz
Uplink transmission channel quality: $\varphi_{u_n,n}$	$\left[1 \times 10^{-14}, 9 \times 10^{-14}\right]$
Transmission power: $p_{u_n,n}$	1 W
Wired transmission rate between edge nodes: $R_{n,n\prime}^{TRA}$	100 Mbit/s
Total CPU cycles of block verification: ϵ	737.5 cycle/bit
Initial value of the Lagrange multiplier: α_n	10^{-7}
Initial value of the Lagrange multiplier: β_n	10^4
Initial value of the Lagrange multiplier: $\delta_{u_n,n}$	10^4
Step size of the iteration: S_n^1	2×10^{-13}
Step size of the iteration: S_n^2	10^7
Step size of the iteration: S_n^3	10^7

Table 2. System simulation parameters.

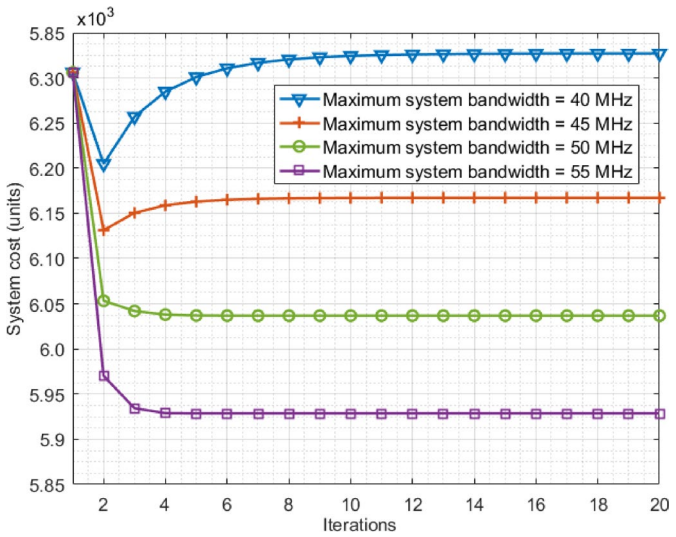

Fig. 4. Algorithm convergence performance.

Simulation analysis

As shown in Fig. 4, we analyzed the convergence performance of the proposed algorithm. First, it can be seen that as the number of algorithm iterations increases, the algorithm can converge within 10 iterations, which also proves that the proposed algorithm has good convergence performance. In addition, we analyzed the impact on the system cost under different system bandwidths. As the bandwidth continuously increases, the system cost will continuously decrease. The main reason is that the larger the system bandwidth, the greater the user data offloading rate. Therefore, the service guarantee delay can be better reduced, the system computing resource configuration can be reduced, and then the overall system cost can be reduced.

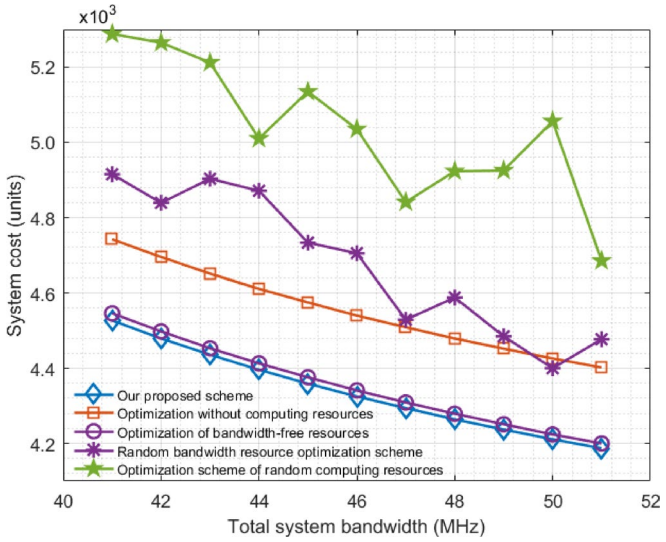

Fig. 5. Performance comparison of different algorithms.

