https://doi.org/10.1186/s13677-023-00452-4

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

# Blockchain-based collaborative edge computing: efficiency, incentive and trust

Qinghang Gao[1], Jianmao Xiao[1*], Yuanlong Cao[1], Shuiguang Deng[2], Chuying Ouyang[3] and Zhiyong Feng[4]

**Abstract**

The rise of 5G technology has driven the development of edge computing. Computation offloading is the key and challenging point in edge computing, which investigates offloading resource-intensive computing tasks from the user side to the cloud or edge side for processing. More consideration needs to be given to load balancing, user variability, and the heterogeneity of edge facilities in relevant research. In addition, most of the research around edge collaboration also revolves around cloud-side collaboration, which pays relatively little attention to the collaboration process between edge nodes, and the incentive and trust issues of the collaboration process need to be addressed. In this paper, we consider the impact of the user demand variability and the edge facility heterogeneity, then propose a method based on Vickrey-Clarke-Groves (VCG) auction theory to accommodate the edge demand response (EDR) process where the number of users and service facilities do not match. The method makes users' bidding rules satisfy the Nash equilibrium and weakly dominant strategy, which can improve the load balancing of edge nodes, has positive significance in improving the edge resource utilization and reducing the system energy consumption. In particular, combined with blockchain, we further optimize the incentive and trust mechanism of edge collaboration and consider three scenarios: no collaboration, internal collaboration, and incentive collaboration. We also consider the impact of the user task's transmission distance on the quality of experience (QoE). In addition, we illustrate the possible forking attack of blockchain in collaborative edge computing and propose a solution. We test the performance of the proposed algorithm on a real-world dataset, and the experimental results verify the algorithm's effectiveness and the edge collaboration's necessity.

**Keywords** Blockchain, Collaborative edge computing, IoT, Load balancing, VCG auction

## Introduction

With the development of the Internet of Things (IoT) and network communication technologies, a large amount of information is sent, transmitted, and processed in various forms, affecting all aspects of people's productive lives [1]. The rise of 5G technology has driven the development of mobile edge computing (MEC). Computation offloading, a very active topic in edge computing, investigates the process of transferring resource-intensive computational tasks from the resource-constrained user side to the cloud or edge side for processing, which involves allocating many resources [2]. Improving the load balancing of edge systems is beneficial to improve resource utilization, reduce system energy consumption, and improve system resilience to abnormal traffic attacks (e.g., distributed denial of service (DDoS) attacks), which is necessary for large-area, high-density 5G ultra-dense cellular networks [3].

Different user demands pose new challenges to edge devices. For demands such as autonomous driving technology and cloud gaming, lower task processing latency and lower user-side power consumption are required.

*Correspondence:
Jianmao Xiao
jm_xiao@jxnu.edu.cn
[1] School of Software, Jiangxi Normal University, Nanchang, China
[2] College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3] Department of Physics, Jiangxi Normal University, Nanchang, China
[4] College of Intelligence and Computing, Tianjin University, Tianjin, China

For IoT devices such as intelligent cameras and temperature sensors, real-world data processing and uploading are needed, as well as larger emergency processing and computing capabilities. Meanwhile, demands for video transmission, hotspot content queries, and other services require new requirements for edge caching. Correspondingly, domestic manufacturers have introduced different types of servers. For example, Huawei has launched general entry-level and general computing servers, while Alibaba has set up computing, general, and memory-based servers based on the ratio of CPU and memory.

Currently, some scholars have conducted in-depth research on computation offloading in edge computing environments and achieved some results. References [4–27] consider the problem of edge demand response (EDR) from the perspectives of communication resources, computing resources, and IoT device power consumption, mainly involving theoretical methods such as reinforcement learning and game theory. For example, references [4–6, 9–11, 13, 21, 22] mainly applied reinforcement learning methods to the specific allocation of edge resources. For example, references [19–27] further considered edge collaboration issues, but mainly focused on cloud-edge collaboration, with less consideration given to collaborative processing among edge nodes.

Due to limited edge resources, servers belonging to different edge facility providers lack the incentive to assist other servers in completing user requests. Moreover, edge collaboration among different application providers may generate user privacy and quality of experience (QoE) issues due to external factors or malicious competition among merchants. Therefore, there is a need to record the collaboration process to trace the execution of user tasks and to reward users based on the performance of their tasks. In order to facilitate traceability and improve the credibility of the recorded content, the recorded content should be public and untameable, and the traditional centralized storage method is obviously unsuitable.

Blockchain is tamper-proof, traceable, and maintained by multiple parties [24, 28, 29], which can be used to store the server execution information divided among different edge facility providers. However, the following two points should be noted. First, the information recorded by the blockchain is open and transparent. The storage method determines that the information recorded at each time should be small. Sensitive information should be avoided as much as possible during the recording process, focusing on metrics that evaluate the effectiveness of task execution (e.g., time, quality of user experience). Second, since each edge facility provider in a certain area has installed multiple servers to handle user requests, paying attention to forking attacks is necessary. If most edge servers (more than 50%) involved in edge collaboration belong to the same edge provider, illegal modification of information may occur, such as overwriting, disregarding inconvenient data in history, and eliminating convenient data of other servers, thus affecting the reputation value of edge servers and creating a bad cycle. The forking attack is a key concern for whether distributed ledger technology can be applied in edge collaboration, but it is a consideration that many scholars have overlooked.

In this paper, we will fully consider the variability of user demand and task volume, as well as the heterogeneous situation of edge servers. We first propose a computation offloading method based on the Vickrey-Clarke-Groves (VCG) auction in game theory. The method adapts to the edge demand scenario with the unequal number of users and edge servers. Compared to classical algorithms and some advanced algorithms, this method has certain improvements in terms of EDR latency, energy consumption, and load balancing. Based on the method, we combine blockchain to optimize the incentive and trust mechanism of edge collaboration to improve edge node load balancing further. In addition, we propose corresponding strategies from both hardware and software perspectives to respond to the forking attack issue that may arise in the process of blockchain application in edge collaboration. We conducted experiments on a real-world dataset from the Central Business District of Melbourne (EUA datasets) [30], and the experimental results verified the algorithm's effectiveness. The main contributions are as follows.

- We attempt to apply the VCG auction method in scenarios where the number of users and edge devices are unequal, to optimize the computation offloading and resource allocation process and improve the load balancing of edge systems. The study considers edge user variability and edge facility heterogeneity.
- We try to combine blockchain to optimize edge collaboration's incentive and trust mechanism. We consider user privacy in data disclosure and the potential forking attack pitfalls in the edge environment, and propose corresponding solutions.
- The performance is experimentally evaluated on a widely-used real-world dataset. Frequently, incentive collaboration outperforms internal collaboration and scenarios without collaboration, especially in service scenarios with uneven task distribution.

The rest of the paper is organized as follows: Related work section presents related work, System model and algorithm design section models the system and proposes solutions, Performance evaluation section designs experiments and evaluates the algorithm's performance, and Conclusion section summarizes the paper.

