J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 8189--8211

Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on optimal allocation of industrial internet resources based on converged edge computing and blockchain

Hongqiang Wu[1], Shaohua Wang[1], Xinlong Tan[1,✉]

[1] *Inspur Yunzhou Industrial Internet Co., Ltd, Jinan, Shandong, 250101, China*

ABSTRACT

Blockchain theory and its key technologies are developing rapidly, and the industrial internet combined with blockchain technology is driving the realization of safe and reliable comprehensive connectivity in multiple fields. In this paper, we propose a resource optimization allocation method for industrial internet that integrates edge computing and blockchain to reduce the task computing energy consumption and computational overhead of the system while improving the efficiency of the consensus process. This optimization problem is constructed as a Markov decision process, and a deep reinforcement learning algorithm is used to solve the optimal resource scheduling strategy under a single edge node. The effectiveness of the proposed resource optimization allocation method for industrial internet fusing edge computing and blockchain is verified by simulation validation. The method is able to obtain better and smoother convergence under the premise of harvesting high total rewards, effectively reduces the computational energy consumption and computational overhead of the device, and at the same time effectively improves the consensus efficiency of the blockchain.

Keywords: Blockchain; Deep reinforcement learning; Mobile edge computing; Resource allocation optimization; Industrial Internet

1. Introduction

The development of social economy and information technology has introduced Internet technology to various industries, and the industrial field is also making more and more progress relying on Internet technology [1]. For enterprises with production and operation characteristics such as a large number of materials, diverse product types, and a large amount of data information, it is crucial to ensure data stability, security, and timeliness [2-3]. The fusion of edge computing and blockchain will inject a new impetus, and in-depth study of its fusion configuration has become the next key direction for enterprises [4].

✉ Corresponding author.
E-mail address: tanxinlong@inspur.com (X. Tan).
Received 05 March 2024; Revised 20 May 2024; Accepted 05 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-451

Currently, the rapid development of industrial Internet attracts extensive attention from the industrial and academic communities. The application of industrial Internet technology can realize the efficient and convenient interaction between machines and machines, and people and machines [5-6]. Multiple types of industrial Internet devices, also known as machine-type communication devices, are widely used in a variety of industrial fields, including environmental monitoring, manufacturing, video surveillance, and intelligent networking [7-8]. It is expected that by 2026, nearly 25 billion number of industrial internet devices will be connected to the network for collecting and transmitting industrial data [9].

Edge computing and cloud computing belong to a complementary relationship. Cloud computing has access to the historical data of edge computing and the advantage of edge computing is that it can process real-time data faster, while cloud computing is more suitable for processing large amounts of data [10-11]. By placing some of the data processing and storage tasks at the edge, the cloud data processing pressure can be greatly reduced and a more efficient data processing process can be realized [12-13].

In industrial scenarios, the security and reliability of industrial data are crucial [14]. However, due to the extremely wide distribution of MTCDs as well as the variety of interacting data, data privacy and security are often difficult to ensure [15]. On the other hand, there is an obvious shortage of energy and computational resources in the industrial Internet, as most MTCDs operate out of manual control, and in order to prolong the working life of the devices, such devices are only equipped with a limited supply of batteries and light computational resources [16-17]. Therefore, some complex and heavy computational tasks are difficult to be accomplished independently within these devices [18].

In the field of industrial internet, the service architecture that integrates edge computing and blockchain technology has significant advantages to reduce cost, improve efficiency, and enhance security [19]. Edge computing is mainly responsible for processing and analyzing large amounts of data generated at the device side to provide faster network service response [20]. Blockchain, on the other hand, provides a secure, reliable, and tamper-free data exchange environment to make resource management in the industrial internet more efficient and transparent. The edge server plays the role of the brain, playing the role of storing and calculating the data uploaded by each device, and distributing them rationally with certain goals as guidelines [21-23]. The end devices store the blockchain data to the edge server, which can save their own storage space and improve the processing efficiency of the edge server to a certain extent. In such an architectural model, edge computing and blockchain devices have a clear division of labor and collaborate with each other [24-25]. Edge computing devices are mainly responsible for consensus reaching tasks, while mobile terminals are treated as blockchain nodes responsible for transaction listening and data synchronization [26].

In this paper, we propose a method for optimal allocation of industrial internet resources that incorporates edge computing and blockchain algorithms. With the objective of minimizing the computational energy consumption and computational overhead of the system, the optimization problem is modeled as a Markov decision process, taking into account the high dynamics and continuity of the environment faced by the system. On this basis, an intelligent cooperative optimization scheduling model for IoT computing and caching resources for cloud edge is constructed, and a deep reinforcement learning algorithm is used to solve the optimization problem and find the optimal optimization strategy. In addition, a CDRL method is proposed in multi-edge nodes, with which the computational resource saving of multi-intelligent body repetitive model is realized. The energy efficiency of the proposed scheme in this paper is demonstrated through simulation experiments.

2. Blockchain Consensus Theory and Its Industrial Internet Applications

2.1. Blockchain Consensus Algorithm

Blockchain technology [27] is quite popular in today's digital economy megatrend. Blockchain is essentially a tamper-proof, decentralized distributed database. The innovativeness of blockchain mainly lies in its integration of key technologies such as hash cipher and asymmetric encryption, peer-to-peer communication, consensus mechanism, digital signature, smart contract, etc. The cryptographic method guarantees the security of the data, the P2P method as well as the distributed storage realizes the decentralization, and maintains the balance of the whole network through the consensus mechanism and the automatic script technology to ensure that the data is trustworthy and reliable. Because of the tamper-proof security characteristics of blockchain, the application fields are mainly financial transactions, identity information management, property tracing, space security, industrial Internet applications and so on.

At present, the most widely used public chain projects, Bitcoin and Ether, use the POW mechanism as the consensus algorithm for the whole network, while the consensus algorithm used by the commercial blockchain system (EOS) network is DPOS.

Most of the current blockchain systems are based on either public or federated chain implementations, and one of the significant differences between public and federated chains is the different consensus algorithms used by the two. Most of the former use POW and POS consensus algorithms to facilitate network-wide consensus, and POW determines the block nodes by calculating the value of nonce, and utilizes arithmetic competition to resist 51% attack and Sybil attack. However, POW causes the network to face problems such as high energy waste, long block confirmation time, and susceptibility to forks due to the nature of arithmetic competition. POS, on the other hand, votes on the basis of the percentage of equity owned by the nodes in the blockchain network, thus electing the next node out of the block with a high number of votes. Despite the low resource consumption of POS, there are problems such as token accumulation and 51% attack in the network. The PBFT consensus algorithm is mainly used in the coalition chain network. Compared with POW and POS consensus, the PBFT mechanism does not require high-performance equipment support, does not generate forks, and has a significant improvement in consensus efficiency.

In the traditional PBFT consensus mechanism, it includes five stages: request, pre-preparation, preparation, confirmation, and reply.

Among the improved consensus optimized based on the PBFT mechanism, Zyzzyva is the most streamlined. It assumes that there are no Byzantine nodes in the network, i.e., all nodes will not have the behavior of maliciously forging messages, and they only need to return their own confirmation messages to the master node.

However, the Zyzzyva mechanism does not support multitasking parallelism. Inspired by Zyzzyva, in order to make the PBFT mechanism to take advantage of its strengths and avoid its weaknesses, this paper still follows its message confirmation mechanism, but reduces its internal pre-preparation, preparation, and confirmation of the three-step message confirmation to a single view, and at the same time, restricts the broadcast range and reduces the complexity of the internal communication in order to realize the purpose of efficiently reaching a consensus, and to fully satisfy the needs of the collaboration of cross-chain transactions in the blockchain-enabled industrial Internet scenario.

2.2. Edge computing

With the development of the big data era, the demand for data interaction is growing explosively, and cloud centers are overburdened with problems such as high energy consumption, high latency, privacy leakage, and oversupply of bandwidth. Edge computing, which is quite popular among researchers in recent years, has emerged to solve its challenges.

Task offloading in edge computing can be generally categorized into independent task offloading and dependent task offloading. Independent task offloading refers to the fact that each task is independent of each other, and the node executing the task only needs to complete the assigned task. Whether it is independent or dependent tasks, task scheduling for such NP-Hard type puzzles is generally solved using classical heuristic algorithms, such as the greedy algorithm [28] or game theory for non-convex problem transformation.

2.3. *Industrial Internet and Blockchain*

Industrial Internet combined with blockchain technology can achieve decentralization, each node has a copy of the data, can independently process real-time data, no longer rely on the central network, to achieve efficient and rapid big data management, also fits the characteristics of industrial Internet for automation and intelligent processing. It also supports decentralized user identity management and decentralized access control, greatly enhancing security and privacy. Blockchain-enabled Industrial Internet is undoubtedly appropriate for smart home and smart medical applications that require high privacy protection and security, such as smart phones, personal computers, smart air conditioners, floor sweeping robots, etc. By uploading sensory data to the blockchain for storage to ensure the security of the data, thus maintaining personal privacy.

The beauty of the blockchain is that as a distributed ledger, its decentralization, security and scalability are known as the “impossible triangle” and cannot be balanced. And the industrial Internet is just right for its scalability to provide a landing scenario. Although industrial Internet devices are always generating data, the data volume of individual devices is small and fragmented, and they are often more than just a single device to independently carry out task processing, usually requiring multiple devices to frequently exchange data, comprehensively measure the actual situation, and carry out real-time decision-making and processing.