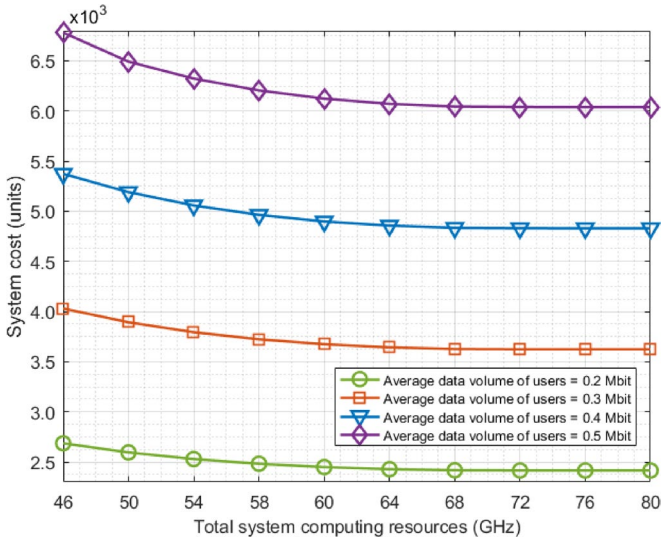

Fig. 6. Performance comparison under different computing power resources.

As shown in Fig. 5, we compared the performance of the proposed scheme with the scheme without computing resource optimization (Scheme 1) under the proposed scheme, the scheme without bandwidth resource optimization (Scheme 2) under the proposed scheme, the random bandwidth resource optimization scheme (Scheme 3), and the random computing resource optimization scheme. It can be seen from the figure that as the total system bandwidth increases, the system cost of all schemes will continuously decrease, and the reason is similar to that in Fig. 2. In addition, we can find that the proposed scheme has better performance advantages in reducing the system cost. The reason is that the proposed scheme jointly optimizes the computing and bandwidth resources in the system, which also proves the effectiveness of the proposed joint optimization configuration of resources.

Figure 6 analyzes the system cost under different computing power resources. It can be seen from the figure that as the system computing resources increase, the system cost first decreases and then tends to be stable. The reason is that limited computing resources will increase the computing delay and the system cost. In addition, as the user data volume increases, the system cost will also increase. The reason is that a larger data volume will additionally increase the computing resources and transmission delay, thus leading to an increase in the system cost.

Figure 7 analyzes the cost changes of the system under different weights. First, from the overall trend, as the system bandwidth increases, the system cost will continuously decrease. Second, under different weights, the system cost is also different. Usually, the smaller the weight, the lower the system cost (for example, the blue bar).

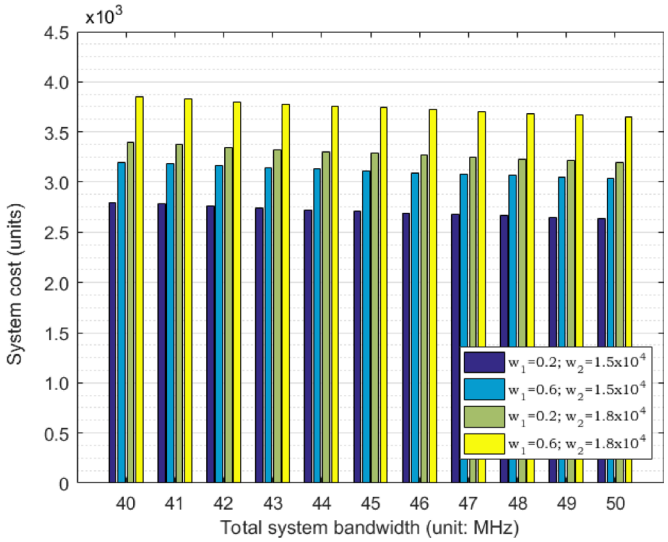

Fig. 7. System cost under different weights.

