Cluster Computing (2023) 26:635–649
https://doi.org/10.1007/s10586-022-03651-x

Joint resource trading and computation offloading in blockchain enhanced D2D-assisted mobile edge computing

Weipeng Jing[1] · Xinyu Fu[1] · Peng Liu[1] · HouBing Song[2]

Received: 16 December 2021 / Revised: 14 May 2022 / Accepted: 17 June 2022 / Published online: 22 August 2022

Abstract
D2D-assisted mobile edge computing has attracted much attention due to its great potential to reduce the pressure on edge servers and improve the utilization of mobile devices. However, users are usually reluctant to share resources with neighbor devices, considering the limitations on resources and energy, and the concerns about security and privacy. In this paper, we firstly propose a blockchain-based resource-sharing system to solve the trust issues between unfamiliar participants. Under this system, we jointly consider resource trading and computation offloading to reduce energy consumption on the premise of maximizing the utilities of both providers and requesters. Resource trading is modeled as Stackelberg Game, and computation offloading is formalized as a mixed optimization problem. Finally, we propose the corresponding algorithms to efficiently get the optimal trading and offloading strategy. The simulation results show the effectiveness of incentivizing more participants.

Keywords Blockchain · D2D · MEC · Resource trading · Computation offloading

1 Introduction

Recent years have witnessed the explosive growth of mobile devices, followed by the ever-enriching mobile applications [1–3]. Computation-intensive applications such as interactive gaming, face recognition, and augmented reality need to be run on mobile devices with limited computing capacity and constrained battery power, which brings great challenges. Mobile Edge Computing (MEC) has gradually become an indispensable paradigm as a new solution. By deploying servers at the edge of the network, MEC can not only provide resources for mobile devices but also overcome the shortcomings of latency and security of Mobile Cloud Computing (MCC) [4–6]. With the help of Device-to-Device (D2D) communication, mobile devices could even offload their tasks to neighbors with idle resources [7, 8]. D2D-assisted MEC has the potential to reduce the pressure on edge servers and access networks, improve the utilization and well adapt to the mobility of mobile devices, which has attracted much attention [9].

Despite the great potential, D2D-assisted MEC faces several tricky challenges. Above all, users may be reluctant to share resources with neighbor devices for free, especially considering their limitations on resources and energy, and their concerns about security. An effective incentive mechanism first needs a sharing system to ensure device security and user privacy. On this basis, an equitable trading strategy is needed to maximize the utilities of both providers and requesters, considering that the interest damage to either part may influence the incentive effect. Besides, most existing research ignored that computation offloading is also an indispensable process for incentive [10]. Users will be willing to participate in

✉ Peng Liu
liupeng@nefu.edu.cn

Weipeng Jing
jwp@nefu.edu.cn

Xinyu Fu
fuxy@nefu.edu.cn

HouBing Song
Houbing.Song@erau.edu

1 College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China

2 Department of Electrical, Computer, Software and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA

resource sharing if they can complete tasks with lower energy consumption by the traded resources.

As a typical Peer-to-Peer (P2P) network, D2D-assisted MEC has no central authority to enforce security policies and lack of trust between unfamiliar participants, which has security and privacy implications. Fortunately, blockchain has been envisioned as a promising approach to such issues, which has been widely used in several domains to build trust and enforce security policies by its entirely decentralized, transparent, tamper-proof character [11–13]. Hence, we introduce blockchain technology to build a secure and reliable resource-sharing system. Under this system, we jointly study the strategy of resource trading and computation offloading, and program them into the smart contract which can not only be triggered automatically on the blockchain but ensures traceability and irreversibility [14, 15].

Unlike other resource-sharing research that focused on the multi-to-one model in MEC, D2D-assisted MEC is more suitable for the one-to-multi model because idle resources of one mobile device are hard to complete the tasks from multiple mobile devices. Hence, we model the resource trading process as a one-leader and multi-followers Stackelberg Game to maximize the utilities of both requesters and providers. Moreover, the computation offloading process is formalized into a mixed optimization problem that jointly optimizes the local CPU frequency, transmission time, and transmission bandwidth to minimize energy consumption. By joint consideration of resource trading and computation offloading, the proposed strategy is able to motivate more users to participate in resource sharing.

The main contributions of this paper are as follows:

- We use blockchain to build a decentralized resource sharing system in D2D-assisted MEC, where mobile devices could be used to share resources without any concerns about security and privacy.
- We design an incentive mechanism under a blockchain-based resource sharing system that jointly considers resource trading and computation offloading processes which could motivate more users to participate in resource sharing.
- We model resource trading as a one-leader and multi-follower Stackelberg game to maximize the utility of both CRR and CPRs. We formalize computation offloading into a mixed optimization problem, and a KKT-based algorithm is proposed to get the optimal strategy efficiently and accurately.
- Numerical simulations are conducted to evaluate the performance of the proposed algorithm for the resource sharing problem. Simulation results show that our work has the potential to motivate more participants.