Given that the POW consensus mechanism needs to solve the mathematical problems in cryptography, i.e., hash random numbers, the requirements for the equipment's “mining” ability are extremely high, even if it is equipped with a powerful miner in the Bitcoin system mining, it will take ten minutes for a block to come out. So, in the blockchain-enabled industrial internet scenario, if the edge equipment with limited computing power adopts POW to reach consensus, it is bound to take a longer time. Considering the large number of nodes in the industrial Internet, the dispersion of real-time sensed data requires joint decision-making response by multiple devices, and the processing efficiency is even slower and slower. Analyzed again from the network load perspective, POW single-chain network for node storage performance requirements are extremely high, which is still a major problem for industrial Internet nodes.

PBFT does not require expensive computational costs, and can decide whether a new block is legal or not by determining whether the number of message confirmations reaches a specific threshold, which is obviously more suitable for industrial Internet application scenarios that require frequent interactions to meet the performance requirements for reaching consensus quickly. However, the internal communication overhead of a complete PBFT process is extremely high, and the communication delay of a blockchain network based on the PBFT consensus mechanism is already extremely high when 100 nodes are accommodated, which is only applicable to small networks, and the scalability is not enough to meet the needs of edge equipment scenarios with many nodes. As for the POS proof-of-stake consensus mechanism, it relies on the nodes themselves to obtain benefits, and its subjectivity and selfishness are not applicable to the network system composed of objective third-party devices.

2.4. *Service Architecture for the Convergence of Edge Computing and Blockchain*

Edge computing and cloud computing are in a complementary relationship. Cloud computing can access the historical data of edge computing, and the advantage of edge computing is that it can process real-time data faster, while cloud computing is more suitable for processing large amounts of data. By placing some of the data processing and storage tasks at the edge, the cloud data processing pressure can be greatly reduced and a more efficient data processing process can be realized.

In the field of industrial Internet, the service architecture that integrates edge computing and blockchain technology has significant advantages, which can reduce costs, improve efficiency, and enhance security. Edge computing is mainly responsible for processing and analyzing large amounts of data generated at the device side to provide faster network service response. Blockchain, on the other hand, provides a secure, reliable, and tamper-free data exchange environment to make resource management in the Industrial Internet more efficient and transparent. Edge servers play the role of the brain, playing the role of storing and calculating the data uploaded by each device and distributing it appropriately with certain goals in mind. End devices store blockchain data to the edge server, which can not only save their own storage space, but also improve the processing efficiency of the edge server to a certain extent. In such an architectural model, edge computing and blockchain devices have a clear division of labor and collaborate with each other. Edge computing devices are mainly responsible for consensus reaching tasks. The mobile terminal is treated as a blockchain node, responsible for transaction listening and data synchronization, as shown in Figure 1.

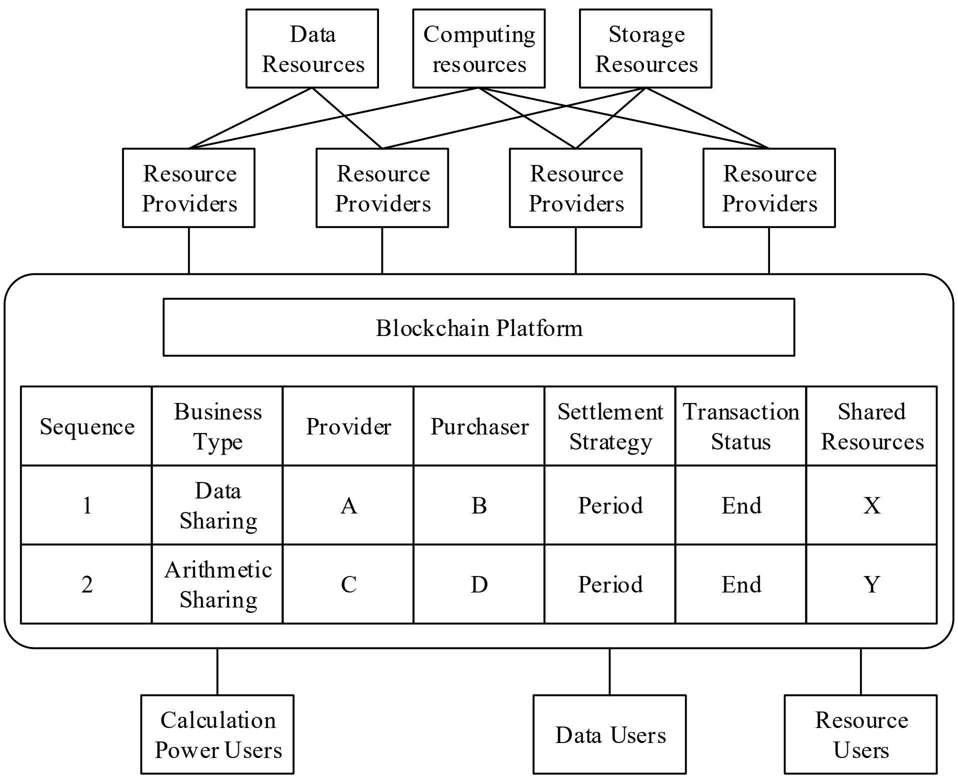

Fig. 1. Edge calculation and blockchain fusion structure

3. Intelligent cooperative optimization scheduling model for resources based on reinforcement learning

3.1. Enhanced learning

Reinforcement learning in which the intelligent body makes a judgment based on the current state and environment it is currently in and makes the next action that makes the reward greater, the action leads to a change in the state and environment and the intelligent body continues to make new actions based on the new state and environment, which ultimately maximizes the reward. Reinforcement learning has a wide range of applications in edge computing research, edge computing requires the allocation and scheduling of a large number of devices and resources, which can be intelligently managed through reinforcement learning.

The reinforcement learning training process is usually modeled as a Markov decision process, which is mathematically described as $\langle \mathrm{S}, \mathrm{A}, \mathrm{P}, \mathrm{R}, \gamma \rangle$, where S represents the set of states; A represents the set of actions of the intelligent body; $\mathrm{P}: \mathrm{S} \times \mathrm{A}$ represents the state transfer probability, i.e., the probability that the intelligent body transfers to state s' after choosing action a in state s; R represents the reward function, i.e., the reward that the intelligent body receives after choosing action a in state s; and γ represents the discount factor, which denotes the the importance weight of the future reward to the current one. Assuming that the probability of moving to the next state s' after choosing action a under the current state s is only related to state s, the state transfer probability P can be expressed as:

$$P_{ss'}^{a} = \mathrm{E}\left(S_{t+1} = s' \mid S_t = s, A_t = a\right) \tag{1}$$

The goal of reinforcement learning is to learn a strategy π that represents a generalization lesson for choosing action a in state s, denoted as:

$$\pi(a \mid s) = P\left(A_t = a \mid S_t = s\right) \tag{2}$$

Using a given strategy, T round of training leads to a long-term cumulative reward G_t:

$$G_t = \sum_{t=1}^{T} \gamma^{t-1} r_t \tag{3}$$

In order to evaluate and select strategies, reinforcement learning introduces the notion of a value function $V_\pi(s)$, representing the expected value of the reward obtained by an intelligent body after adopting a certain determined strategy π, denoted as:

$$V_\pi(s) = \mathrm{E}_\pi\left(G_t \mid S_t = s\right) = \mathrm{E}_\pi\left(\sum_{t=1}^{T} \gamma^{t-1} r_t \mid S_t = s\right) \tag{4}$$

Accordingly, to represent the expected reward for each action, reinforcement learning introduces the $Q_\pi(s,a)$ function, representing the expected value of the reward for choosing action a at state s under strategy π, denoted as:

$$Q_\pi(s,a) = \mathrm{E}_\pi\left(G_t \mid S_t = s, A_t = a\right) = \mathrm{E}_\pi\left(\sum_{t=1}^{T} r^{t-1} r_t \mid S_t = s, A_t = a\right) \tag{5}$$

According to the Bellman equation, the recursive form of the value function and the $Q_\pi(s,a)$-function can be obtained:

$$V_\pi(s) = \mathrm{E}\left(G_t \mid S_t = s\right) = \mathrm{E}_\pi\left(r_{t+1} + \gamma V_\pi\left(S_{t+1}\right) \mid S_t = s\right) \tag{6}$$

$$Q_\pi(s,a) = \mathrm{E}_\pi\left(r_{t+1} + \gamma Q\left(S_{t+1}, A_{t+1}\right) \mid S_t = s, A_t = a\right) \tag{7}$$

Based on the selected strategy and the $Q_\pi(s,a)$ function corresponding to each action, the transformation relation between $Q_\pi(s,a)$ and $V_\pi(s)$ can be derived as:

$$V_\pi(s) = \sum_{a \in A} \pi(a \mid s) Q_\pi(s,a) \tag{8}$$

$$Q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi\left(s'\right) \tag{9}$$

Finally, the transformation relation in the above equation can be brought into the recursive equation to obtain the iterative update formula for the value function and the $Q_\pi(s,a)$-function as:

$$V_{\pi}(s)=\sum_{a\in A}\pi(a\mid s)\left(R_{s}^{a}+\gamma\sum_{s'\in S}P_{ss'}^{a}V_{\pi}\left(s'\right)\right) \tag{10}$$

$$Q_{\pi}(s,a)=R_{s}^{a}+\gamma\sum_{s'\in S'}P_{ss'}^{a}\sum_{a'\in A}\pi\left(a'\mid s'\right)Q_{\pi}\left(s',a'\right) \tag{11}$$