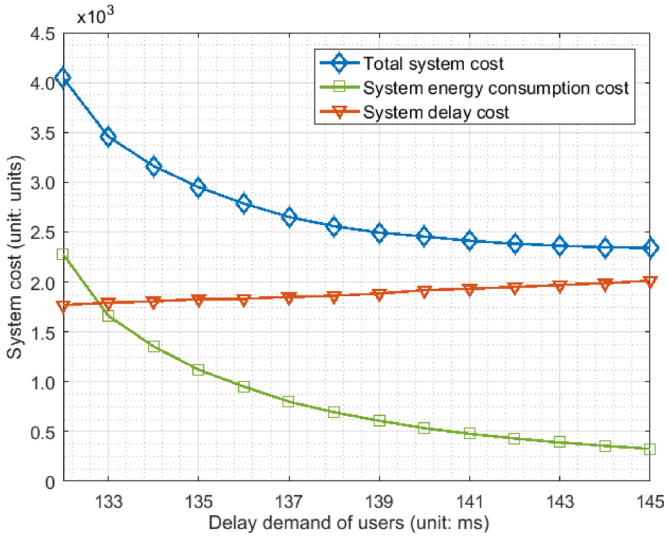

Fig. 8. System cost under different user service requirements.

We can reasonably configure the weights of energy consumption and delay to control and guarantee the service quality of users and control the energy consumption of the system.

Figure 8 analyzes the cost of the system under different user requirements. First, it can be seen that as the user requirements continuously decrease (the service guarantee delay increases), the total system cost and the system energy consumption cost will decrease, while the system delay cost will increase. The main reason is that smaller service guarantee requirements require more computing resources, thus leading to an increase in the system cost. In addition, the decrease in user service requirements (increase in delay) will directly increase the system delay cost.

Figure 9 analyzes the system's resource configuration and system costs. Figure 9a shows the data sizes of different users. We continuously increase the data volumes of Users 1–3. Figure 9b analyzes the resource configuration of different users. The larger the user's data, the more bandwidth resources the system will allocate to them, but the computing resources will not increase. The reason is that more computing resource configurations will consume more energy. To reduce the system cost, the proposed algorithm tends to increase the bandwidth resource configuration. As can be seen from Fig. 9c, the larger the user's data volume, the higher the system cost. In addition, the system's delay cost is always greater than the system's energy consumption cost. The reason lies in the system's weight configuration. We configure a larger weight for the delay to prioritize ensuring the user's service quality.

Figure 10 illustrates the performance comparison between the proposed scheme and existing alternatives, including resource allocation methods based on Proximal Policy Optimization (PPO), Particle Swarm

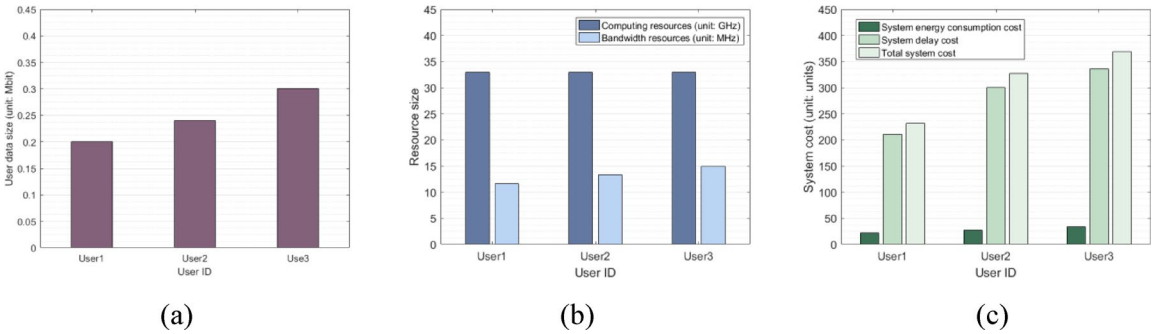

Fig. 9. (**a**) System cost under different user data sizes; (**b**) System cost under different user resource configurations; (**c**) System cost under different conditions.

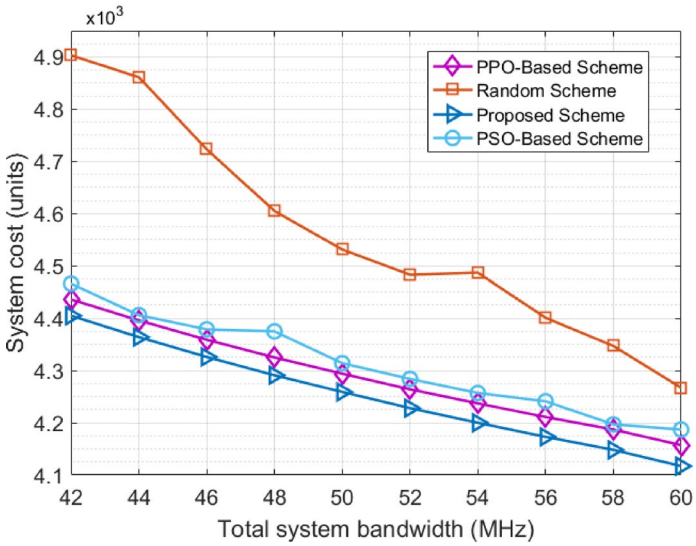

Fig. 10. Performance comparison of the proposed scheme with other schemes.