The rest of this article is organized as follows: Sect. 2 reviews related work of resource trading and computation offloading. Section 3 introduces the blockchain-based resource sharing system; Sect. 4 explains the system model, including the resource trading model and computation offloading model; Sect. 5 introduces the simulation experiment and shows the analysis of the experimental results; Sect. 6 concludes this paper.

2 Related work

So far, numerous studies have been carried out on resource sharing problems in the context of edge computing. These works can be classified into the following three categories.

2.1 Blockchain framework for resource sharing

Many existing works introduce blockchain technology to solve the security and privacy issues among untrusted users in distributed and peer-to-peer networks [16]. The authors in [17–20] studied multi-user resource trading in the blockchain. Liu proposed a blockchain-based MEC network architecture in [17] to handle the large-scale video requests from distinct user devices in a decentralized and secure manner. The author in [18] proposed a blockchain-based energy trading scheme to supervise and manage the energy trading process and designed a timed commitments-based mechanism to guarantee verifiable fairness during energy trading. The author in [19] set up an operator-assisted data offloading platform powered by blockchain to implement a rating system for sellers and to make reliable transactions for payments. In [20], Chen presented a secure user offloading mechanism in heterogeneous wireless networks. Considering that the conventional auction-based trading participants may collude or take selfish actions, the authors employed the Ethereum framework for trustless, secure, and distributed auctioning.

2.2 Blockchain-based incentive mechanism

In [21], Li established a pure P2P computing resource trading system in a blockchain network to achieve secure and trusted computing resource trading. The authors introduced a broker to manage the trading market and then, proposed an iterative double-sided auction-based algorithm for computing resource trading. The author in [22] proposed a blockchain-based framework for MEC video streaming and designed an incentive mechanism to facilitate collaboration among content creators, video transcoders, and consumers. The authors in [23] addressed the joint computation offloading and coin-loaning problem. They introduced the banks to provide loan services to mobile devices and minimized the cost of all mobile devices based on a game-theory-based method. The authors in [24] proposed a secure, decentralized Internet of Vehicles(IoV) data-trading system by exploiting blockchain technology. They designed an efficient debit-credit mechanism to support efficient data-trading in IoV and formulated a two-stage Stackelberg game to maximize the profits of borrower vehicles and lender vehicles jointly.

2.3 Task offloading strategy in MEC

Computation offloading is a technique that has the potential to improve application performance in MEC [25]. In the work of [26–28], different performance optimization schemes were designed to minimize energy or time consumption by optimizing computing and offloading strategies. The authors of [26] have discussed a game-theoretic model in a blockchain-based peer-to-peer network of mobile users to compute tasks of mobile devices with low computation power in a cost and time-optimal way. In [27], the authors designed a secure data offloading strategy based on the hashgraph consensus mechanism and implemented a game-theoretic bargaining model to ensure task computation in a cost-optimal and time-efficient manner. In [28], Min investigated multiuser partial computation offloading in a quasi-static scenario and proposed a dynamic matching algorithm to minimize terminal energy consumption. The authors in [29, 30] studied the problem of maximizing energy utility. The author in [29] proposed a D2D offloading architecture to offload wireless traffic and tasks and introduced offloading and balancing techniques to increase the total energy efficiency. In [30], a resource allocation strategy in a D2D-assisted edge computing system with hybrid energy harvesting has been proposed to maximize energy efficiency. Quantum-behaved particle swarm optimization (QPSO) algorithm was used to obtain the suboptimal solution.

However, most studies have not considered that one device does not have enough idle resources to service multiple devices in a normal D2D scenario, and how to reduce the energy consumption while guaranteeing the high utility of both the CRR and CRPs. Therefore, we proposed a blockchain-based resource sharing model in which multiple providers provide computing services to one requester. The study devised the resource trading and computation offloading algorithms which achieved the optimal results jointly in the D2D-assisted MEC network.