The goal of reinforcement learning is to obtain the optimal policy π^{*} through training and learning, which is generally done by finding the optimal value function, defined as the maximum value of the value function among all the generated policies, i.e:

$$V^{*}(s)=\max_{a}\left(R_{s}^{a}+\gamma\sum_{s'\in s}P_{ss'}^{a}V^{*}\left(s'\right)\right) \tag{12}$$

Similarly the maximum value of the $Q_{\pi}(s,a)$-function can be obtained, i.e:

$$Q^{*}(s,a)=R_{s}^{a}+\gamma\sum_{s'\in S}P_{ss'}^{a}\max_{a'}Q^{*}\left(s',a'\right) \tag{13}$$

This can be obtained from the relationship between the two:

$$V^{*}(s)=\max_{a}Q^{*}(s,a) \tag{14}$$

According to the definition of $Q_{\pi}(s,a)$ function, in state s, the action with the largest $Q_{\pi}(s,a)$ function value is the optimal action, so the optimal policy can be obtained by maximizing the $Q^{*}(s,a)$ function:

$$\pi^{*}(a\mid s)=\begin{cases}1, & if a=\arg\max_{a\in A}Q^{*}(s,a)\\ 0, & otherwise\end{cases} \tag{15}$$

3.2. Deep reinforcement learning

Deep reinforcement learning is a new research hotspot in the field of artificial intelligence, which combines the perceptual ability of deep learning with the decision-making ability of reinforcement learning, and uses deep neural networks as the function approximator of reinforcement learning to approximate the estimation value function and the policy function, which possesses a better fitting effect, and breaks through the limitations of the state space and the action space which are discrete and limited. Utilizing the nonlinear capability of neural networks, deep reinforcement learning can learn more complex policies and has achieved impressive results, which have been applied by scholars in complex and variable computational offloading domains. The following section provides an overview of well-known deep reinforcement learning algorithms.

Combining convolutional neural networks with Q learning algorithms in traditional reinforcement learning, the deep Q network (DQN) algorithm was proposed [29], which is the originator of deep reinforcement learning, solving the problem that Q learning can only deal with a discrete state space, and adopting neural networks to represent continuous states. Compared with traditional Q learning, DQN has the following two main important improvements:

(1) Experience playback pool: construct a buffer to store previous experience, store the combination of states, actions, rewards, and next states in the training process in the experience playback pool, and the intelligent body randomly samples from the experience pool to get new actions, which improves the training efficiency.

(2) Separate target network: introduce an independent target network that is slower than the current neural network, which will estimate the Q-value of each discrete action, and adopt a greedy strategy to select the action with the largest Q-value to execute, and the training process is more stable.

DQN can better handle the continuous state space discrete action space case, but is less capable of handling continuous action space. Although the step can be subdivided into more discrete actions for training, the cost and effect are not ideal, which led scholars to think of determining the policy gradient algorithm.

The policy gradient (PG) algorithm is a more straightforward approach that sets the policy as a function of the actions obtained from the state and lets the neural network fit the policy function, i.e., what action should be performed in the current state. In the non-deterministic policy approach, for the discrete action space, the network outputs the probability values of executing various actions in the current state, and for the continuous action space, a stochastic Gaussian policy is generally used, and the network outputs the mean and standard deviation of the Gaussian policy, and from this, samples the corresponding actions.

The deterministic policy gradient (DPG) algorithm proposes a deterministic, offline policy from a theoretical point of view, and gives the policy gradient formula and parameter update method. Compared to stochastic functions, representing a strategy with a deterministic function has its advantages and disadvantages: the advantage is that it can be theoretically proved that the gradient of a deterministic strategy is the expectation of the gradient of a Q-function, which is computationally more efficient; the disadvantage is that, for each state, the chosen action is deterministic, leading to the inability of the intelligent body to explore it. To address this problem, DPG uses an offline policy approach, where the sampled policies are in stochastic form and the policies to be optimized are in deterministic form, fully guaranteeing exploration.

The Deep Deterministic Policy Gradient (DDPG) algorithm combines DPG and DQN, and proposes the Soft Actor Critic (SAC) algorithm, which introduces maximized entropy to ensure the stability and exploration ability of the algorithm in contrast to traditional deep reinforcement learning algorithms.The SAC algorithm optimizes the policies to maximize the expected entropy of the action distribution, so as to make the distribution of the actions in the output of the policies more even, and also to allow for more uniform distribution of the actions in the training The SAC algorithm learns more approximate optimization behaviors during the training process, i.e., in some states, there may be more than one optimal action, so that the probability of choosing them is the same, which can improve the efficiency of learning, and at the same time encourages the intelligent body to increase the exploration efforts and helps to avoid falling into the local optimum, which is expressed by the formula of the strategy as follows:

$$\pi^* = \arg\max_{\pi} \sum_{t} E_{(s_t,a_t)\sim\rho_\pi}\left[r\left(s_t,a_t\right)+\alpha H\left(\pi\left(\cdot\mid s_t\right)\right)\right] \tag{16}$$

where $H\left(\pi\left(\cdot\mid s_t\right)\right)=E_{a_t}\left[-\log\pi\left(a_t\mid s_t\right)\right]$ is the strategy entropy and α denotes the temperature parameter, which represents the entropy and the weights before the system rewards.

3.3. Intelligent Collaboration Industrial Internet Resource Intelligent Allocation Model Construction

3.3.1. Network modeling. In this section, an IoT system is designed based on the cloud-edge-end architecture and co-enabled by edge nodes and blockchain with caching and computing capabilities. It is also considered in this system that due to the weak computational capability of the VR devices at the device layer, the computational tasks at the device layer can be offloaded to the edge nodes through the wireless communication network.

3.3.2. Blockchain model. Let $V_l^{tru} \in [0,1]$ represent the trust value of the blockchain node l, and that only nodes with a trust value of more than 0.5 can be trusted as candidates to participate in the blockchain consensus.

Specifically, the direct trust value V_l^{dir} can be given by subjective logic, which is calculated as follows:

$$V_l^{dir} = \begin{cases} \left(HO_l - 0.5\right) \times \varepsilon(t) + 0.5, \ If \ HO_l \geq 0.5 \\ HO_l \times \varepsilon(t), \ Other \end{cases} \tag{17}$$

For the indirect trust value V_l^{ind} of a blockchain node, the number of times this node has participated in consensus in the past is considered as an important metric. Therefore, the indirect trust value can be obtained as follows:

$$V_l^{ind} = \frac{CP_l}{R} \tag{18}$$

Combining the direct and indirect trust values, the individual blockchain node trust value expression can be obtained as follows:

$$V_l^{tru} = w_{dir} \cdot \frac{\sum V_l^{dir}}{L} + w_{\text{ind}} \cdot V_l^{\text{ind}} \tag{19}$$

In the formula, w_{dir} and w_{ind} represent the weights of direct and indirect trust values, respectively, which satisfy $w_{dir} + w_{ind} = 1$.

In addition, the dBFT consensus protocol ensures the effectiveness of the blockchain system when the malicious nodes are less than $R/3$. The process of the dBFT protocol is mainly divided into the following three stages:

1) Pre-preparation. Under this process, the CPU computing cycles to be consumed can be calculated as follows:

$$c_{pp}(t) = \underbrace{\left(1 + (1+\lambda) \cdot \frac{g(t)}{o(t)}\right) \cdot (\theta + \delta)}_{\text{Validation}} + \underbrace{\theta + (R-1) \cdot \delta}_{\text{Generate}} \tag{20}$$

where λ represents the probability of successful block checksum, $g(t)$ is defined as the transaction batch size at time slot t, which depends on the block size. $o(t)$ represents the average size of transactions.

2) Preparation. In the preparation phase, all the slave nodes will generate $R-1$ MACs and a signature, which is then, sent to all the nodes participating in the consensus for verification. Therefore, for this phase, the required CPU computation cycles can be expressed as:

$$c_{pr}(t) = \underbrace{4f \cdot (\theta + \delta)}_{\text{Validation}} + \underbrace{\theta + (R-1) \cdot \delta}_{\text{Generate}} \tag{21}$$

3) Submit. In the submit phase, the nodes that passed the previous validation phase will generate $R-1$ MAC and a signature and send it to other consensus nodes for validation. If the number of nodes that failed the verification is less than f, the consensus process will be determined as complete. Thus, for the commit phase, the required CPU computation cycle can be expressed as:

$$c_{co}(t) = \underbrace{2f \cdot (\theta + \delta)}_{\text{Validation}} + \underbrace{\theta + (R-1) \cdot \delta}_{\text{Generate}} \tag{22}$$

Based on the above analysis, the CPU computational resources required for the entire consensus process are denoted as:

$$\begin{aligned} c_{tot}(t) = & \left[(1+\lambda) \cdot \frac{g(t)}{o(t)} + 6f + 5 \right] \cdot \theta \\ & + \left[(1+\lambda) \cdot \frac{g(t)}{o(t)} + 6f + 3m - 1 \right] \cdot \delta \end{aligned} \tag{23}$$

In this section, it is assumed that THz communication is used for wireless communication between cloud servers and edge nodes. The derived data rate of THz communication link channel is expressed as:

$$R_{e,c}(t) = B_{e,c} \log_2\left(1+\gamma_{e,c}{}^2\right) \tag{24}$$

where $B_{e,c}$ represents the communication bandwidth between the edge node and the cloud server, and γ_{ec} represents the signal-to-noise ratio of the terahertz communication channel.