## Related work

Currently, some scholars have conducted research on the problem of computation offloading and edge collaboration in edge environments and achieved some results. Some relevant studies are summarized as follows:

The literature [4] addressed the integrity problem of managing high-density IoT devices in the current edge computing environment, modeled the allocation process of service resources as a Markov decision process (MDP), and trained the allocation policy to maximize the confidence gain by reinforcement learning (RL) methods. The literature [5] proposed a reinforcement learning-based task offloading strategy that considers offloading tasks from IoT devices to the edge to improve the battery lifetime. The literature [6] integrated the channel quality and queue state between the user and edge sides, modeled the computation offloading problem as a MDP, and proposed a deep Q-network-based offloading strategy to minimize the long-term cost. The literature [7] considered the cost-effectiveness issue of supporting non-orthogonal multiple access (NOMA) in the IoT scenario from the perspective of edge caching. To minimize energy consumption, the literature [8] investigated the problem of computation offloading and resource allocation in smart buildings and environments, and combines stochastic optimization techniques. Considering the limited resources of wireless networks and IoT devices, to balance the latency and energy consumption, the literature [9] represented the computation offloading and user scheduling process as a MDP and proposed a neural network architecture combined with deep reinforcement learning (DRL). Also using DRL method, the literature [10] studied the Industrial Internet of Things (IIoT) scenario with multiple IoT devices and multiple edge servers, and this article aimed to minimize long-term energy consumption. And the literature [11] proposed a dynamic task offloading strategy, aimed at optimizing the task scheduling problem in the MEC system of Digital Twin technology. The literature [12] considered scheduling cloud computing resources to reduce energy consumption on mobile, modeled the selection problem of mobile services as an energy consumption model, and solved it by genetic algorithms. The literature [13] applied blockchain to task mining, modeled the task offloading problem as a Markov decision, and introduced adaptive genetic algorithms to DRL to improve the convergence speed. To determine the user-task offloading ratio and implement adaptive task scheduling in a highly dynamic Telematics environment, the literature [14] proposed a bilateral matching algorithm to determine the optimal scheduling method, which calculates the offloading rate by convex optimization and thus implements a non-cooperative game. The literature [15] integrated computational and communication resources and proposed a primal pairwise optimization framework to schedule user tasks online, specifically considering the user's task size, latency and preferences to maximize social welfare. The literature [16] modeled the resource competition and service selection problem as a nonconvex optimization problem and proposed an online offloading strategy based on Lyapunov optimization for more complex edge environments. Combined with game theory, the literature [17, 18] explored the computation offloading problem from different perspective angles of QoS-aware and NOMA-enabled.

Considering the limited energy and resources at the edge, literature [19] proposed a cloud-side collaborative system to coordinate computation offloading and resource allocation at the cloud and edge, based on a simulated annealing algorithm for joint optimization to maximize profits. By applying game theory methods, the literature [20] studied the problem of multi-user computation offloading with cloud-edge collaboration, aiming to maximize the QoE of the users under limited communication and computing resources. To reduce the service latency and improve the quality of service of in-vehicle networks, literature [21] proposed a collaborative computing approach based on artificial intelligence to formulate the task offloading and computation process as a MDP. Combined with DRL, service delays and failure penalties are modeled as costs, which are minimized by collaborative workload and server selection. In collaboratively utilizing edge cloud and Internet of Vehicles (IoV) resources, the literature [22] proposed a DRL-based method to optimize computation offloading and resource allocation for vehicle tasks. The literature [23] designed an incentive mechanism based on auction theory to enable user devices and neighboring devices to help each other and thus optimize long-term system welfare. The literature [24] applied the blockchain to the collaborative process of edge caching by establishing reputation assessment through distributed consensus and using it to verify whether the edge servers are trustworthy as well as to motivate the servers to participate in the collaborative process. Literature [25] applied the game theory and the Lagrange multiplier method to the computation offloading and resource allocation decision process of cloud-edge collaboration, significantly improving the system utility and reducing task latency. The literature [26] jointly considered the problem of allocating computational and communication resources, transforming the problem into a convex optimization problem, and determining whether user tasks are uploaded to the cloud for processing, with the main objective of minimizing the weighting and latency of user devices. The literature [27] proposed a pipeline-based collaborative computing framework where each mobile user, edge node can offload tasks to different edge nodes or cloud for processing depending on the computational and communication capabilities to minimize the total latency, etc.

In this section, we reviewed research related to computation offloading and edge collaboration. However, these studies have somewhat overlooked the differences in user task demands and heterogeneity of edge servers. Research on edge collaboration has also mainly focused on cloud-edge collaboration, with less consideration given to the collaboration process among edge servers. In the next section, we will establish a system model for the efficiency, incentives, and trust issues in edge collaboration, and develop corresponding algorithms.

## System model and algorithm design

In this system, mobile users and IoT devices are denoted by $\mathcal{U} = \{u_1, u_2, ..., u_i\}$, edge servers are denoted by $\mathcal{S} = \left\{s_1, s_2, ..., s_j\right\}$, and physical machines within the edge servers are denoted by $\mathcal{K} = \left\{k_1, k_2, ..., k_t\right\}$. The requirements of different users or IoT devices vary and mainly involve communication, computation, and caching. In this article, we categorize user tasks into two types simply: computational tasks and cache tasks. Each edge server can install multiple physical facilities, such as five [31]. We assume that three of these are general-purpose devices, one is a computational device, and one is a cache device. The general-purpose devices maintain a relatively stable service rate when processing either compute-bound or cache-bound tasks, while the specialized devices' service rates are more affected by the type of task they receive. For example, when a computational server processes compute-bound tasks, its service rate will be higher than the general rate, and conversely, it will be lower than the general rate when processing cache-bound tasks. Combining the data sets, we construct the following general scenario diagram for what we are studying.

In the bottom half of Fig. 1, we represent the edge servers (usually base stations) and edge users (mobile users or IoT devices) in the dataset as solid red dots and purple dots, respectively. The top half of the figure describes real-world application scenarios and their corresponding network topologies. From the figure, it can be observed that there are many-to-many relationships between users and edge servers, where each user can choose from multiple edge servers and each edge server can serve multiple users within its service range. The edge servers can collaborate by exchanging data through wired or wireless means to further alleviate service pressure and improve service quality.

### VCG auction mechanism

In a heterogeneous edge network, the performance of different edge servers may be affected by the received task types. Users may conceal the actual task types, so it becomes more important to determine the actual task types of users in the EDR process. We have envisioned determining and learning the characteristics of different user categories by deep reinforcement learning. However, the higher online computational demands may not be well suited for the high latency required EDR process. Thus, we considered using game theory-related elements. We know that the VCG auction satisfies a second-order confined auction, where the price that each buyer will eventually pay is the next highest price, and all have the incentive to bid truthfully.

The mechanism's design contains two main parts: the allocation rule and the bid rule. The allocation rule can be efficient if it maximizes social welfare [32]. Combined with the idea of VCG auctions, we argue that the higher the service rate of the edge server selected for different users, the higher the utility obtained per unit of task volume processed, and the higher the utility users can obtain. When users with high task volume are assigned to the server with a high service rate in priority, the server's resources can be fully utilized, and social welfare can be maximized.

In the two-phase EDR process, which mainly contains two processes, user-edge server, and edge server-physical machine, the process is subject to capacity constraints, latency constraints, and proximity constraints in the edge computing [31]. We further consider the case of physical machine heterogeneity and classify them into general-purpose, computational, and caching based on the ratio of CPU and memory within different physical machines. The general-purpose physical machines are stable at a general level when handling user tasks, and the specialized physical machines have more processing power for specialized tasks but not for non-professional tasks, as evidenced by the service rate. Combining VCG auctions for a two-phase EDR process, physical machines with high service rates will bring higher utility per unit task volume and tend to assign users with high task volume to edge servers (physical machines) with high service rate, thus obtaining the greatest social welfare. For seller $s_j$ (server), the allocation rule is expressed as follows:

$$\begin{aligned} & Q_j^* \in arg \max_{p_{i,j} \in \Delta} \sum_{i \in U} p_{i,j} \cdot x_{i,j} \\ & \begin{cases} p_{i,j} = 1 \ if \ x_{i,j} \geq x_{-i,j} \\ p_{i,j} = 0 \ if \ x_{i,j} < x_{-i,j} \end{cases} \end{aligned} \tag{1}$$

where $x_{i,j}$ denotes the utility of user $u_i$ selecting edge server $s_j$; $x_{-i,j}$ denotes the utility of users other than user $u_i$ selecting edge server $s_j$; $p_{i,j}$ denotes the probability of user $u_i$ selecting edge server $s_j$.