Optimization (PSO), and a random resource allocation approach. From the chart, it can be observed that as the total system bandwidth increases, the system cost for all four algorithms shows a decreasing trend. This is mainly because a larger system bandwidth reduces the latency of task offloading in the network, thereby utilizing computational resources more efficiently and ultimately lowering the system cost. Furthermore, Fig. 2 demonstrates that our proposed scheme outperforms the other three approaches. This superiority can be attributed to several factors: First, we decomposed the original problem and applied convex optimization techniques to achieve an effective solution. Second, the Particle Swarm Optimization algorithm tends to get trapped in local optima, making it difficult to find the global optimum. Finally, although PPO, as a type of reinforcement learning, aims to search for the optimal solution within the feasible set, its results are highly sensitive to hyperparameter choices and are also prone to local optima issues. Therefore, our scheme avoids local optima while ensuring higher-quality solutions, providing a more efficient and reliable approach to system resource allocation.

Figure 11 illustrates the impact of the number of users and the total network resources on the overall system cost. First, as shown in the figure, increasing the total available bandwidth in the network leads to a continuous decrease in system cost. This is because greater bandwidth reduces data processing latency at the user side, as well as the associated energy consumption, thereby lowering the overall system cost. On the other hand, with the network bandwidth held constant, the system cost increases as the number of users grows. The main reason for this is that each user receives a smaller share of the available bandwidth when more users are added, which results in increased data processing delays. To maintain the required quality of service (QoS), the system must allocate additional computing resources, which in turn increases energy consumption and thus raises the overall system cost.

As shown in Fig. 12, we analyzed the impact of different weights on the system cost. First, as the system bandwidth increases, the cost for all weight combinations decreases. This is because a larger bandwidth reduces network latency, thereby lowering the computational resources required by the system, which in turn reduces the overall system cost. Additionally, we can observe that the system cost varies under different weight

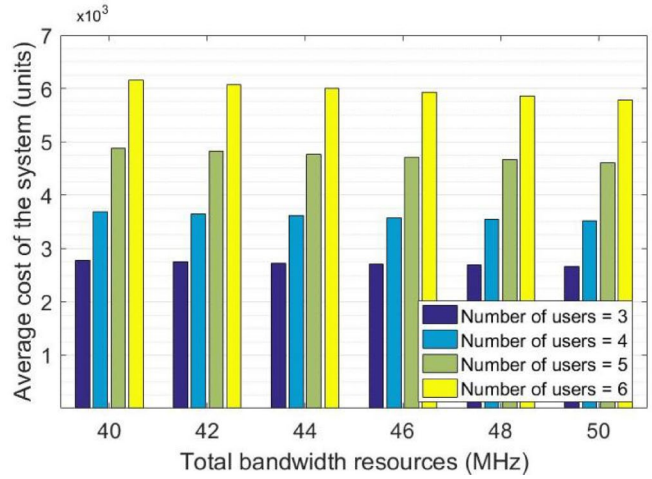

Fig. 11. System performance with Increasing Network Size and total resources.