3 Blockchain-based computing resource sharing system

This section explains how resource trading and computation offloading work on the blockchain. We use resource coins to stimulate devices to participate in resource tradings. CRPs can earn resource coins by selling their idle resources, and CRRs pay resource coins to purchase computing resources from CRPs. As shown in Fig. 1, we consider the resource sharing system consisting of edge servers, mobile devices, and blockchain. The MEC servers are deployed at the edge of the D2D-assisted MEC network and have sufficient storage resources to handle the massive data records in resource sharing. The trusted local base stations(LBS) provide the necessary network assistance to trigger the execution of smart contracts. The resource trading and computation offloading mechanism can be encoded as a series of smart contracts, which are placed into the blockchain. The smart contract is automatically triggered to obtain an optimal resource-sharing strategy. The following are more details about key operations of blockchain-enabled resource sharing.

(1) System initialization: In order to ensure the authenticity and integrity of digital messages, the blockchain system needs to be initialized by the cryptography technique. Each user equipment in the D2D-assisted MEC network becomes a legitimate participant after registration on the trusted LBS. LBS gives each legitimate participant an identity ID (ID), a public key (PLk), a private key (PVk), a certificate (Cert) and a wallet address (WA).

(2) Trigger smart contract for resource trading and computation offloading: Users who lack computing resources send LBS requests to purchase resources. CRR_i's request information Req_i contains task details $(task_i)$, public key (PLk_i), certificate $(Cert_i)$, timestamp for request generation (ts_i), where $(task_i)$ includes the number of resources required by the computing task $(taskR_i)$, local idle resources $(locR_i)$, the geographic location of CRR_i $(site_i)$ and the maximum acceptable resource price $(\bar{p}_i)$. After receiving Req_i, CRP_j sends equipment information Rep_i to LBS. Rep_i contains service information $(service_j)$, public key (PLk_j), certificate $(Cert_j)$, and time stamp generated in response (ts_j), where $service_j$ includes the number of free resources of CRP_j $(number_j)$, the distance between CRP_j and CRR_i $(distance_j)$. After validating certification, LBS writes the received information into the smart contract. This smart contract will be automatically implemented by using the proposed trading model in Sect. 4.1 and the computation offloading model in Sect. 4.2.

(3) Payment with resource coins: According to the resource price derived from the resource trading smart contract, the CRRs use their private key to unlock some of their Unspent Transaction Output (UTXO), and sign the latest transactions that transfer money to the corresponding CRPS' address. The

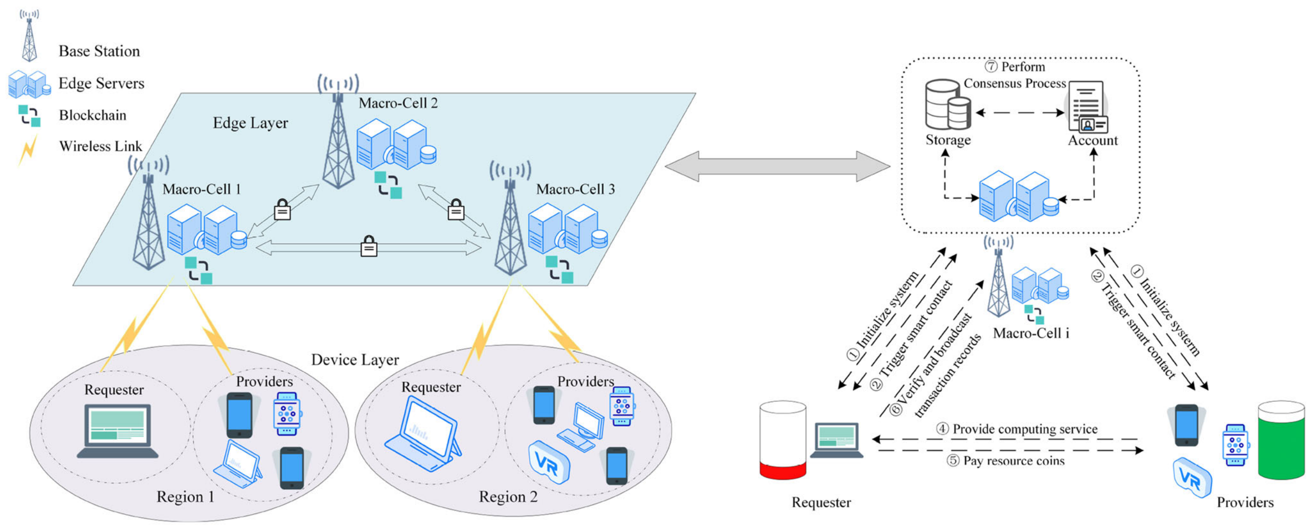

Fig. 1 Blockchain-based computing resource sharing system

resource coin can be transferred from the wallet address of the CRRs to CRPs [9].

(4) Verification and broadcast trading records: After completing the resource tradings and computation offloading, CRRs obtain the resource trading records. Then they encrypt these whole transactions with signatures and broadcast them to all users.