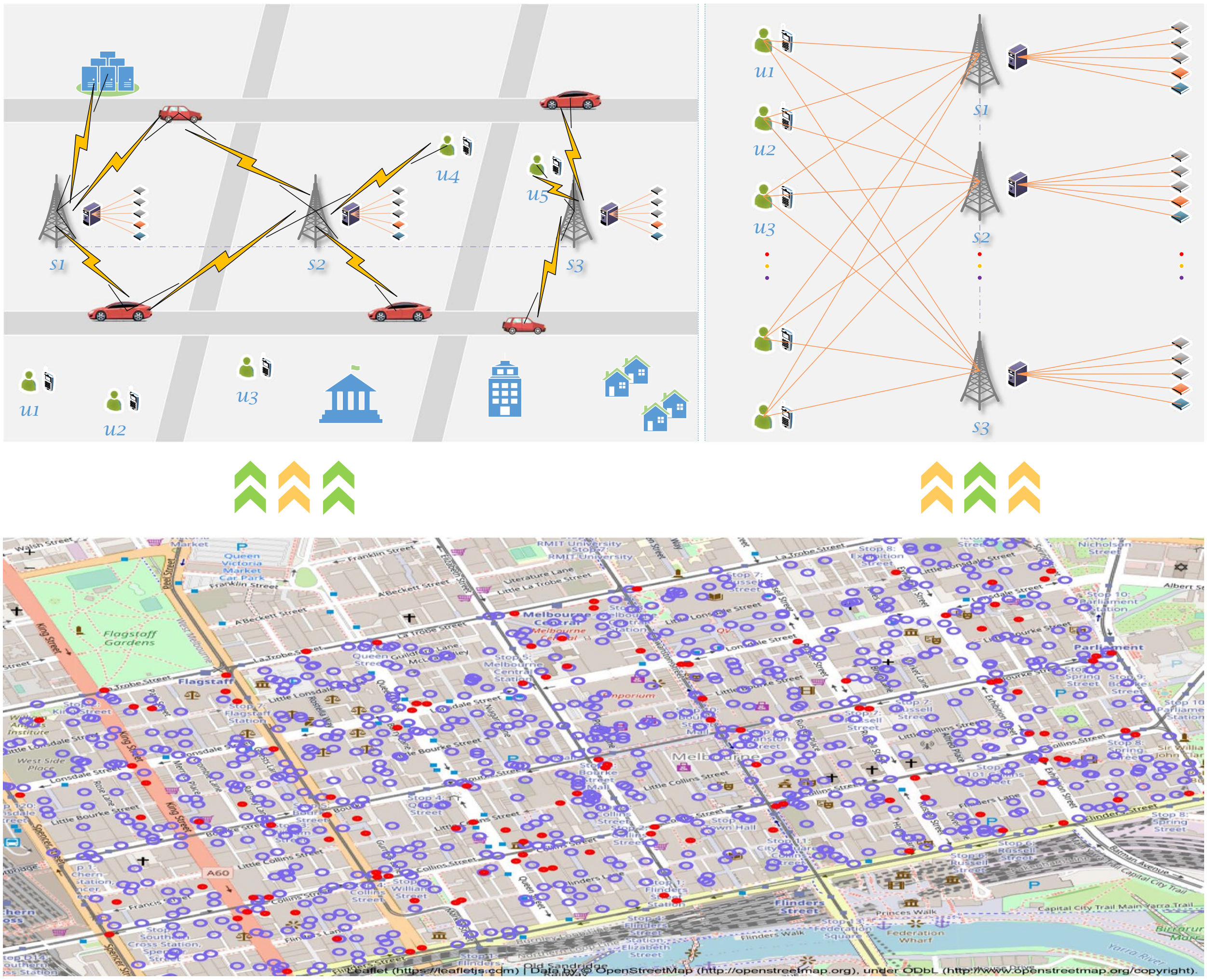

u4
u5
s1
s2
s3
u1
u2
u3
Flagstaff Gardens
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

**Fig. 1** EDR process in edge heterogeneous networks

After the allocation process, the price to be paid by the user needs to be further calculated. The bidding rule of VCG auction satisfies the Nash equilibrium, weak dominance strategy, which can ensure the user's true transmission of task type and task volume. The price to be paid by the user is the effect of the externality to other participants caused by the user's choice to participate in the bidding or not to participate in the bidding. If the externality is positive, the user receives revenue; if it is negative, it indicates that the user's participation brings negative externalities to others and needs to be paid. Specifically, suppose user $u_i$ is successful in the auction. In that case, the price to be paid by user $u_i$ is the difference between the sum of the utility values that all other participants would have received if user $u_i$ did not participate in the auction (or if the bid was zero) minus the sum of the utility values that all other participants would have received if user $u_i$ had participated in the auction. The social welfare $W_{i,j}$ of user $u_i$ participating in the auction and the social welfare $W_{-i,j}$ of user $u_i$ not participating in the auction are calculated as follows:

$$W_{i,j} = \sum_{i \in U} Q^*_{i,j} \cdot x_{i,j} \tag{2}$$

$$W_{-i,j} = \sum_{-i \in U(\neq i)} Q^*_{-i,j} \cdot x_{-i,j} \tag{3}$$

The payment rule and the equilibrium payoff for user $u_i$ are as follows:

$$\begin{aligned} M^V_{i,j} &= W_{-i,j} - \left( W_{i,j} - Q^*_{i,j} \cdot x_{i,j} \right) \\ &= \sum_{-i \in U(\neq i)} Q^*_{-i,j} \cdot x_{-i,j} - \begin{cases} 0 & if\ x_{i,j} \geq x_{-i,j} \\ \max\ x_{-i,j} & if\ x_{i,j} < x_{-i,j} \end{cases} \\ &= \max\ x_{-i,j} - \begin{cases} 0 & if\ x_{i,j} \geq x_{-i,j} \\ \max\ x_{-i,j} & if\ x_{i,j} < x_{-i,j} \end{cases} \end{aligned} \tag{4}$$

$$EP_i = Q_{i,j}^{*} \cdot x_{i,j} - M_{i,j}^{V} \tag{5}$$

For a bid, if the user bids the highest price, the user pays the next highest price; otherwise, the user does not get the bid and pays a price of 0. From here, we can also see that the VCG auction still satisfies the subprime auction mechanism that allows for efficient allocation (i.e., social welfare maximization), and the mechanism in which each buyer truthfully reports his or her information and pays its externalities is a weak dominance strategy.

In particular, in the multi-user-multi-server scenario, the equilibrium payoff for user $u_i$ is as follows:

$$EP_i^{*} = \sum_{j \in S} Q_{i,j}^{*} \cdot x_{i,j} - \sum_{j \in S} M_{i,j}^{V} \tag{6}$$

## Collaborative edge computing

Due to the variability in the performance of different edge nodes and the variability in the distribution of users in their service range, it is necessary to consider edge collaboration among edge servers. Since different edge servers may belong to different edge facility providers or be leased by different edge content providers, the incentive and trust issues among edge nodes need to be considered. Blockchain, which is tamper-proof, traceable, and maintained by multiple parties, is suitable for distributed edge collaborative computing scenarios and can be used to store server execution information belonging to different edge facility providers. However, attention should be paid to privacy protection and forking attack.