If the AI intelligences deployed at the edge node choose to offload the consensus task from the edge node to the cloud server, the processing delay and energy consumption caused by the consensus mainly includes the transmission delay and energy consumption as well as the cloud server computation delay and energy consumption, which is denoted as:

$$t_c(t) = \underbrace{\frac{g(t)}{R_{e,c}(t)}}_{\text{Transmission}} + \underbrace{\frac{c_{tot}(t)}{F_c}}_{\text{Computing}} \tag{25}$$

and:

$$e_c(t) = \underbrace{p_k \frac{g(t)}{R_{e,c}(t)}}_{\text{Transmission}} + \underbrace{p_c \frac{c_{tot}(t)}{F_c}}_{\text{Computing}} \tag{26}$$

Where, p_k represents the transmission power of the edge node, p_c represents the computational power of the cloud server, and F_c represents the CPU computation frequency of the edge node.

If the intelligence chooses the edge node to perform the consensus task, the processing delay and energy consumption caused by consensus is given by the following equation:

$$t_b(t) = \frac{c_{tot}(t)}{F_m} \tag{27}$$

and:

$$e_b(t) = p_m \frac{c_{tot}(t)}{F_m} \tag{28}$$

where p_m represents the computational power of the edge node and F_c represents the CPU computational frequency of the edge node.

In summary, the computational overhead and energy consumption of the whole consensus process can be calculated as:

$$M_b = a_b(t)\cdot\left(t_b \cdot \vartheta(t)\right) + \left(1-a_b(t)\right)\cdot\left(t_c \cdot \varpi(t)\right) \tag{29}$$

and:

$$E_b = a_b(t)\cdot e_b + \left(1-a_b(t)\right)\cdot e_c \tag{30}$$

3.3.3. Cache model. The data rate of the wireless channel between the device layer and the edge layer is expressed as follows:

$$R_{e,m}(t) = B_{e,m} \log_2\left(1+\gamma_{e,m}{}^2\right) \tag{31}$$

Therefore, in the case of this cache, the whole process energy consumption occurs mainly in data transmission, which can be expressed by the following equation:

$$e_{c1} = p_e \cdot \frac{G}{R_{e,m}(t)} \tag{32}$$

where p_e represents the transmission power of the VR device.

The energy consumption is related to the cache popularity ∂_e, which can be given by Eq:

$$e_{c2} = p_e \cdot \frac{G}{R_{e,m}(t)} + \left(1-\partial_e\right) \cdot p_m \cdot \frac{c_m(t)}{F_m} \tag{33}$$

where $c_m(t)$ represents the CPU computational resources consumed for performing the computational task in time slot t.

In summary, the energy consumption of this caching process at time slot t can be defined as follows:

$$E_c = a_c(t) \cdot e_{c1} + \left(1 - a_c(t)\right) \cdot e_{c2} \tag{34}$$

where $a_c(t) = \{0,1\}$ indicates the caching decision of the computational task of the VR device, $a_c(t) = 1$ indicates that the edge node cached the requested task, otherwise, the edge node has not yet cached the requested task.

3.3.4. Computational models. Assume that the computational power of the VR device is c_u. Therefore, if the VR device chooses to process the computational task locally, the energy consumption at time slot t can be expressed as follows:

$$e_m = \mu \cdot \left(c_u\right)^2 \cdot c_m(t) \tag{35}$$

where μ is set to 10^{-27}.

If the VR device offloads the computational tasks to the edge node, the resulting energy consumption mainly consists of data transmission and computation, which can be expressed as follows:

$$e_v = p_e \cdot \frac{D(t)}{R_{e,m}(t)} + p_m \cdot \frac{c_m(t)}{F_m} \tag{36}$$

where $D(t)$ represents is the size of the data task for time slot t.

Moreover, in this case, the computational time and computational overhead of the edge node processing task can be calculated as:

$$t_v = \frac{c_m(t)}{F_m} \tag{37}$$

and:

$$M_s = \rho(t) + t_v \cdot \vartheta(t) \tag{38}$$

where $\rho(t)$ is the fixed management overhead of the edge node.

To summarize, the energy consumption and computational overhead of the whole process of whether a VR device computing task needs to be offloaded or not can be expressed as follows:

$$E_m = a_m(t) \cdot e_m + \left(1 - a_m(t)\right) \cdot e_v \tag{39}$$

and:

$$M_c = \left(1 - a_m(t)\right) \cdot M_s \tag{40}$$

where $a_m(t) = \{0,1\}$ denotes the offloading decision of the computational task of the VR device, and $a_m(t) = 1$ denotes that the VR device locally processes the requested task, otherwise, offloads the computational task to the edge node.

Based on the above analysis, the total energy consumption and computational overhead of the whole system are denoted as:

$$E_{tot} = E_b + E_c + E_m \tag{41}$$

and:

$$M_{tot} = M_b + M_c \tag{42}$$

3.4. Optimization problem modeling

In this section, the VR device computation task offloading decision, caching decision, and consensus task computation offloading decision within the architecture are modeled as discrete MDPs and are solved iteratively using the CDRL algorithm to obtain the most optimal policy based on the optimization objective of minimizing energy consumption and computation overhead.

Let $\psi(t)$ denote the state space of the environment in time slot t, which is represented as follows:

$$\psi(t) = \{\eta(t), \exists(t), \tau(t), \partial_e(t)\} \tag{43}$$

The probability of automatic transfer to state $\psi(t+1)$ can be expressed by the following equation:

$$P_r(\psi(t+1) \mid \psi(t), a(t)) = \int_{\psi(t+1)} f\left(\psi(t), a(t), s'\right) ds' \tag{44}$$

At time slot t, the action space $a(t)$ is represented as follows:

$$a(t) = \{a_m(t), a_b(t), a_c(t)\} \tag{45}$$

For the IoT framework with cloud-side-end collaboration, this paper intends to optimize the performance of computational overhead and energy consumption in the long run. In this paper, $r(t)$ is defined to represent the immediate reward function, which is defined as follows:

$$r(t) = \begin{cases} \omega_1 \cdot \dfrac{1}{M_{tot}(t)} + \omega_2 \dfrac{1}{E_{tot}(t)} \; If \;\; C1\text{-}C4 \; are \; satisfied \\ \omega_1 \cdot \dfrac{1}{M_{tot}(t)} + \omega_2 \dfrac{1}{E_{tot}(t)} - v Other \\ s.t. C1: \varepsilon(t) \ge \varepsilon_{\min} \\ C2: G \le J(t) \\ C3: D(t) \le S(t) \\ C4: g(t) \le \varsigma(t) \end{cases} \tag{46}$$

3.5. Intelligent cooperative optimization scheduling method for resources based on CDRL algorithm

In DQN, there is an action state value function $Q(\psi_t, a_t)$ to estimate the value of each step of the action and it is updated by iteration and its iterative expression is shown below:

$$\begin{aligned} Q(\psi_t, a_t) \leftarrow & \, Q(\psi_t, a_t) \\ & + \alpha\left[r(\psi_t, a_t) + \gamma \max Q(\psi_{t+1}, a_{t+1}) - Q(\psi_t, a_t)\right] \end{aligned} \tag{47}$$

where $\gamma \in (0,1)$ represents the discount factor that balances long-term and immediate rewards, and $\alpha \in (0,1)$ represents the learning rate.

The optimization objective contains two components: exploitation and exploration, which can be expressed as:

$$\max \underbrace{r\left(\psi_t, a_t\right)}_{\text{Utilization}} + \underbrace{\varepsilon E\left\{D_{KL}\left(\xi_i \| \; \xi_{g_t}\right)\left[\psi_t, a_t\right]\right\}}_{\text{Explore}} \tag{48}$$

This section introduces the concept of “scaling”, which enables intelligences to actively collaborate with other intelligences, an approach known as CDRL. Therefore, the optimization objective can be expressed as:

$$\begin{aligned} &\max \underbrace{r\left(\psi_t, a_t\right)}_{\text{Utilization}} + \underbrace{\varepsilon E\left\{D_{KL}\left(\xi_i \| \; \xi_{g_\Delta}\right)\left[\psi_t, a_t\right]\right\}}_{\text{Explore}} \\ &-\underbrace{\varepsilon E\left\{D_{KL}\left(\xi_i \| \; \xi_{-i}\right)\left[\psi_t, a_t\right]\right\}}_{\text{Expand}} \end{aligned} \tag{49}$$

In order to fully utilize the training experience of the other intelligent edge nodes, it is considered to fully integrate the other intelligences with the node's own model as the initial model of this intelligent node for training, which can be represented as:

$$\xi_{start} = \frac{\xi_i + \sum_{j=1}^{n} \xi_j}{n+1} \tag{50}$$

4. Resource optimization scheduling model simulation

This section first introduces the environment and parameters of the simulation experiments, followed by showing and discussing the relevant simulation results under different frameworks and different methods. In order to evaluate the proposed method in this paper more intuitively, the framework and algorithms of this paper's method will be simulated and analyzed.