(5) Perform the proposed proof of supply score consensus process: Each user has a corresponding supply score(s_j), and the supply score increases as resources supply increases. We set $s_j = s'_j + a_j$, where s'_j and a_j represent the existing supply score and new resources supply. The user with the highest supply score constructs a new block composed of all transactions, including resource transaction records and the updated supply score of the CRPs. The new block is broadcast to other user devices. These user devices audit the block data and broadcast their audit results with their signatures to each other. Meanwhile, each user device also needs to verify whether the building block device has the highest supply score and send the verification result to other user devices. If all the user devices agree on the block data, the new data block will be placed into the blockchain based on the timestamp. Finally, the supply score of the building block device halves($s_j = 1/2s'_j$).

4 System model for D2D computing resource trading and computation offloading

As shown in Fig. 2, the proposed blockchain-based computing resource sharing system contains two core system submodels: the resource trading model and the computation offloading model. In this section, it is considered that n CRPs provide computing resources to one CRR.

4.1 Resource trading model

To construct an efficient resource trading scheme, we model the interaction between CRPs $\mathbf{S}$ and the CRR B as a Stackelberg game, which consists of two types of participants: leader and followers. In the proposed D2D-assisted MEC, the B acts as the leader to determine resource allocation, and the $\mathbf{S}$ act as the followers to set the price.

4.1.1 Problem formulation for resource allocation

Let $\mathbf{S} = \{S_1, ..., S_j, ..., S_n\}$ and B denote the set of CRPs and the CRR in resource sharing. The demands of B from different CRPs are defined as a vector $\mathbf{a} = \{a_1, ..., a_j, ..., a_n\}$. $U_b(\mathbf{a})$ is defined as the utility of completing tasks of B, which consists of the local utility and offloading utility. Referring to [21] and [31], here the utility function of B is as follows.

$$U_b(\mathbf{a}) = \ln\Big(1 + d_{max} - \sum_{j=1}^{N} a_j\Big) + \sum_{j=1}^{N} [\omega \ln(a_j - a_j * z_j + 1) - a_j p_j], \tag{1}$$

where d_{max} denotes the number of resources required to complete the tasks. a_j indicates the number of computing resources purchased from the CRP S_j. Considering that the buying willingness increases with the increasing demand, the willingness of the CRR B is given by

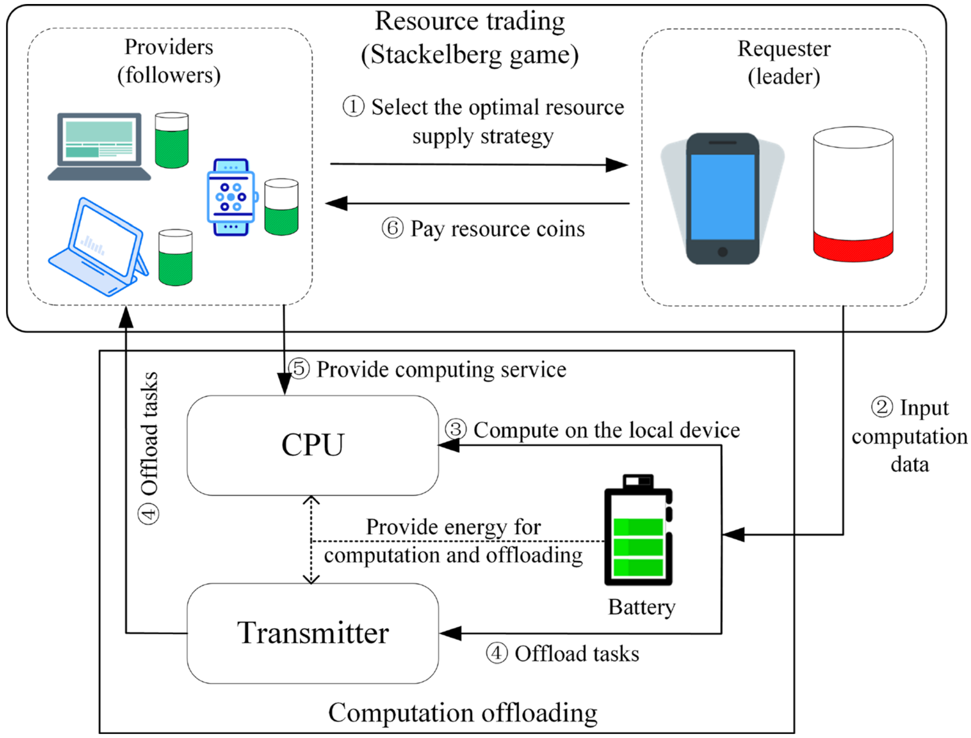

Fig. 2 D2D computing resource trading and computation offloading

$$\omega = \frac{1}{1 - \frac{d_{min}}{d_{max}}}, \tag{2}$$

where d_{min} represents the minimum number of resources that buyer B needs to buy. z_j is the distance factor, which is proportional to the distance between B and the CRP S_j. p_j denotes the resource pricing proposed by S_j. Therefore, to maximize the utility of B, the resource allocation problem can be expressed as P1:

$$\begin{aligned} P1 : \max_{\mathbf{a}} \quad & U_b(\mathbf{a}), \\ s.t. \quad & C1 : 0 \le a_j \le R_j, \\ & C2 : d_{min} \le \sum_{j=1}^{N} a_j \le d_{max}, \end{aligned} \tag{3}$$

where $C1$ ensures the resource demand from the CRP S_j does not exceed the idle resource of S_j, and cannot be negative. Constraint $C2$ makes sure that the total number of purchased resources is sufficient and not excessive.

4.1.2 Problem formulation for resource pricing

The CRP S_j sets resource pricing to minimize utility. We consider that the utility function of S_j consists of two parts, the utility generated by self-used resources and selling resources, where the self-used resources are the spare resources left after selling. The utility generated by selling is related to resource price, service cost, and competition with other providers. Let C_j be the cost of the CRP S_j, which is given as

$$C_j = n_1 a_j^2 + n_2 a_j, \tag{4}$$

where $n_1 > 0$, $n_2 > 0$ are cost factors [21].

There is competition among providers. Providers who bid less are more competitive. The competitiveness coefficient of the CRP S_j is expressed as

$$k_j = \frac{\bar{p} - p_j}{\sum_{k=1}^{N} \bar{p} - p_k}, \tag{5}$$

where $\bar{p} > 0$ is the maximum price CRPs can propose [32].

Therefore, we propose the utility function of the CRP S_j as follows

$$U_s(p_j) = \ln[1 + (R_j - a_j)] + k_j(a_j p_j - C_j), \tag{6}$$

where R_j is the idle resources of the CRP S_j. a_j indicates the resource demand from S_j. p_j means the resource price quoted by S_j. To maximize the utility of S_j, we can formulate the resource pricing problem as P2:

$$\begin{aligned} P2 : \max_{p_j} \quad & U_s(p_j), \\ s.t. \quad & C3 : 0 \le p_j \le \bar{p}, \\ & C4 : a_j p_j - (n_1 a_j^2 + n_2 a_j) \ge 0, \end{aligned} \tag{7}$$

where the constraint $C3$ limits the value range of p_j. $C4$ aims to enable S_j to obtain positive utility in resource trading.

4.1.3 Stackelberg game in resource trading

The Stackelberg game has been widely adopted to capture the sequential interactions among strategic agents [32]. A game of resource allocation and pricing exists between the CRR B and CRPs $\mathbf{S}$. We model it as a Stackelberg game. In the Stackelberg game, the CRR as the leader first announces the different resource demands from different CRPs. CRPs as followers, follow this announcement by announcing the price they set. Both the leader and followers constantly adjust their strategies for the sake of more utility. The CRR adjusts the resource allocation strategy to obtain the maximum utility by solving P1 and CRPs adjust their pricing strategies by solving problem P2.

Game Analysis: We have already formulated the mathematical model of the two sides in the Stackelberg game. Specifically, the goal of the Stackelberg game is to find the Nash equilibrium. The Nash equilibrium is the optimal result of the game, where no participant has the motivation to deviate from its strategy after considering the opponent's choice [33]. In our problem, the Nash equilibrium of the proposed Stackelberg game is defined as follows.

Let $\mathbf{a}^*$ and $\mathbf{p}^*$ be the optimal resource allocation and the optimal pricing, respectively. Then, when $(\mathbf{a}^*, \mathbf{p}^*)$ satisfies $U_b(\mathbf{a}^*|\mathbf{p}^*) \geq U_b(\mathbf{a}|\mathbf{p}^*)$,

$$\begin{aligned} &U_s(p_j^*|a_j^*) \geq U_s(p_j|a_j^*), a_j^* \in \mathbf{a}^*, p_j^* \in \mathbf{p}^*, \\ &a_j \in \mathbf{a}, p_j \in \mathbf{p}, j \in \{1, 2, ..., n\}, \end{aligned}$$

then $(\mathbf{a}^*, \mathbf{p}^*)$ is the Nash equilibrium point.