In terms of incentives, we use $\mathcal{CV}_j$ to denote the reputation value of edge server $s_j$. Nodes with high reputation value will receive task requests from other nodes with a higher probability. After each round of sub-task completion, the edge content provider will cash out the reward in equal proportion to the reputation value. The reputation value is cleared after cashing out. In order to ensure the traceability of the collaboration process between edge servers, the specific information of each user task processed by the edge collaboration is recorded as a block. The server that assists in completing the edge task has the authority to keep the account and update the reputation (the reputation value can be exchanged for the reward after the system completes the task per unit of time). The credit value is updated as follows:

$$\mathcal{CV}_j^{(n+1)} = \mathcal{CV}_j^{(n)} + \left[1 + \tau \cdot (1 - p_i)\right] \cdot \mathcal{CV} \tag{7}$$

$$p_i = \frac{d_{st,i}^{act}}{d_{st,i}^{ori}} \tag{8}$$

where $\mathcal{CV}_j^{n}$ denotes the reputation value of server $s_j$ after processing user task updates for the nth time, $\tau$ represents the weight value to assist the server to complete the task efficiently, and $p_i$ denotes the ratio of the actual sojourn time of user $u_i$'s task to the computation time processed at the initially assigned edge server. $\mathcal{CV}$ denotes the unit reputation for processing each task.

When performing edge collaboration, priority is given to edge servers with high utility values, which are calculated as follows:

$$\mathcal{EV}_{j,act}^{(n+1)} = \alpha \cdot \mathcal{CV}_{j,act}^{(n+1)} + \beta \cdot l_{s_j^{ori}, s_j^{act}} \tag{9}$$

$$\mathcal{A}_t = arg\ \max\left[\mathcal{EV}_{j,act}^{(n+1)}\right] \quad s_j^{act} \in S \tag{10}$$

where $\alpha$ and $\beta$ represent the weights of reputation value and distance, respectively, and $\alpha + \beta = 1$; $l_{s_j^{ori}, s_j^{act}}$ denotes the physical distance between the requester and the edge server expecting to receive the task.

After completing the task, the edge server with bookkeeping privileges records the task information as $\left\{u_i, task_{u_i}^{volu}, task_{u_i}^{hash}, task_{s_j^{act}}^{hash}, s_j^{ori}, s_j^{act}, d_{st}^{ori}, d_{st}^{act}, \mathcal{H}\right\}$ , where $task_{u_i}^{volu}$ , $task_{u_i}^{hash}$ denote the task volume of user $u_i$ and the corresponding hash value of the specific task, respectively, $s_j^{ori}$ , $d_{st}^{ori}$ denote the edge server initially assigned to the user task and its corresponding processing time, $s_j^{act}$ , $d_{st}^{act}$ denote the edge server actually processing the user task and the actual processing time, and $task_{s_j^{act}}^{hash}$ denote the hash value of the task execution. $\mathcal{H}$ denotes the hash value of the information recorded in the previous block. The edge server will publish the content after bookkeeping, and other edge servers will make a copy of the book content after checking that the content is correct. At this point the new block has been verified and can be added to the blockchain.

We know that when the arithmetic power of one party in the blockchain exceeds 50% of the overall arithmetic power of the system, the probability of the system suffering from a forking attack will be greatly increased. This will fundamentally affect the tamper-evident characteristics of the blockchain and reduce the system's trustworthiness. In the collaborative edge computing environment, multiple edge servers in a certain area may belong to the same edge facility provider, which is similar to the "mining pool" mentioned in blockchain and must be taken seriously. Based on the system model, we propose three possible adverse effects of a forking attack, and make some suggestions from both hardware and software perspectives. The details are shown in Fig. 2.

In Scenario# 1, since one of the servers of edge facility provider A has a problem in co-processing task $t_3$, to avoid adverse effects, the edge facility provider proactively abandons the reward from task $t_3$ and initiates a forking attack to overwrite the record. In Scenario# 2, edge facility provider B performs well in co-processing task $t_3$, and its reputation value increases significantly. Competitor A maliciously launches a forking attack to overwrite the record. In Scenario# 3, during the reputation value accumulation process, edge facility provider A has more opportunities to participate in edge collaboration than other edge facility providers due to its larger computing resources. Furthermore, the collaborative computing process increases the reputation value. Over time, the reputation value of the servers under edge facility provider A will become larger and larger, leading to more possibility of its participation in collaborative computing until it has full control of the system. At this point, it will deviate from the original purpose of the distributed ledger.

We propose the following recommendations to prevent forking attacks for those three possible scenarios mentioned above. First, the edge facility providers to which they belong should be considered when selecting edge servers as co-servers. As much as possible, each edge facility provider should provide the similar number of servers with similar performance to reduce the possibility of forking attack from the hardware level. Second, in the algorithm scheduling process of edge collaborative computing, a threshold value should be set to prevent servers belonging to the same edge provider from continuously participating in collaborative computing. Finally, the system should reduce the period of redeeming reputation value to avoid excessive accumulation of reputation value to affect system stability.

By applying blockchain, we can ensure that the records of task execution are continuous and tamper-proof, which is beneficial for establishing a trust mechanism among edge nodes. We can observe that edge servers with high task execution efficiency will receive higher reputation values, leading to more rewards or compensation. This can incentivize servers to actively participate in edge collaboration while ensuring the quality of task execution.

The pseudo-code for algorithm implementation is as follows: Pseudo-code 1 describes the process of computation offloading and resource allocation, mainly involves two-phase of users to edge servers and physical facilities. Lines 1 to 13 are the preparation process. Lines 14 to 37 represent the process of user tasks to edge servers. Lines 38 to 41 represent the task allocation process to physical facilities within the edge server, similar to the first stage but considering the impact of user and physical facility types in the response process. Pseudo-code 2 describes the basic process of edge collaboration. Lines 1, 2, 4, and 5 are incentive and selection processes. Lines 3 and 6 consider privacy and security issues in conjunction with blockchain.

In this section, we first investigate how to use VCG auction theory to improve the two-phase EDR when the number of users and edge devices does not match. This is the foundation of studying edge collaboration. Subsequently, we combine blockchain to study the incentive and trust issues in edge collaboration. We will verify the performance of the proposed methods in the next section.