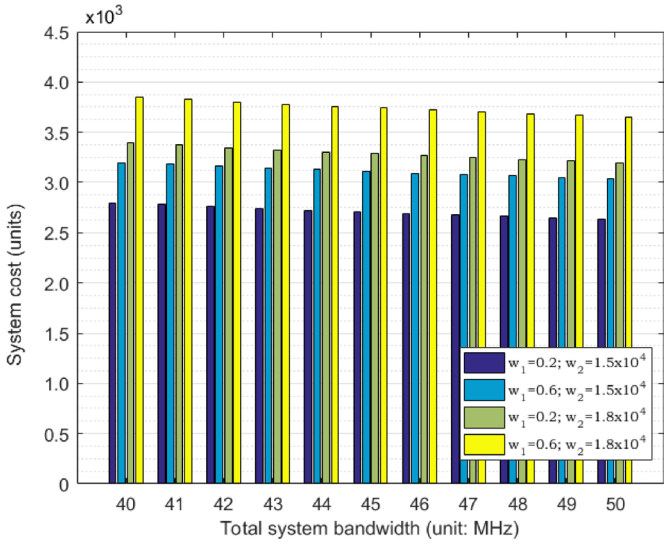

Fig. 12. The impact of different weight combinations on system cost.

combinations. Smaller weight combinations tend to result in lower costs, while larger weight combinations lead to higher system costs.

In Fig. 13, we study the security performance of the proposed system. As the system bandwidth increases, the system cost gradually decreases. This is mainly because, with increased bandwidth, system latency decreases, and the total computational resources required also decrease, leading to a reduction in system cost. Additionally, we compare the performance of the proposed scheme without blockchain (PSwoB). During the simulation process, we introduce two types of deceptive attacks: one exaggerates the computing resource decision (e.g., 20% higher than the actual decision), and the other deflates the computing resource decision (e.g., 20% lower than the actual decision). As shown in Fig. 6, when decision information in the network is maliciously tampered with, the total system cost increases. The reasons can be summarized as follows: 1) When the computing resource decision is maliciously exaggerated, it causes a rapid increase in system energy consumption, leading to higher costs; 2) When the computing resource decision is maliciously reduced, it causes a rapid increase in system latency, which in turn leads to an increase in the total system cost.

Figure 14 analyzes the impact of the system matching coefficient on user Quality of Service (QoS). First, we set the system matching coefficient w2 to 1.5 × 104. As shown in the figure, as the matching coefficient increases, the task processing latency for users also rises. This phenomenon is primarily due to the fact that a larger matching coefficient w1 leads to higher system costs. To achieve an optimal balance, the allocation of computational resources in the system decreases, and since bandwidth resources are limited, user service latency increases. Additionally, as bandwidth increases, user service latency continuously decreases. This is because greater system bandwidth enhances user offloading rates, thereby reducing service latency.

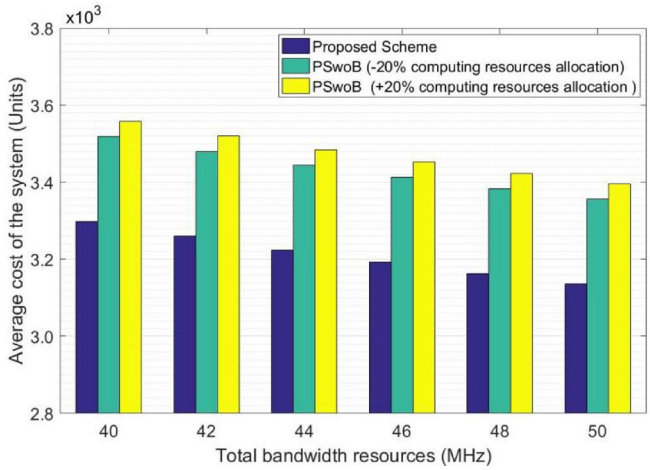

Fig. 13. The security performance of the proposed system.

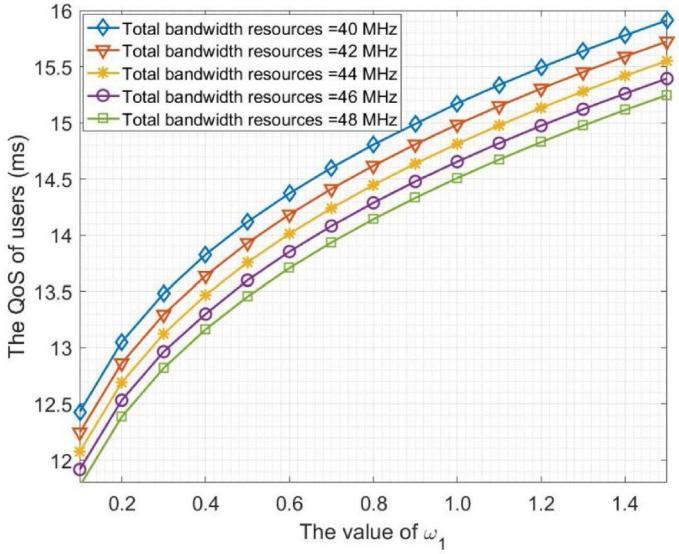

Fig. 14. Impact of system matching coefficient on user QoS.