To verify the uniqueness and existence of the Nash equilibrium in our Stackelberg game, we take first and second order derivatives of the S_j's utility function with respect to $\mathbf{p}$. Let $p_o = \sum_{k \neq j}^N \bar{p} - p_k$. The derivatives are given by:

$$\begin{aligned} \frac{\partial U_s(p_j)}{\partial p_j} = &\frac{-p_o}{(p_o + \bar{p} - p_j)^2} \cdot (a_j p_j - C_j) + a_j \\ &\cdot \left(1 - \frac{p_o}{p_o + \bar{p} - p_j}\right) \end{aligned} \tag{8}$$

$$\frac{\partial^2 U_s(p_j)}{\partial p_j^2} = \frac{-2a_j p_o}{(p_o + \bar{p} - p_j)^2} + \frac{-2p_o(a_j p_j - C_j)}{(p_o + \bar{p} - p_j)^3}. \tag{9}$$

Then according to constraints $C3$ and $C4$, we have $\frac{\partial^2 U_s(p_j)}{\partial p_j^2} \leq 0$. When $\frac{\partial U_s(p_j)}{\partial p_j} = 0$, the optimal price of S_j is given by:

$$p_j^* = p_o + \bar{p} - \sqrt{p_o^2 + p_o \bar{p} + n_2 p_o - n_1 p_o a_j}. \tag{10}$$

Then we define $b_o = p_o^2 + p_o \bar{p} + n_2 p_o$ and put Eq. (10) into (1). The second-order derivative of the function (1) with respect to a_j is obtained as

$$\begin{aligned} \frac{\partial^2 U_b(\mathbf{a})}{\partial a_j^2} = &\frac{-1}{(1 + d_{max} - \sum_{j=1}^{N} a_j)^2} + \frac{-\omega(1 - z_j)^2}{(a_j - a_j * z_j + 1)^2} \\ &+ \frac{-n_1 p_o}{\sqrt{b_o - n_1 p_o a_j}} + \frac{-n_1^2 p_o^2 a_j}{4\sqrt{(b_o - n_1 p_o a_j)^3}} \leq 0 \end{aligned} \tag{11}$$

Therefore, $U_b(\mathbf{a})$ and $U_s(p_j)$ are strictly concave, and Nash equilibrium exists in this Stackelberg game.

4.1.4 Resource trading algorithm based on game theory

It is assumed that n CRPs are participating in the trading. The competitiveness coefficient k_j and resource supplied of each CRP a_j are set to $1/n$ and d_{max}/n. In the first round of the game, CRPs solve problem P2 according to the initial $\{a_j\}$ and $\{k_j\}$. Then get the pricing strategy $\mathbf{p} =$

Algorithm 1 Resource Trading Algorithm Based on Game Theory

Require: d_{max}, d_{min}, n_1, n_2, $\mathbf{z} = \{z_1, ..., z_j ... z_n\}$, n.
Ensure: Optimal resource allocation strategy $\mathbf{a}$ and pricing strategy $\mathbf{p}$.

```
Initialization: a_j = d_max/n, k_j = 1/n, j ∈ {1, 2, ..., n}, i = 1.
The CRP S_j solves P2 according to a_j. Obtain the latest pricing set p_new
and utility set u_i^s
repeat
    i=i+1
    Solve P1 based on p_new to obtain a new resource allocation set a_new
    and utility u_i^b.
    Update k_j and solve P2 based on a_new. Update p_new and get u_i^s.
until u_i^b = u_{i-1}^b and u_i^s = u_{i-1}^s
return a_new, p_new
```

$\{p_1, p_2, ..., p_n\}$ and the CRPs' utility set $\mathbf{u^s} = \{u_1^s, u_2^s, ..., u_n^s\}$. In the next rounds, the CRR uses the latest pricing strategy to solve the problem P1 to obtain a new resource allocation strategy $\mathbf{a}$ and utility u^b. Then, each CRP updates its competitiveness coefficient k_j based on the pricing strategy in the previous round and uses the latest k_j and resource demand a_j to solve P2 to get new pricing. The iteration ends when CRR's utility and each CRP's utility stop changing. Finally, we obtain a resource trading strategy that meets the utility of both CRR and CRPs.

4.2 Computation offloading model

After obtaining the resource trading strategy, the resource allocation information is written into the smart contract of computation offloading. The smart contract is triggered automatically to minimize the energy for completing tasks of the CRR B.