The simulation experiment environment of this paper is Python 3.6 and Tensorflow 1.31.1 environment which is widely used to deploy machine learning algorithms. In the network scenario, the existence of 6 cells corresponding to 6 local controllers is considered, as well as 3 servers used to support the computation of the node consensus process, including 2 MEC servers and 1 cloud computing server. In addition, there are 4 consensus nodes deployed in the blockchain system for data consensus.

Figure 2 shows the convergence comparison of the optimization effect of different methods. In the figure, as the number of iterations increases, the optimization curves of each method reach convergence after about 2400 iterations, and compared with other methods, the optimization of the method proposed in this paper can get higher total rewards, up to 4600. The main reason is that reasonable offloading decisions and appropriate block sizes help to reduce the device energy consumption and computational overheads, and the selection of more efficient and cheaper servers also saves the system overhead.

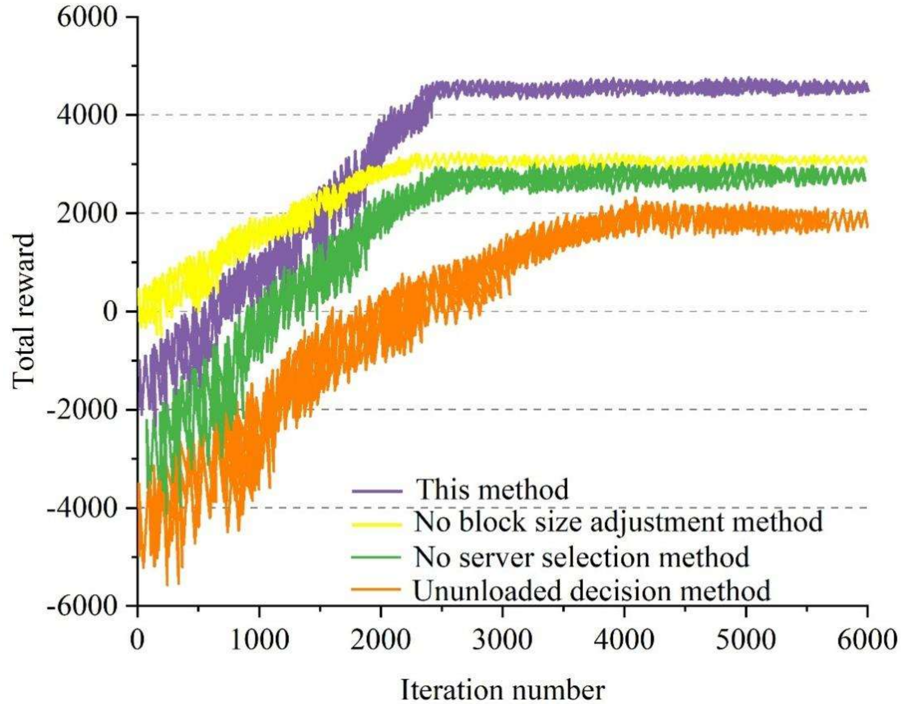

Fig. 2. Contrast of curve convergence under different framework

In addition, Fig. 3 shows a comparison of the convergence of the optimization curves of the CDRL method proposed in this paper with the Q-learning method. As shown in the figure, the proposed method can achieve higher total rewards and has more stable convergence. The main reason for this is that the state dynamics in blockchain-based industrial internet systems are high-dimensional, and it is difficult for the Q-learning method to explore and evaluate all action-state pairs, and its incomplete exploration makes it difficult for the agent to obtain an optimal policy. On the contrary, the proposed CDRL method evaluates the advantages and disadvantages of actions in different states through deep neural networks, which is more adapted to the high-dimensional and complex environment of the system. Therefore, the proposed method can achieve better optimization results.

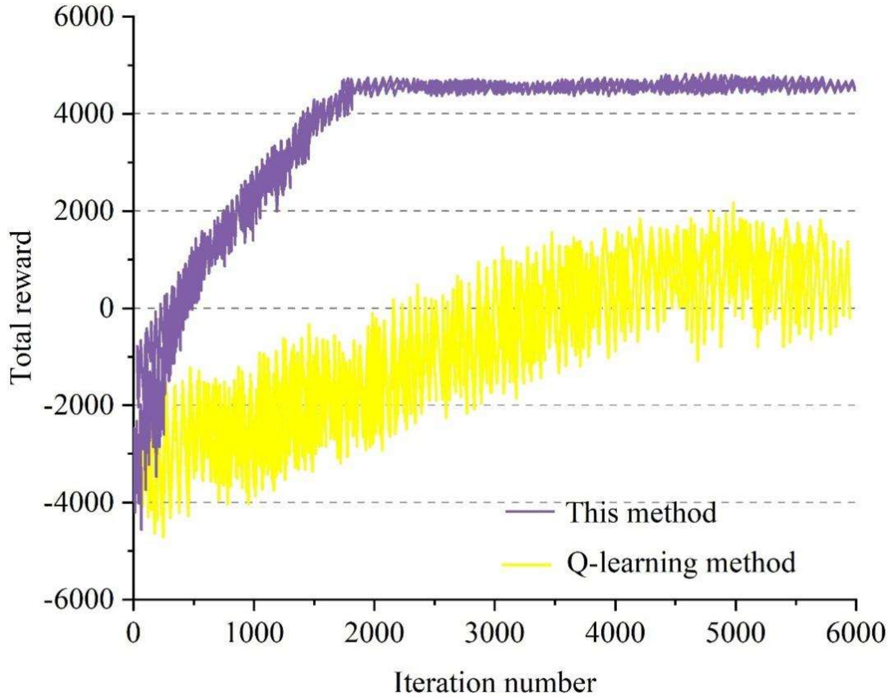

Fig. 3. This method is compared with the convergence of the method curve

Fig. 4 and Fig. 5 show the graphs of device energy consumption versus the number of cells under different frameworks and methods of optimization. As shown in the figure, when the number of cells increases, the device energy consumption also increases. Meanwhile, the equipment energy consumption optimized by the methods proposed in this paper are all significantly lower than other methods when the number of cells is greater than 4. The main reason is that, compared with other methods, the offloading decision of the proposed method is more reasonable, and the decision of whether to offload the computation task after balancing the energy consumption of the device and the computation overhead effectively relieves the energy consumption pressure of the device when it faces the complex computation task.

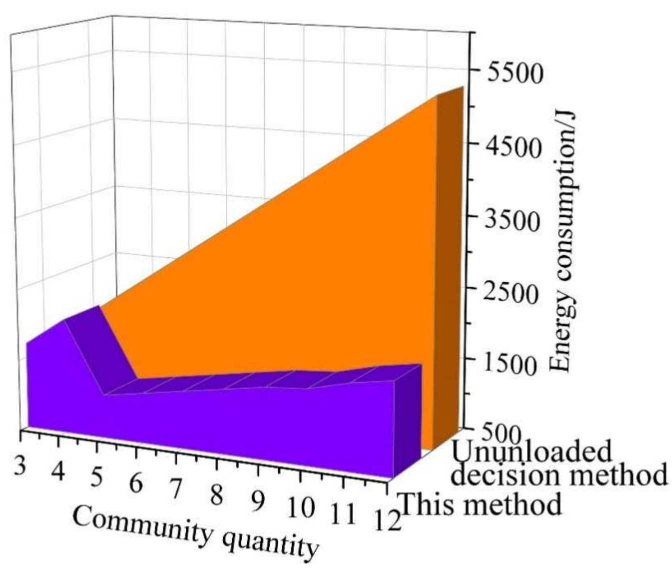

Fig. 4. The amount of energy and community under different frameworks

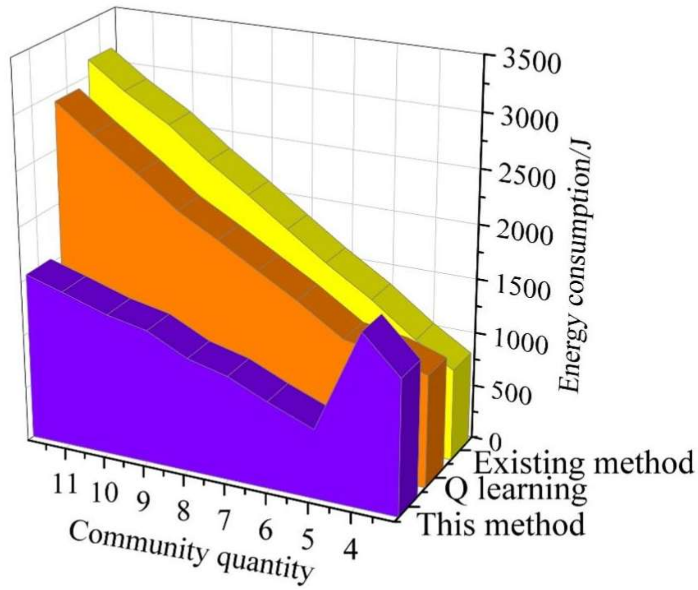

Fig. 5. The amount of energy and community in different ways

Fig. 6 and Fig. 7 show the computational overhead versus the number of cells under different frameworks and methods, respectively. In the figure, the computational overhead increases as the number of cells increases, but the computational overhead under the proposed method in this paper is always lower than other methods. The main reason is that the increased amount of computational tasks occupies more server computational resources, deriving more computational overhead, and larger-sized blocks are generated after adjustment in order to carry more transactions. As a result, the computational overhead increases significantly. However, the method in this paper effectively saves the additional computational overhead by timely adjusting the block size and appropriately selecting cheaper and more efficient servers.