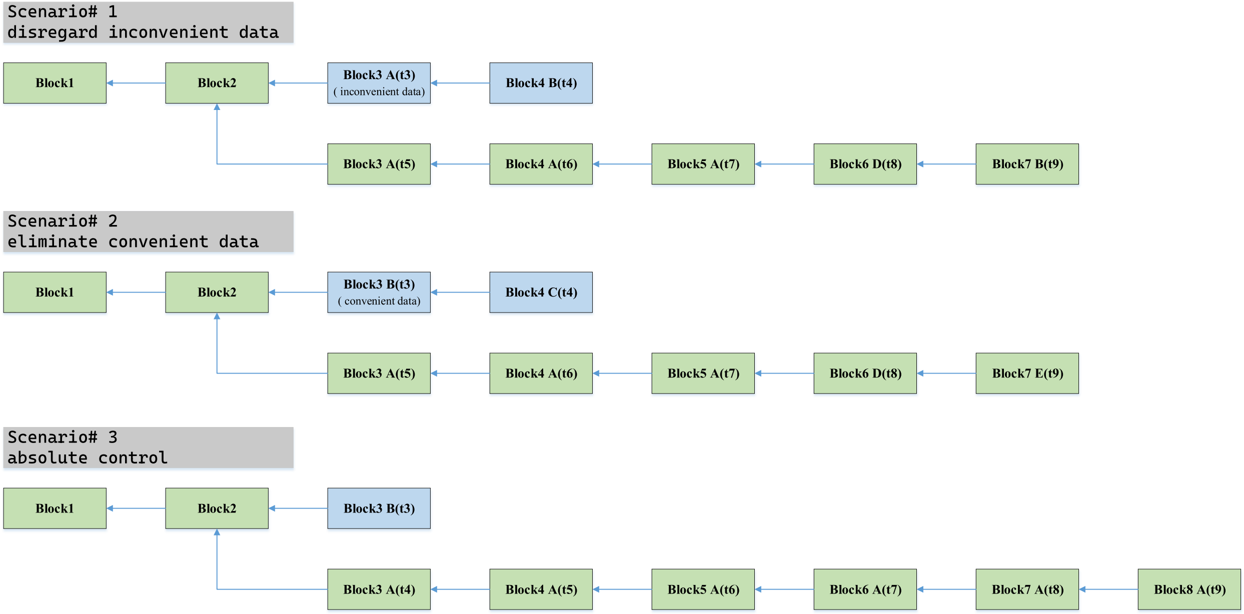


**Fig. 2** The forking attack's impact on collaborative edge computing

```
Require: $\mathcal{U} = \{1, 2, \dots, u\}$, $\mathcal{S} = \{1, 2, \dots, s\}$, $\mathcal{K} = \{1, 2, \dots, k\}$
Based on the effective signal coverage of the edge server, obtain the set of servers that can be selected by users and obtain the set of users that the edge server can serve.
for $s_j \in \mathcal{S}$ do
    for $m_e \in \mathcal{K}_{s_j}$ do
        Generate uniformly distributed random variables $\lambda_e$, $\mu_e$, $\zeta_e$ from [0,1)
        if $\mu_e \le \lambda_e$ then
            $\mu_e \leftrightarrow \lambda_e$
        end if
    end for
    Obtain the mean value $\lambda_j$, $\mu_j$, $\zeta_j$
end for
for $u_i \in \mathcal{U}$ do
    Generate normally distributed random variables $\tau_i$ from (5,10)
end for
Phase #1
Sort the users by task volume (from largest to smallest) and get a new set $\mathcal{U}^*$.
for $u_i \in \mathcal{U}^*$ do
    if $u_i$ not in $\mathcal{U}^{'}$ then
        From the set of user-selectable edge servers $\mathcal{S}_i$, select the server with the highest average service rate, and record the server number in $index$.
        while $index$ in $\mathcal{S}^{'}$ do
            After ignoring the server with the number $index$, reselect the edge server with the highest average service rate and update the number of the newly selected server to $index$.
            Check if the user selectable list becomes empty. If the list is empty, record the user number in the list $\mathcal{U}^{''}$ and abort the one-level loop.
        end while
        if $u_i$ not in $\mathcal{U}^{''}$ then
            Record the user-server ($u_i$ - $index$) selection relationship.
            Update the set of successfully assigned users $\mathcal{U}^{'}$.
            Update the set of selected edge servers $\mathcal{S}^{'}$.
        end if
        if $(i+1)$ % $s$ == 0 then
            The edge servers not selected in this round are sorted from the largest to the smallest based on the average service rate. Subsequently, for each edge server, the user with the highest task volume is selected from its optional user list (note: users cannot be selected repeatedly).
            Clear the list $\mathcal{S}^{'}$.
        end if
        if if $(i+1)$ % $s$ == 0 or $(i+1) == u$ then
            Assign users in list $\mathcal{U}^{''}$ that have not been successfully assigned an edge server.
        end if
    end if
end for
The batch process ends and users who still have not been assigned an edge server are processed and assigned to the server with the highest service rate in their optional range.
Phase #2
for $s_j \in \mathcal{S}$ do
    Reassign the users in the server to each physical machine. Consider the relationship between user type and physical machine service rate.
end for
```

**Algorithm 1** E-VCG

```
Require: $\mathcal{U} = \{1, 2, \dots, u\}$, $\mathcal{S} = \{1, 2, \dots, s\}$, $\alpha$, $\beta$, $\tau$
$\mathcal{EV}_{j,act}^{(n+1)} = \alpha \cdot \mathcal{CV}_{j,act}^{(n)} + \beta \cdot l_{s_j^{ori}, s_j^{act}}$
$\mathcal{A}_t = arg\ \max\left[\mathcal{EV}_{j,act}^{(n+1)}\right] \quad s_j^{act} \in S$
Judge whether to select servers of the same edge facility provider for more than 6 consecutive times. If yes, choose again.
$p_i = \frac{d_{st,i}^{act}}{d_{st,i}^{ori}}$
$\mathcal{CV}_j^{(n+1)} = \mathcal{CV}_j^{(n)} + [1 + \tau \cdot (1 - p_i)] \cdot \mathcal{CV}$
The task recipient has permission to record in the ledger due to his work contribution, and $\left\{u_i, task_{u_i}^{volu}, task_{u_i}^{hash}, task_{s_j^{act}}^{hash}, s_j^{ori}, s_j^{act}, d_{st}^{ori}, d_{st}^{act}, \mathcal{H}\right\}$ is added as a new block after the other servers check that the content is correct.
```

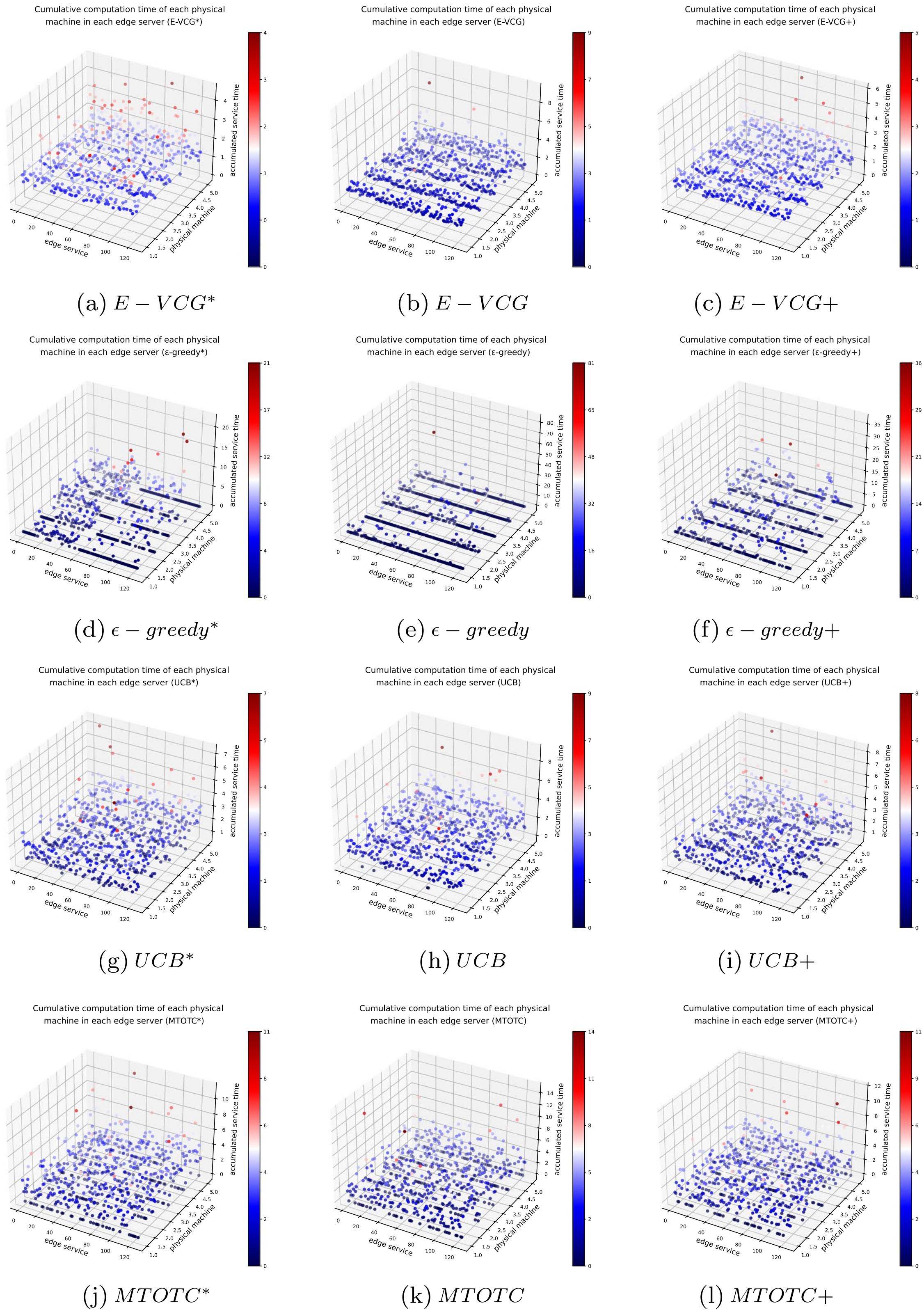
**Algorithm 2** + Cooperative Computing

## Performance evaluation

In this section, we conducted experiments based on a real-world dataset from the Central Business District of Melbourne to evaluate the performance of our algorithm. The selected dataset includes 125 service base stations and 816 randomly selected users. By utilizing this location information, we can further determine the physical distances between users, between edge servers, and between users and edge servers. Combining this with the proximity constraints of the edge environment, we can further determine the subset of relationships between users and edge servers and proceed with the experiments. We conducted experimental comparisons with the following algorithms in different scenarios:

- Improved $\varepsilon - greedy$: $\varepsilon$ keeps getting smaller as the number of selections increases, and the edge node with the highest utility value is explored or selected with a certain probability.
- *UCB*: First explore all optional but not yet selected edge servers, then select the edge node with the highest utility value, and update the utility value after each selection.
- *MTOTC*: For each user, the selection probability of all its selectable nodes sums to 1. The game stops when the probability of a user selecting a node is 1 or exceeds a set threshold.

In this article, we mainly focus on the edge-to-end collaboration issues in edge demand response processes, specifically examining the load balancing of edge servers under different methods. The experimental process considers the diversity of user demands and the heterogeneity of edge facilities. Specifically, the task types and volumes of users may differ, and the service preferences of the physical facilities opened in edge servers may also vary. Initially, we only consider the computing offloading



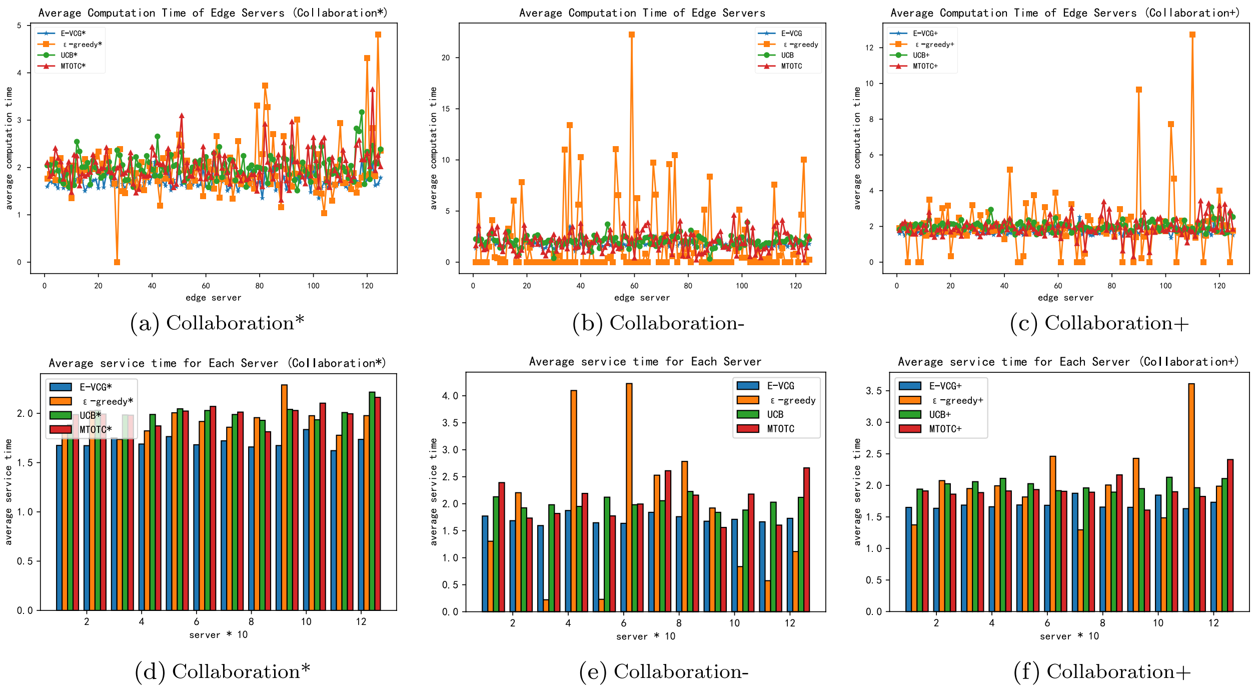
(a) $E-VCG*$ (b) $E-VCG$ (c) $E-VCG+$

(d) $\epsilon-greedy*$ (e) $\epsilon-greedy$ (f) $\epsilon-greedy+$

(g) $UCB*$ (h) $UCB$ (i) $UCB+$

(j) $MTOTC*$ (k) $MTOTC$ (l) $MTOTC+$

**Fig. 3** Cumulative service time for each physical machine



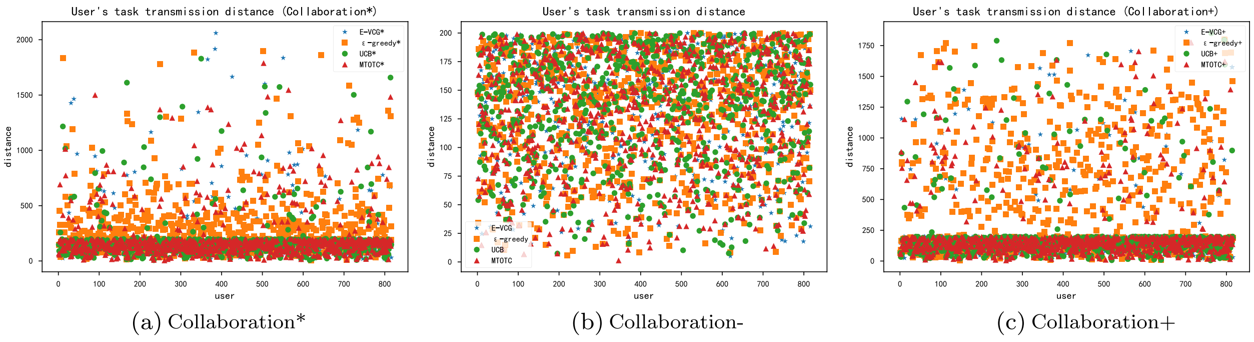
(a) Collaboration* (b) Collaboration- (c) Collaboration+

(d) Collaboration* (e) Collaboration- (f) Collaboration+

**Fig. 4** Average service time for each server

and resource allocation processes between users and edge servers. Subsequently, since each edge facility provider may install multiple edge service devices in the area, we further consider the unconditional collaboration between these service devices (referred to as internal collaboration). In practical environments, as each edge server may belong to different edge facility providers, the overall distribution of edge facility providers is relatively uniform, as they consider the environmental characteristics and user situation when deploying devices. However, internal service collaboration within the provider may lead to longer-distance information transmission and thus affect user experience. Therefore, we further consider the edge collaboration between different edge facility providers (referred to as incentive collaboration).