Conclusions

This paper proposes an efficient resource allocation scheme based on blockchain to address the challenges of secure and efficient joint optimization and allocation of heterogeneous resources in the computing power network. First, the blockchain technology is used to construct a resource joint optimization architecture to ensure the secure management and control of network resources. Second, a system model aiming to minimize the system cost is established to solve the joint optimization problem of bandwidth and computing power resources. Finally, by decomposing the original problem and designing an optimization algorithm based on Lagrange multiplier iteration, the solution complexity is effectively reduced, and the joint optimization and allocation of resources are realized. The simulation results show that this scheme not only ensures the user's service quality but also significantly reduces the resource allocation cost.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 22 March 2025; Accepted: 3 June 2025
Published online: 01 July 2025

References

1. Alzubi, J. A., Alzubi, O. A., Singh, A. & Ramachandran, M. Cloud-IIoT-based electronic health record privacy-preserving by CNN and blockchain-enabled federated learning. *IEEE Trans. Ind. Inf.* **19**(1), 1080–1087 (2023).
2. Alzubi, O. A., Alzubi, J. A., Shankar, K. & Gupta, D. Blockchain and artificial intelligence enabled privacy-preserving medical data transmission in Internet of Things. *Trans Emerg. Telecommun. Technol.* **32**(12), 4316–4360 (2021).
3. Alzubi, J. A., Alzubi, O. A., Singh, A. & Mahmod, A. T. A blockchain-enabled security management framework for mobile edge computing. *Int. J. Netw. Manag.* **33**(5), 2228–2240 (2023).
4. Alzubi, J. A., Manikandan, R., Alzubi, O. A., Gayathri, N. & Patan, R. A survey on Internet of Things architectures. *Int. J. Emerg. Technol.* **10**(1), 47–53 (2019).
5. Zhao, J., Quan, H., Xia, M. & Wang, D. Adaptive resource allocation for mobile edge computing in internet of vehicles: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **73**(4), 5834–5848 (2024).
6. Su, J., Liu, Z., Li, Y., Lee, J. & Guan, X. Communication and computing balanced resource allocation in D2D-based vehicular MEC networks. *IEEE Internet Things J.* **11**(24), 40689–40701 (2024).
7. Shang, C., Sun, Y., Luo, H. & Guizani, M. Computation offloading and resource allocation in NOMA–MEC: A deep reinforcement learning approach. *IEEE Internet Things J.* **10**(17), 15464–15476 (2023).
8. Zhang, W., Zhang, G. & Mao, S. Deep-reinforcement-learning-based joint caching and resources allocation for cooperative MEC. *IEEE Internet Things J.* **11**(7), 12203–12215 (2024).
9. Zhang, L., Zhang, Y. & Zheng, J. Deep reinforcement learning based joint uplink and downlink resource allocation for URLLC. *IEEE Trans. Veh. Technol.* **74**(4), 6048–6063 (2025).
10. Xu, Y. et al. Joint resource allocation for UAV-assisted V2X communication with mean field multi-agent reinforcement learning. *IEEE Trans. Veh. Technol.* **74**(1), 1209–1223 (2025).
11. Liu, C., Xia, M., Zhao, J., Li, H. & Gong, Y. Optimal resource allocation for integrated sensing and communications in internet of vehicles: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **74**(2), 3028–3038 (2025).
12. Liu, F., Yu, H., Huang, J. & Taleb, T. Joint service migration and resource allocation in edge IoT system based on deep reinforcement learning. *IEEE Internet Things J.* **11**(7), 11341–11352 (2024).
13. Zhang, L., Song, Q., Wu, M., Qi, W. & Guo, L. Joint terminal pairing and multi-dimensional resource allocation for cooperative computation in a WP-MEC system. *IEEE Trans. Green Commun. Netw.* **7**(3), 1447–1456 (2023).