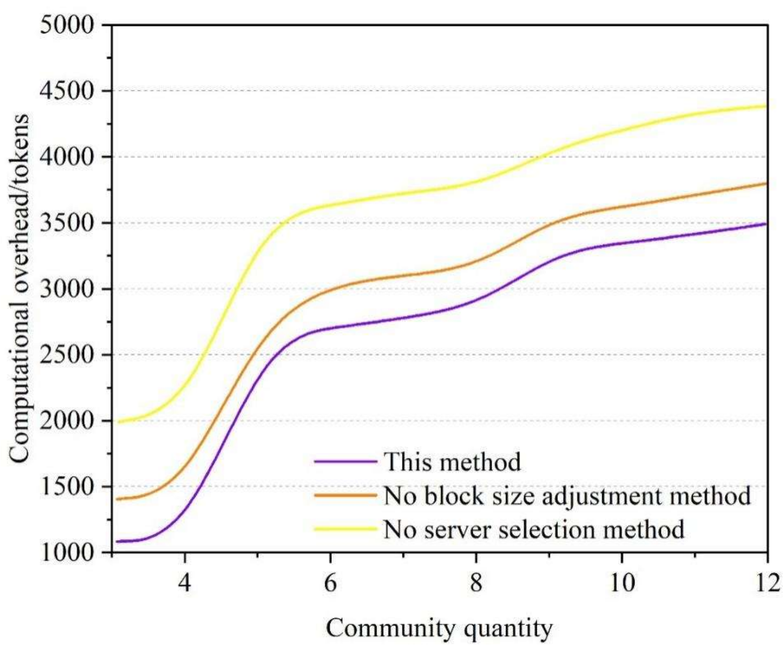

Fig. 6. The calculation cost and the quantity diagram of the neighborhood

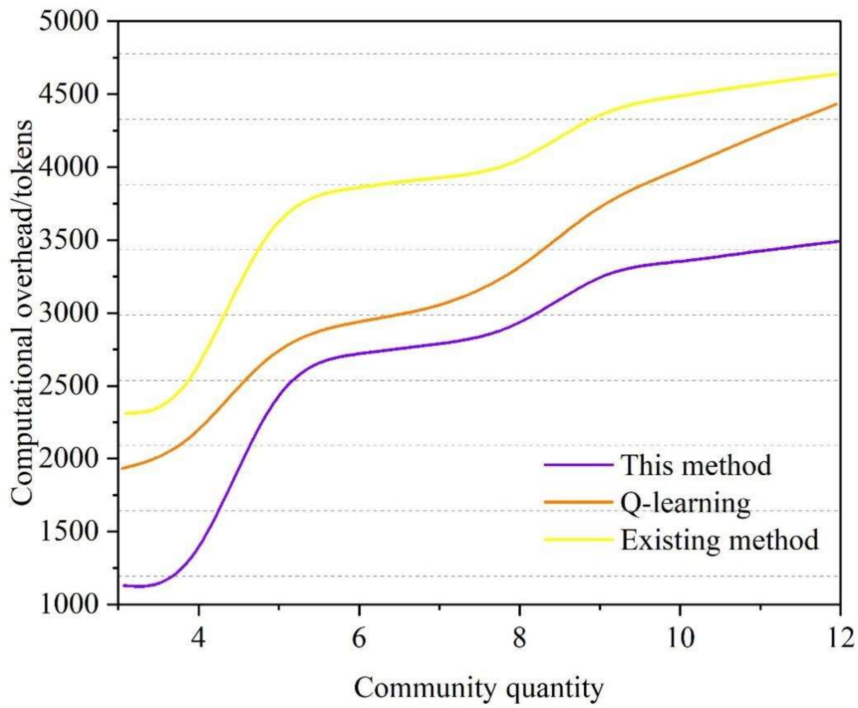

Fig. 7. The calculation cost and the quantity diagram of the neighborhood

Fig. 8 and Fig. 9 show the system energy overhead and versus the number of cells under different frameworks and methods, respectively. The system energy overhead and grows with the number of cells under all methods. The system energy overhead and includes both device energy and computation overhead, and the proposed method can achieve better results compared to the existing methods due to the existence of joint optimization of offloading decision, block sizing and server selection.

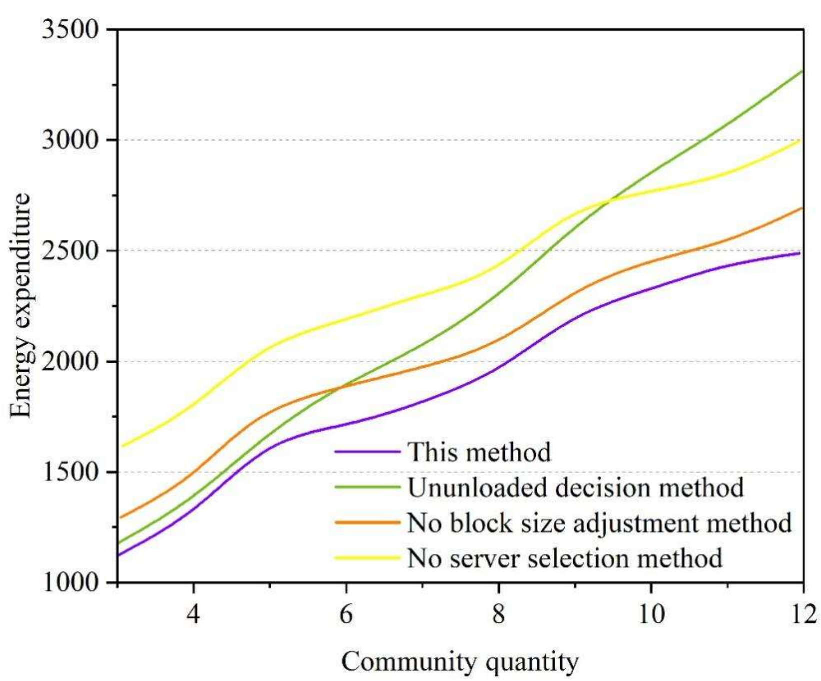

Fig. 8. The amount of energy consumption and the number of communities

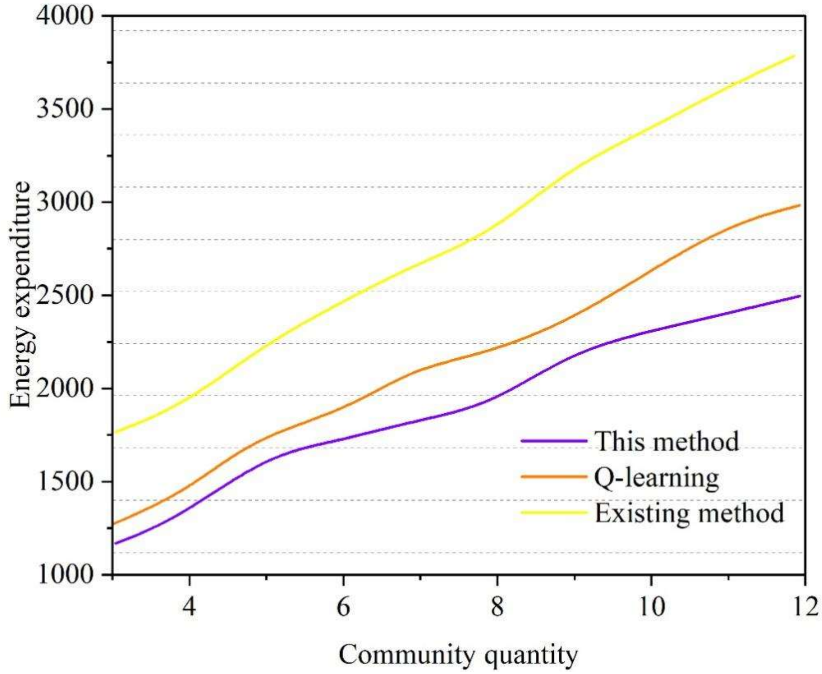

Fig. 9. The amount of energy consumption and the number of communities

Figures 10 and 11 compare the blockchain consensus processing time, with the experimental setup using DAG vs. DPOS consensus algorithms. The results show that the processing time rises steadily as the transaction concurrency and data size increase. However, when the transaction concurrency reaches 50, the processing time fluctuates due to the consensus algorithm performance bottleneck. This indicates that frequent interaction between sharding and blockchain affects the processing time in high concurrency scenarios.