Figure 3 illustrates the distribution of cumulative computation time of physical facilities within each edge server under three different scenarios: internal collaboration, incentive collaboration, and no collaboration, combined with different methods. In Fig. 3, the first column shows the distribution of cumulative computation time of physical facilities within each edge server under the incentive collaboration scenario after responding to user demands. The second and third columns respectively represent the specific results without collaboration and with internal collaboration. In addition, each row in the figure compares the distribution of cumulative computation time of each method under different scenarios. From the figure, it can be seen that the cumulative computation time of the E-VCG, UCB, and MTOTC



(a) Collaboration* (b) Collaboration- (c) Collaboration+

**Fig. 5** User's task transmission distance

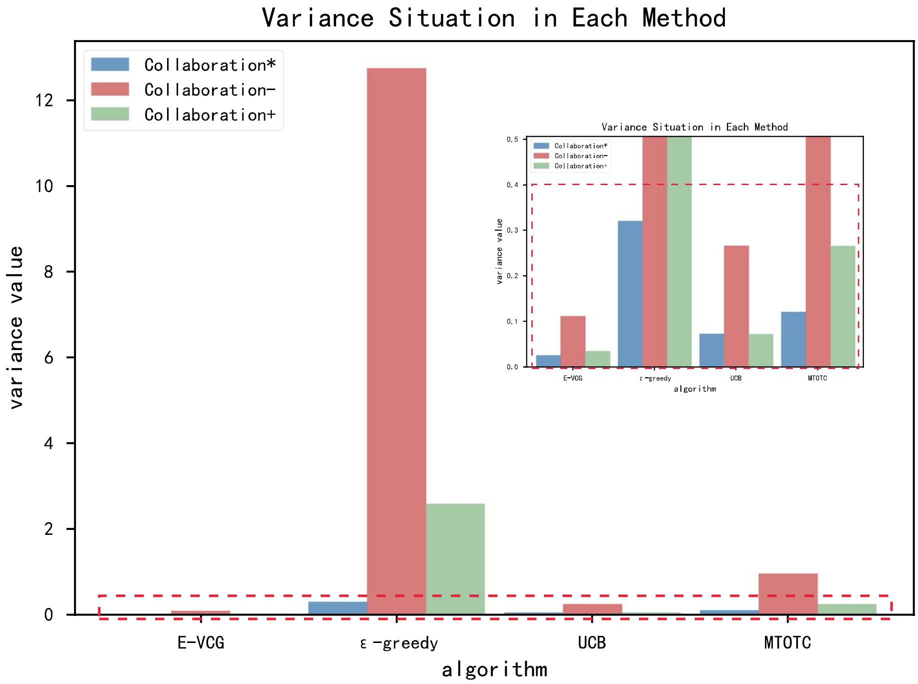


**Fig. 6** Variance comparison

methods is stably distributed at a low level. The $\varepsilon$-greedy method tends to select physical facilities with higher service rates, resulting in higher service times for some physical facilities. After adding incentive collaboration or internal collaboration, the distribution of cumulative computation time of each method is improved to a certain extent. In particular, from Fig. 3(d-f), it can be seen that collaboration methods are more effective for algorithms with uneven demand response, and the incentive collaboration method is slightly better than the internal collaboration method.

Subsequently, we calculated the average of the cumulative service time for each physical facility within the edge server and compared the differences between the servers by the average. We represent the distribution of the average computation time across the edge servers in Fig. 4. Again, the first column represents the effect of adding incentive synergy, the second column is the effect without considering synergy, and the third column is the effect with considering internal synergy. Figure 4(a-c) represents the point distribution, from which we can see that the MTOTC algorithm, UCB algorithm and E-VCG algorithm have relatively balanced performance, and E-VCG performs better. After adding synergy, the distribution of the average computation time of each method becomes more convergent, especially the $\varepsilon$-greedy method. Further, we take the average of 10 edge servers as a unit and represent it as Fig. 4(d-f). From the figure, we can see that the E-VCG method slightly outperforms the UCB method, and the UCB method slightly outperforms the MTOTC method. The overall distribution effect after adding internal collaboration is better than without considering collaboration, and the incentive collaboration is slightly better than internal collaboration.

Considering that the task transmission process may cause significant transmission latency and affect the user quality of experience, we have collected data on the transmission distances of user tasks in three different scenarios, which are presented in Fig. 5. Figure 5(b) shows the task transmission distances without considering edge collaboration. Due to the proximity constraint of the edge environment, the transmission distances for all users do not exceed 200 $m$. In Fig. 5(a,c), points that are farther than 200 $m$ indicate that the user tasks were processed through edge collaboration. From the figure, we can visually see that the transmission distances for incentive collaboration mostly remain within 750 $m$, which is a certain advantage compared to internal collaboration.

To more specifically quantify the relevant metrics, we calculated the variances based on the average computation time of each edge server and represented them in Fig. 6. We believe that Fig. 6 can reflect the performance of each method in terms of load balancing in the edge demand response process. After adding edge collaboration, the comparison of each method showed the same trend as before, while varying degrees of improvement in load balancing. Specifically, after considering internal collaboration, the E-VCG, $\varepsilon$-greedy, UCB, and MTOTC methods improved load balancing by 68.63%, 79.60%, 73.21%, and 72.75%, respectively. After considering incentive collaboration, the E-VCG, $\varepsilon$-greedy, UCB, and MTOTC methods improved load balancing by 76.53%, 97.49%, 72.89%, and 87.59%, respectively. Interestingly, taking the UCB and MTOTC methods as examples, we can see that the UCB method is slightly better than the MTOTC method, but the MTOTC method with incentive collaboration is better than the UCB method without collaboration. This indirectly illustrates the positive significance of edge collaboration in balancing edge task distribution. In addition, based on the performance changes of the $\varepsilon$-greedy method, we can conclude that edge collaboration performs better in scenarios where task distribution is uneven.

## Conclusion

In this paper, we investigate the problem of computation offloading and collaborative edge computing, considering the heterogeneity of edge users and service facilities. We first propose the E-VCG method to schedule the limited edge resources. The method makes the bidding rules of users satisfy the Nash equilibrium, improving the edge resource utilization. Subsequently, we discuss edge collaboration with blockchain, specifically considering three forms of no-cooperation, internal collaboration, and incentive collaboration. We also elaborate on the possible forking attack by blockchain in edge collaboration applications and propose some solutions. We conduct experiments on real-world datasets, and the experiments verify the necessity of edge collaboration and the effectiveness of the proposed method.

This research primarily considers computation offloading and edge collaboration in static scenarios, and relevant experiments are also conducted on static datasets. In future research, the impact of user mobility can be considered in dynamic scenarios.

### Abbreviations

| | |
|---|---|
| DRL | Deep reinforcement learning |
| DDoS | Distributed denial of service |
| EDR | Edge demand response |
| IoT | Internet of Things |
| IIoT | Industrial Internet of Things |
| IoV | Internet of Vehicles |
| MEC | Mobile edge computing |
| MDP | Markov decision process |
| NOMA | Non-orthogonal multiple access |
| QoE | Quality of experience |
| RL | Reinforcement learning |
| VCG | Vickrey-Clarke-Groves |

### Acknowledgements

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

### Authors' contributions

Qinghang Gao created the model and conducted the main theoretical and formula derivation. He designed the experiment and algorithm, visualized and analyzed the experimental results, and wrote the original draft. Jianmao Xiao directed the planning and execution of research activities. He assisted with the theory and formula derivation, controlled the experiments, and checked and embellished the content and syntax. Yuanlong Cao coordinated the planning and execution of the research activities. Shuiguang Deng, Chuying Ouyang, and Zhiyong Feng advised and provided technical support for the main model building, experimental design, revision, and embellishment of the manuscript content. All authors have read and approved the manuscript.

### Funding

This work is supported by Jiangxi Provincial Natural Science Foundation under Grant No.20224BAB212015, 20224ACB202007, Jiangxi Province Science and Technology Project (03 Special 5G Project) under Grant No.20224ABC03A13, the National Natural Science Foundation of China (NSFC) under Grant No.51962010 and No. 61962026, the National Natural Science Key Foundation of China under Grant No.61832014.

### Availability of data and materials

The data used to support the findings of this study is cited in the article and can be viewed via the link https://github.com/swinedge/eua-dataset.

## Declarations

### Competing interests

The authors declare no competing interests.