4.2.1 Problem formulation for energy-efficient computation offloading

The energy consumption can be divided into two parts: computation energy and offloading energy, and the model is presented as follows:

Computation energy consumption model: We model the computation energy consumption as $P = \kappa f^3$, where f and κ are a coefficient depending on chip architecture and computational speed measured by the number of CPU cycles per second. Note that f can be flexibly adjusted via DVFS technology to satisfy users' demands [33]. When the computation resource consumed by the task is a, the computational time is $t_{com} = a/f$, the corresponding energy consumption is $E_{com} = a\kappa f^2$. Therefore, the local computation energy consumption of the CRR B is given by

$$E_{req} = (d_{max} - \sum_{j=1}^{N} a_j)\kappa f_l^2, \tag{12}$$

where f_l denotes B's CPU frequency. d_{max} is the number of computing resources required to complete tasks. a_j denotes the computing resource supplies of S_j derived from the proposed trading model.

In addition, the total computation energy consumption of CRPs $\mathbf{S}$ is expressed as

$$E_{pro} = \sum_{j=1}^{N} a_j \kappa f_j^2, \tag{13}$$

where f_j denotes S_j's CPU frequency.

Offloading energy consumption model: Let c_j denote the number of CPU cycles required for computing 1-bit data on the CRP S_j. The quantity of data to be processed in S_j is a_j/c_j. Then the transmission rate can be expressed as

$$r_j = \frac{a_j}{c_j t_j}, \tag{14}$$

where t_j is the time allocated to the CRP S_j for offloading. Let h_j denote the channel gain and P_j denote the transmission power. Then the achievable rate is denoted as

$$r_j = \eta_j B \log_2(1 + \frac{P_j h_j^2}{\sigma}), \tag{15}$$

where B and σ are the bandwidth and the variance of complex white Gaussian channel noise. η_j indicates the ratio of bandwidth allocated for offloading tasks to S_j. Let $\lambda_j = 1/\eta_j$. Then according to (14) and (15), we derive

$$P_j = \frac{(2^{\frac{a_j \lambda_j}{c_j B t_j}} - 1)\sigma}{h_j^2}. \tag{16}$$

Define a function $f(x) = \sigma(2^{\frac{a_j x}{c_j B}} - 1)$. The energy consumption for offloading a_j/c_j-bit data to S_j is denoted as

$$E_j(\lambda_j, t_j) = P_j t_j = \frac{t_j}{h_j^2} f(\frac{\lambda_j}{t_j}). \tag{17}$$

Therefore, the total energy consumption of offloading can be expressed as

$$E_{off} = \sum_{j=1}^{N} \frac{t_j}{h_j^2} f(\frac{\lambda_j}{t_j}). \tag{18}$$

In summary, the energy-efficient computation offloading problem with latency, bandwidth, transmission power, and CPU frequency constraints can be stated as

$$\begin{aligned} P3: \min_{f_l, \boldsymbol{\lambda}, \mathbf{t}} \quad & E(f_l, \boldsymbol{\lambda}, \mathbf{t}) = E_{req} + E_{pro} + E_{off} \\ s.t. \quad & C5: 0 < f_l \leq f_l^{max}, \\ & C6: t_{loc} \leq T, \\ & C7: \sum_{j=1}^{N} \frac{1}{\lambda_j} \leq 1, \\ & C8: \lambda_j \geq 1, \\ & C9: t_j \geq 0, \\ & C10: t_j + \frac{a_j}{f_j} \leq T, \\ & C11: 0 \leq P_j \leq P_j^{max}, \end{aligned} \tag{19}$$

where f_l^{max} denotes maximum CPU frequency of the CRR B. t_{loc} and T are local computing time and maximum computational delay. $\boldsymbol{\lambda} = \{\lambda_1, ..., \lambda_j, ..., \lambda_n\}$ donates bandwidth allocation set. $\mathbf{t} = \{t_1, ...t_j, ..., t_n\}$ is the time allocation set. P_j^{max} refers to the maximum transmission power of the CRP S_j. Constraint $C5$ limits the range of local CPU frequency. $C6$ represents latency requirements. $C7$ and $C8$ state that all allocated bandwidth to CRPs can not exceed B's available bandwidth. $C9$ represents the non-negative feature of the allocated offloading time. $C10$ indicates that the sum of S_j's offloading time and computation time cannot exceed the maximum delay. $C11$ is the peak and non-negative transmission power constraint imposed by S_j.

4.2.2 Energy-efficient computation offloading scheme

The energy-efficient computation offloading problem can be solved in two parts since the local computation and offloading to other devices are executed synchronously.