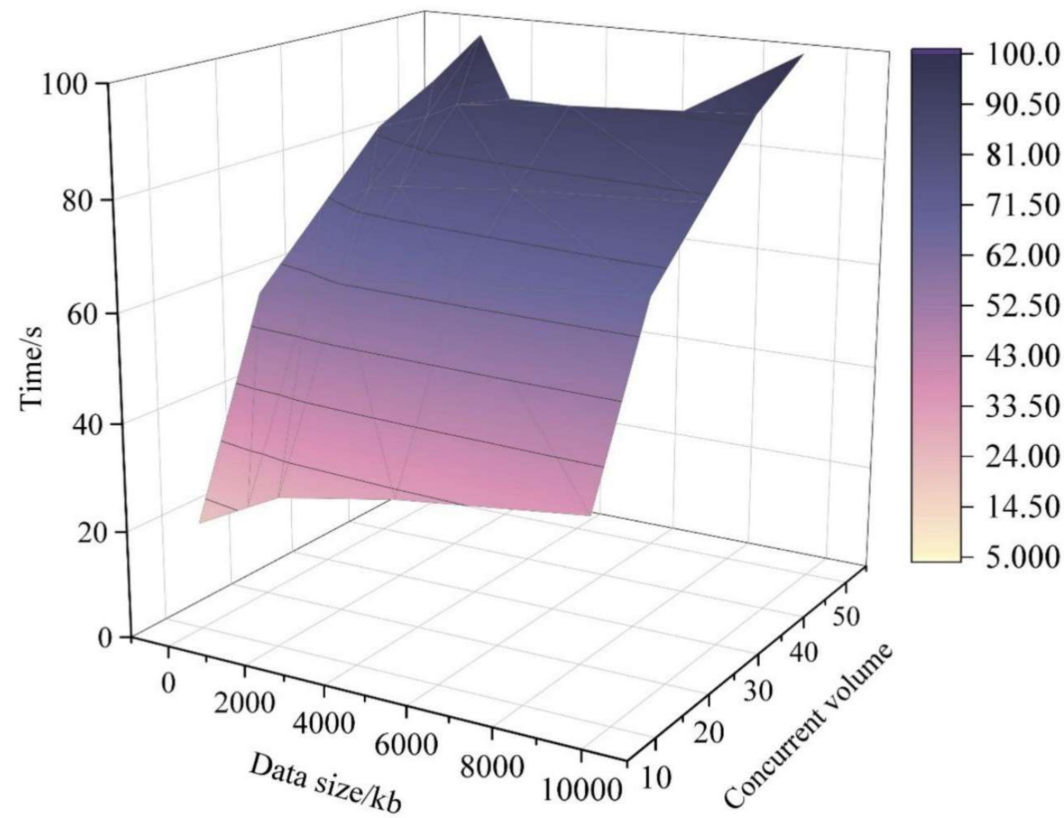

Fig. 10. Processing time of DPOS block chain consensus

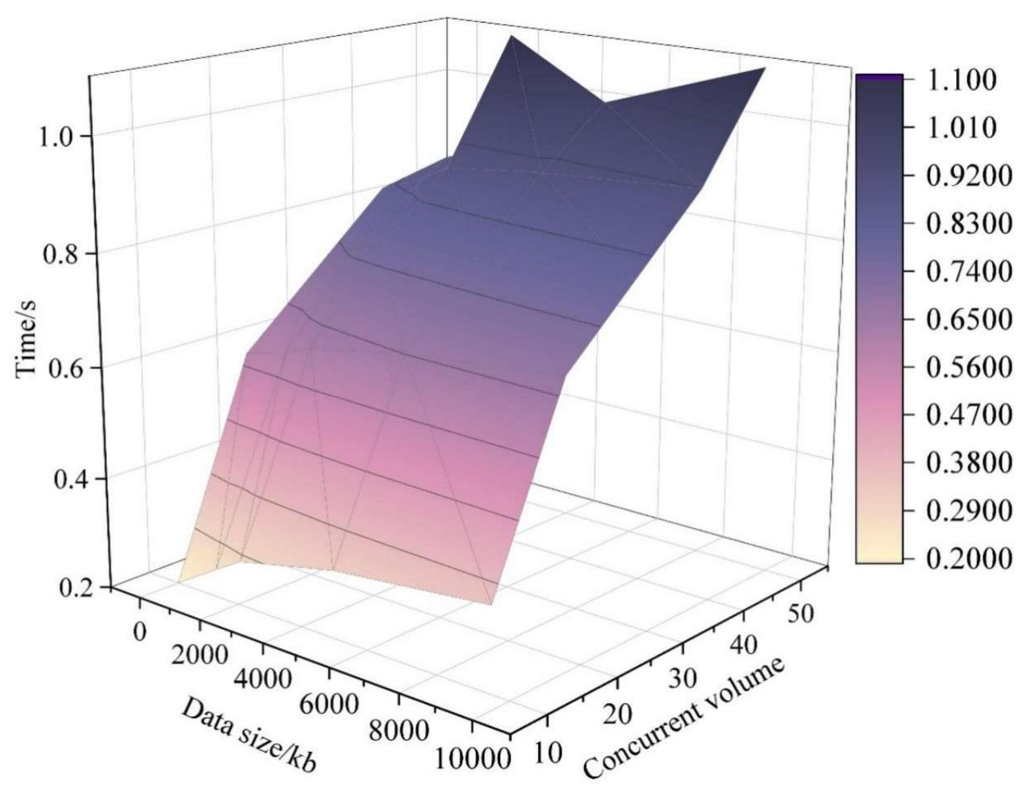

Fig. 11. Processing time of DAG block chain consensus

5. Industrial Internet system performance testing

During the simulation testing phase, work is carried out on packaging industrial edge applications using Docker technology. During testing, containers are created from images and code is tuned, with common parameters including global variables, hardware mapping, etc. In the performance testing phase of the blockchain platform, key factors include the throughput of the blockchain network as well as the average latency. Blockchain throughput is the average number of requests processed by the blockchain network per second, and the average latency is the amount of time per unit spent specifically on processing the demand. The test operations of the blockchain platform include data creation, deletion, lookup, modification, etc., and the execution time of a single round of testing is 100 s. Figures 12 and 13 show the performance graphs of the blockchain network processing lookup requests and the blockchain network processing modification requests, respectively. When the speed of search request submission is less than 400 times/s, the throughput of blockchain network processing modification is basically equal to the speed of request submission, and the average delay is ⩽0.03 s. When the speed of search request submission is greater than 400 times/s, the throughput of blockchain network processing modification is lower than the value of the speed of request submission, and the average delay has increased significantly. That is, the maximum throughput of the lookup processed by the blockchain network is 400 times/s. The maximum throughput of the blockchain for processing modification requests is 120 times/s, and the average latency is about 0.52 s. It is thus concluded that blockchain intervention in the industrial Internet can effectively improve the response rate.