Received: 28 November 2022 Accepted: 30 April 2023
Published online: 08 May 2023

### References

1. Dong P, Ge J, Wang X, Guo S (2022) Collaborative edge computing for social internet of things: Applications, solutions, and challenges. IEEE Trans Comput Soc Syst 9(1):291–301. https://doi.org/10.1109/TCSS.2021.3072693
2. Deng S, Zhao H, Fang W, Yin J, Dustdar S, Zomaya AY (2020) Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet Things J 7(8):7457–7469. https://doi.org/10.1109/JIOT.2020.2984887
3. Gao Q, Wang H, Wan L, Xiao J, Wang L (2022) G/m/1-based ddos attack mitigation in 5g ultradense cellular networks. Wirel Commun Mob Comput 2022. https://doi.org/10.1155/2022/4282859
4. Deng S, Xiang Z, Zhao P, Taheri J, Gao H, Yin J, Zomaya AY (2020) Dynamical resource allocation in edge for trustable internet-of-things systems: A reinforcement learning method. IEEE Trans Ind Inform 16(9):6103–6113. https://doi.org/10.1109/TII.2020.2974875
5. Min M, Xiao L, Chen Y, Cheng P, Wu D, Zhuang W (2019) Learning-based computation offloading for iot devices with energy harvesting. IEEE Trans Veh Technol 68(2):1930–1941. https://doi.org/10.1109/TVT.2018.2890685
6. Chen X, Zhang H, Wu C, Mao S, Ji Y, Bennis M (2018) Performance optimization in mobile-edge computing via deep reinforcement learning. In: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall). pp 1–6. https://doi.org/10.1109/VTCFall.2018.8690980
7. Chen Y, Xing H, Ma Z, et al (2022) Cost-efficient edge caching for noma-enabled iot services. China Commun
8. Li K, Zhao J, Hu J et al (2022) Dynamic energy efficient task offloading and resource allocation for noma-enabled iot in smart buildings and environment. Build Environ. https://doi.org/10.1016/j.buildenv.2022.109513

9. Lei L, Xu H, Xiong X, Zheng K, Xiang W, Wang X (2019) Multiuser resource control with deep reinforcement learning in iot edge computing. IEEE Internet Things J 6(6):10119–10133. https://doi.org/10.1109/JIOT.2019.2935543
10. Huang J, Gao H, Wan S et al (2023) Aoi-aware energy control and computation offloading for industrial iot. Futur Gener Comput Syst 139:29–37
11. Chen Y, Gu W, Xu J, et al (2022) Dynamic task offloading for digital twin-empowered mobile edge computing via deep reinforcement learning. China Commun
12. Deng S, Wu H, Tan W, Xiang Z, Wu Z (2017) Mobile service selection for composition: An energy consumption perspective. IEEE Trans Autom Sci Eng 14(3):1478–1490. https://doi.org/10.1109/TASE.2015.2438020
13. Qiu X, Liu L, Chen W, Hong Z, Zheng Z (2019) Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing. IEEE Trans Veh Technol 68(8):8050–8062. https://doi.org/10.1109/TVT.2019.2924015
14. Ning Z, Dong P, Wang X, Hu X, Liu J, Guo L, Hu B, Kwok RYK, Leung VCM (2022) Partial computation offloading and adaptive task scheduling for 5g-enabled vehicular networks. IEEE Trans Mob Comput 21(4):1319–1333. https://doi.org/10.1109/TMC.2020.3025116
15. Li G, Cai J (2020) An online incentive mechanism for collaborative task offloading in mobile edge computing. IEEE Trans Wirel Commun 19(1):624–636. https://doi.org/10.1109/TWC.2019.2947046
16. Zhao H, Du W, Liu W, Lei T, Lei Q (2018) Qoe aware and cell capacity enhanced computation offloading for multi-server mobile edge computing systems with energy harvesting devices. In: 2018 IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI. pp 671–678. https://doi.org/10.1109/SmartWorld.2018.00133
17. Chen Y, Hu J, Zhao J, Min G (2023) QoS-Aware Computation Offloading in LEO Satellite Edge Computing for IoT: A Game-Theoretical Approach. Chin J Electron
18. Chen Y, Zhao J, Hu J, et al (2023) Distributed Task Offloading and Resource Purchasing in NOMA-enabled Mobile Edge Computing: Hierarchical Game Theoretical Approaches. ACM Transactions on Embedded Computing Systems
19. Yuan H, Zhou M (2021) Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems. IEEE Trans Autom Sci Eng 18(3):1277–1287. https://doi.org/10.1109/TASE.2020.3000946
20. Chen Y, Zhao J, Wu Y et al (2022) Qoe-aware decentralized task offloading and resource allocation for end-edge-cloud systems: A game-theoretical approach. IEEE Trans Mob Comput. https://doi.org/10.1109/TMC.2022.3223119
21. Li M, Gao J, Zhao L, Shen X (2020) Deep reinforcement learning for collaborative edge computing in vehicular networks. IEEE Trans Cogn Commun Netw 6(4):1122–1135. https://doi.org/10.1109/TCCN.2020.3003036
22. Huang J, Wan J, Lv B, Ye Q et al (2023) Joint computation offloading and resource allocation for edge-cloud collaboration in internet of vehicles via deep reinforcement learning. IEEE Syst J. https://doi.org/10.1109/JSYST.2023.3249217
23. He J, Zhang D, Zhou Y, Zhang Y (2020) A truthful online mechanism for collaborative computation offloading in mobile edge computing. IEEE Trans Ind Inform 16(7):4832–4841. https://doi.org/10.1109/TII.2019.2960127
24. Yuan L, He Q, Chen F, Zhang J, Qi L, Xu X, Xiang Y, Yang Y (2022) Csedge: Enabling collaborative edge storage for multi-access edge computing based on blockchain. IEEE Trans Parallel Distrib Syst 33(8):1873–1887. https://doi.org/10.1109/TPDS.2021.3131680
25. Zhao J, Li Q, Gong Y, Zhang K (2019) Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. IEEE Trans Veh Technol 68(8):7944–7956. https://doi.org/10.1109/TVT.2019.2917890
26. Ren J, Yu G, He Y, Li GY (2019) Collaborative cloud and edge computing for latency minimization. IEEE Trans Veh Technol 68(5):5031–5044. https://doi.org/10.1109/TVT.2019.2904244
27. Kai C, Zhou H, Yi Y, Huang W (2021) Collaborative cloud-edge-end task offloading in mobile-edge computing networks with limited communication capability. IEEE Trans Cogn Commun Netw 7(2):624–634. https://doi.org/10.1109/TCCN.2020.3018159
28. Gadekallu TR, Pham QV, Nguyen DC, Maddikunta PKR, Deepa N, Prabadevi B, Pathirana PN, Zhao J, Hwang WJ (2022) Blockchain for edge of things: Applications, opportunities, and challenges. IEEE Internet Things J 9(2):964–988. https://doi.org/10.1109/JIOT.2021.3119639
29. He Y, Wang Y, Qiu C, Lin Q, Li J, Ming Z (2021) Blockchain-based edge computing resource allocation in iot: A deep reinforcement learning approach. IEEE Internet Things J 8(4):2226–2237. https://doi.org/10.1109/JIOT.2020.3035437
30. Lai P, He Q, Abdelrazek M, Chen F, Hosking J, Grundy J, Yang Y (2018) Optimal edge user allocation in edge computing with variable sized vector bin packing. In: International Conference on Service-Oriented Computing. Springer, Hangzhou, pp 230–245
31. Cui G, He Q, Xia X, Chen F, Gu T, Jin H, Yang Y (2021) Demand response in noma-based mobile edge computing: A two-phase game-theoretical approach. IEEE Trans Mob Comput 1–1. https://doi.org/10.1109/TMC.2021.3108581
32. Krishna V (2009) Auction theory. Academic press, Cambridge

## Publisher's Note

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