14. Chen, Y., Yang, Y., Wu, Y., Huang, J. & Zhao, L. Joint trajectory optimization and resource allocation in UAV-MEC systems: A Lyapunov-assisted DRL approach. *IEEE Trans. Serv. Comput.* **18**(2), 854–867 (2025).
15. Zhang, L., Xiao, K., Jin, L., Dong, P. & Tong, Z. Mobility-aware and double auction-based joint task offloading and resource allocation algorithm in MEC. *IEEE Trans. Netw. Serv. Manag.* **21**(1), 821–837 (2024).
16. Li, Y., Li, L. & Fan, P. Mobility-aware computation offloading and resource allocation for NOMA MEC in vehicular networks. *IEEE Trans. Veh. Technol.* **73**(8), 11934–11948 (2024).
17. Xu, J., Li, K., Chen, Y. & Huang, J. Optimal task scheduling and resource allocation for self-powered sensors in internet of things: An energy efficient approach. *IEEE Trans. Netw. Serv. Manage.* **21**(4), 4410–4420 (2024).
18. Zhang, H. et al. Partial offloading and resource allocation for MEC-assisted vehicular networks. *IEEE Trans. Veh. Technol.* **73**(1), 1276–1288 (2024).
19. Qin, L., Lu, H., Chen, Y., Chong, B. & Wu, F. Toward decentralized task offloading and resource allocation in user-centric MEC. *IEEE Trans. Mob. Comput.* **23**(12), 11807–11823 (2024).
20. Zhang, Y. et al. Resource optimization of UAV-assisted internet of things empowered by MEC and blockchain. *J. Beijing Univ. Technol.* **48**, 935–943 (2022).
21. Dong, Z., Meng, Q., Guo, H., Tian, F. & Zou, Y. Convex optimization resource allocation algorithm in wireless access system based on consortium blockchain. *J. Nanjing Univ. Posts Telecommun. Nat. Sci. Ed.* **41**, 39–45 (2021).
22. Ma, L. & Li, Y. Research on edge computing offloading scheme for video stream based on blockchain. *J. Qingdao Univ. Nat. Sci. Ed.* **37**, 33–40 (2024).
23. Wang, S., Wang, X., Xu, X. & Pan, Z. Method for optimal allocation and scheduling of data resources based on blockchain. *J. Anyang Inst. Technol.* **22**, 55–59 (2023).
24. Jiang, L., Xie, S. & Tian, H. Blockchain sharding and resource adaptive optimization mechanism for digital twin edge Network. *J. Commun.* **44**, 12–23 (2023).
25. Du, J. et al. Joint optimization algorithm for edge computing system driven by blockchain. *J. Xian Univ. Posts Telecommun.* **28**, 1–11 (2023).
26. Yang, L., Li, M., Ye, X., Sun, E. & Zhang, Y. Research on optimal allocation of industrial internet resources integrating edge computing and blockchain. *Chin. High Technol. Lett.* **30**, 1253–1263 (2020).
27. Liu, T., Tang, L., He, X. & Chen, Q. Optimal task offloading scheme based on network delay and resource management in joint blockchain and fog computing system. *J. Electron. Inf. Technol.* **42**, 2180–2185 (2020).
28. Feng, J., Yu, F. R., Pei, Q., Du, J. & Zhu, L. Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems. *IEEE Trans. Wirel. Commun.* **19**(6), 4321–4334 (2020).

Author contributions

Conceptualization: Q.G. and Y.X.; Methodology: Q.G.and L.W.; Data analysis: C.L.; Writing—original draft preparation: Q.G. and Y.X.; Writing—review and editing: Y.L. and L.W.; Visualization: Q.G. and Y.X.; Funding acquisition: Q.G. and Y.X. All authors have read and agreed to the published version of the manuscript.

Funding

This study was funded by the National Natural Science Foundation of China (grant number: 62401041); Young Talent Fund of Xi'an Association for Science and Technology (grant number: 959202413080).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-05560-6.

Correspondence and requests for materials should be addressed to Y.X.