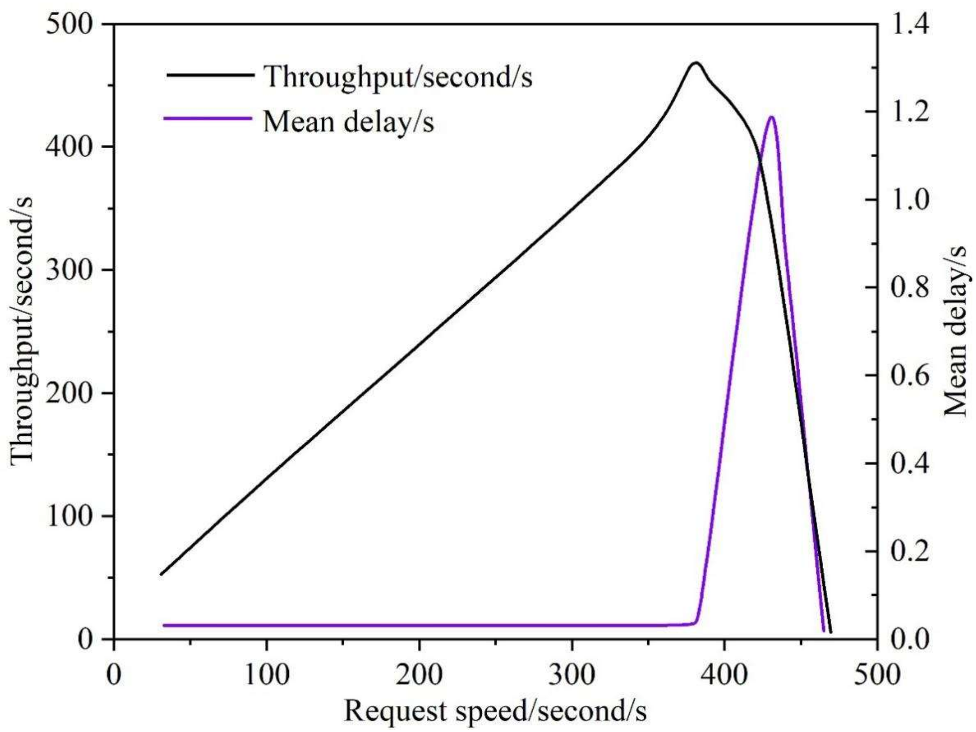

Fig. 12. Block network processing lookup request

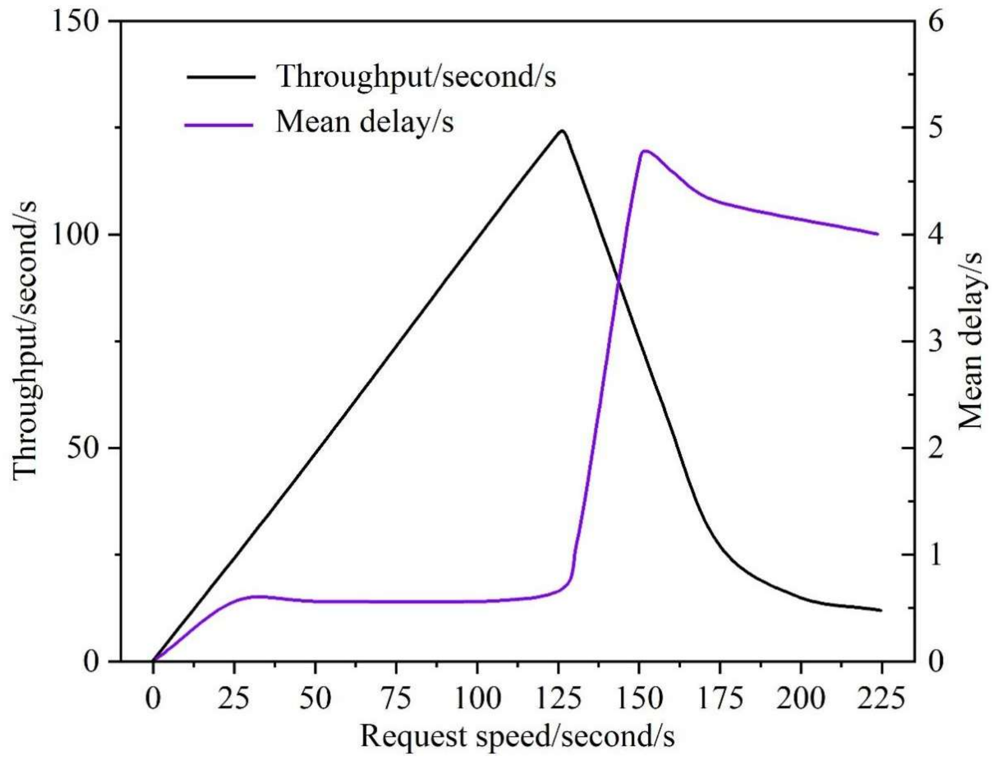

Fig. 13. Block chain network processing modification request

Table 1 shows the test results of request computation time and request response time. In the testing phase, the experiment sets the platform network broadband to 10Mbps, and sets the incoming delay and outgoing delay to 25ms. The request response time of the cloud platform is significantly higher than the response time of the edge nodes, the time consumed by the MQTT message processing in the request computation is almost negligible, and there is a communication delay of about 60ms in the difference of the request response time between the cloud platform and the edge nodes. It is concluded as follows: deploying computers at the edge of the network can effectively shorten the time consumption of computing in industrial scenarios and effectively improve the computational efficiency.

Table 1. Test results for the request operation time

	Request operation time		Request response time	
	Message processing	Image processing	Message processing	Image processing
Marginal node	0.52	12.51	4.12	51.56
Cloud platform	0.48	10.03	13.89	168.97

6. Conclusion

In this paper, for the industrial internet system integrating edge computing and blockchain, an intelligent cooperative optimization scheduling model based on reinforcement learning is proposed to improve the efficiency of blockchain consensus and data processing.

The simulation results prove that the proposed method not only has more stable and fast convergence, but also can obtain higher total rewards, with the highest reward reaching 4,600.In addition, the algorithm in this paper can more effectively reduce the energy consumption of the equipment, save the extra computational overhead, and effectively improve the consensus efficiency of the blockchain.

It is proved by the results of system performance test that when the submission speed of lookup request is less than 400 times/s, the throughput of blockchain network processing modification and the speed of request submission are almost maintained in the equilibrium state of speed equivalence, and the average latency time in this case is less than 0.03 s. When the submission speed of lookup request is greater than 400 times/s, the value of the speed of request submission exceeds the throughput of blockchain network processing modification, and the average delay time increases significantly. Blockchain intervention in industrial internet can effectively improve the system response rate. Deploying computers at the edge can also shorten the computing time and increase the computing speed in industrial scenarios.

References

[1] Liu, X., & Zhang, X. (2019). NOMA-based resource allocation for cluster-based cognitive industrial internet of things. IEEE transactions on industrial informatics, 16(8), 5379-5388.

[2] Qin, W., Chen, S., & Peng, M. (2020). Recent advances in Industrial Internet: insights and challenges. Digital Communications and Networks, 6(1), 1-13.

[3] Liu, X., Zhai, X. B., Lu, W., & Wu, C. (2019). QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA. IEEE Transactions on Industrial Informatics, 17(3), 2052-2061.

[4] Li, X., Li, D., Wan, J., Liu, C., & Imran, M. (2018). Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing. IEEE Internet of Things Journal, 5(3), 1351-1360.

[5] Chen, Y., Liu, Z., Zhang, Y., Wu, Y., Chen, X., & Zhao, L. (2020). Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things. IEEE Transactions on Industrial Informatics, 17(7), 4925-4934.

[6] Abuhasel, K. A., & Khan, M. A. (2020). A secure industrial internet of things (IIoT) framework for resource management in smart manufacturing. IEEE Access, 8, 117354-117364.

[7] Li, S., Ni, Q., Sun, Y., Min, G., & Al-Rubaye, S. (2018). Energy-efficient resource allocation for industrial cyber-physical IoT systems in 5G era. IEEE Transactions on Industrial Informatics, 14(6), 2618-2628.

[8] Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. IEEE transactions on industrial informatics, 14(11), 4724-4734.

[9] Zhou, J., Gao, L., Lu, C., & Yao, X. (2023). Towards multi-task transfer optimization of cloud service collaboration in industrial internet platform. Robotics and Computer-Integrated Manufacturing, 80, 102472.

[10] Liu, Y., Yang, C., Huang, K., Gui, W., & Hu, S. (2022). A systematic procurement supply chain optimization technique based on industrial internet of things and application. IEEE Internet of Things Journal, 10(8), 7272-7292.

[11] Mao, W., Zhao, Z., Chang, Z., Min, G., & Gao, W. (2021). Energy-efficient industrial Internet of Things: Overview and open issues. IEEE transactions on industrial informatics, 17(11), 7225-7237.

[12] Dolui, K., & Datta, S. K. (2017, June). Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In 2017 Global Internet of Things Summit (GIoTS) (pp. 1-6). IEEE.

[13] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal, 7(8), 7457-7469.

[14] Cao, B., Wang, X., Zhang, W., Song, H., & Lv, Z. (2020). A many-objective optimization model of industrial internet of things based on private blockchain. IEEE Network, 34(5), 78-83.

[15] Messaoud, S., Bradai, A., & Moulay, E. (2019). Online GMM clustering and mini-batch gradient descent

based optimization for industrial IoT 4.0. IEEE Transactions on Industrial Informatics, 16(2), 1427-1435.

[16] Zhou, Z., Guo, Y., He, Y., Zhao, X., & Bazzi, W. M. (2019). Access control and resource allocation for M2M communications in industrial automation. IEEE Transactions on Industrial Informatics, 15(5), 3093-3103.

[17] Haghnegahdar, L., Joshi, S. S., & Dahotre, N. B. (2022). From IoT-based cloud manufacturing approach to intelligent additive manufacturing: Industrial Internet of Things—An overview. The International Journal of Advanced Manufacturing Technology, 119(3), 1461-1478.

[18] Liu, M., Yu, F. R., Teng, Y., Leung, V. C., & Song, M. (2019). Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: A deep reinforcement learning approach. IEEE Transactions on Industrial Informatics, 15(6), 3559-3570.

[19] Jain, D. K., Tyagi, S. K. S., Neelakandan, S., Prakash, M., & Natrayan, L. (2021). Metaheuristic optimization-based resource allocation technique for cybertwin-driven 6G on IoE environment. IEEE Transactions on Industrial Informatics, 18(7), 4884-4892.

[20] Xu, H., Yu, W., Griffith, D., & Golmie, N. (2018). A survey on industrial Internet of Things: A cyber-physical systems perspective. Ieee access, 6, 78238-78259.

[21] Sun, L., Wan, L., & Wang, X. (2020). Learning-based resource allocation strategy for industrial IoT in UAV-enabled MEC systems. IEEE Transactions on Industrial Informatics, 17(7), 5031-5040.

[22] Liu, X., Pei, J., Liu, L., Cheng, H., Zhou, M., & Pardalos, P. M. (2017). Optimization and management in manufacturing engineering: Resource collaborative optimization and management through the Internet of Things (Vol. 126). Springer.

[23] Wan, J., Chen, B., Imran, M., Tao, F., Li, D., Liu, C., & Ahmad, S. (2018). Toward dynamic resources management for IoT-based manufacturing. IEEE Communications Magazine, 56(2), 52-59.

[24] Li, G., Wu, J., Li, J., Wang, K., & Ye, T. (2018). Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things. IEEE Transactions on Industrial Informatics, 14(10), 4702-4711.

[25] Yang, H., Alphones, A., Zhong, W. D., Chen, C., & Xie, X. (2019). Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks. IEEE Transactions on Industrial Informatics, 16(8), 5565-5576.

[26] Li, J. Q., Yu, F. R., Deng, G., Luo, C., Ming, Z., & Yan, Q. (2017). Industrial internet: A survey on the enabling technologies, applications, and challenges. IEEE Communications Surveys & Tutorials, 19(3), 1504-1526.

[27] Yu Wantao,Wong Chee Yew,Jacobs Mark & Chavez Roberto. (2025). Modern slavery supply chain capabilities: the effects of Blockchain technology and employees' digital dexterity. International Journal of Operations & Production Management(1),210-235.

[28] Ocampo Anton Louise Pernez De & Dadios Elmer. (2019). Radial greed algorithm with rectified chromaticity for anchorless region proposal applied in aerial surveillance. International Journal of Advances in Intelligent Informatics(3),193-193.

[29] Yibo Lai,Weiyan Zheng,Zhiqing Sun,Yan Zhou & Yuling Chen. (2024). Micro-grid source-load storage energy minimization method based on improved competitive depth Q - network algorithm and digital twinning. Energy Informatics(1),126-126.