The first part minimizes the local computing energy consumption by adjusting the calculation frequency of B. The local computing energy minimization problem is formulated as follows:

$$
\begin{aligned}
P4 : \min_{f_l} \quad & E_{req}(f_l), \\
s.t. \quad & C5 : 0 < f_l \leq f_l^{max}, \\
& C6 : t_{loc} \leq T, \\
& t_{loc} = \frac{d_{max} - \sum_{j=1}^{N} a_j}{f_l}.
\end{aligned}
$$

$E_{req}(f_l)$ is a monotonically increasing function in $(0, f_l^{max}]$. Therefore, when f_l takes the minimum value, the local computational energy consumption is the minimum. Because f_l is inversely proportional to t_{loc}, when t_{loc} takes the maximum value T, f_l takes the minimum value, and the local calculation consumes the least energy. The optimal local computation frequency is given by:

$$
f_l = \frac{d_{max} - \sum_{j=1}^{N} a_j}{T}. \tag{20}
$$

In the second part, the offloading bandwidth and offloading time are optimally allocated to minimize the offloading energy consumption. The offloading energy minimization problem is formulated as:

$$
\begin{aligned}
P5 : \min_{\boldsymbol{\lambda}, \mathbf{t}} \quad & E_{off}(\boldsymbol{\lambda}, \mathbf{t}) = \sum_{j=1}^{N} \frac{t_j}{h^2} f\left(\frac{\lambda_j}{t_j}\right), \\
s.t. \quad & C7 : \sum_{j=1}^{N} \frac{1}{\lambda_j} \leq 1, \\
& C8 : \lambda_j \geq 1, \\
& C9 : t_j \geq 0, \\
& C10 : t_j + \frac{a_j}{f_j} \leq T, \\
& C11 : 0 \leq P_j \leq P_j^{max}.
\end{aligned}
$$

Since $f(x)$ is a convex function when $x \geq 0$, its perspective function $\frac{t_j}{h^2} f(\frac{\lambda_j}{t_j})$ is jointly convex with respect to $t_j \geq 0$ and $\lambda_j \geq 0$ [34]. Thus, the objective function, the summation of a set of convex functions preserves the convexity.

We define $m_j = a_j/(c_j B)$ and $T_j = T - a_j/f_j$. To solve the problem $P5$, the corresponding Lagrange function is written as:

$$
\begin{aligned}
L = & \sum_{j=1}^{N} \left[\frac{t_j}{h^2} \cdot \sigma \left(2^{\frac{m_j \lambda_j}{t_j}} - 1 \right) + \mu_j (t_j - T_j) \right] \\
& + \gamma \left(\sum_{j=1}^{N} \frac{1}{\lambda_j} - 1 \right),
\end{aligned} \tag{21}
$$

where μ_j and γ are the Lagrange multipliers associated with allocated time and bandwidth respectively. The Karush-Kuhn-Tucker (KKT) conditions for the problem $P5$ are given by:

$$
\begin{aligned}
\frac{\partial L}{\partial t_j} = & \frac{\sigma}{h^2} \cdot (2^{\frac{m_j \lambda_j}{t_j}} - 1) + \frac{t_j}{h^2} \cdot \sigma \cdot \ln 2 \cdot 2^{\frac{m_j \lambda_j}{t_j}} \cdot (-\frac{m_j \lambda_j}{t_j^2}) + \mu_j \\
& = 0,
\end{aligned} \tag{22}
$$

$$
\frac{\partial L}{\partial \lambda_j} = \frac{m_j}{h^2} \cdot \sigma \cdot \ln 2 \cdot 2^{\frac{m_j \lambda_j}{t_j}} - \frac{\gamma}{\lambda_j^2} = 0, \tag{23}
$$

$$
t_j - T_j \leq 0, \quad \mu_j \geq 0, \quad \mu_j (t_j - T_j) = 0, \tag{24}
$$

$$
\sum_{j=1}^{N} \frac{1}{\lambda_j} - 1 \leq 0, \quad \gamma \geq 0, \quad \gamma (\sum_{j=1}^{N} \frac{1}{\lambda_j} - 1) = 0. \tag{25}
$$

It can be derived from (22) that

$$
2^{\frac{m_j \lambda_j}{t_j}} (1 - \ln 2 \cdot \frac{m_j \lambda_j}{t_j}) + \frac{h^2 \mu_j - \sigma}{\sigma} = 0. \tag{26}
$$

From (23) we have

$$
\lambda_j^2 \cdot 2^{\frac{m_j \lambda_j}{t_j}} - \frac{h^2 \gamma}{m_j \sigma \ln 2} = 0. \tag{27}
$$

In (25) there is $\gamma \geq 0$. Nevertheless, when $\gamma = 0$, Eq. (27)

has no solution. Therefore, we obtain $\gamma > 0$. The condition $\mu_j \geq 0$ exists in (24). We obtain $\lambda_j = 0$ by putting $\mu_j = 0$ into Eq. (26), which does not satisfy constraint $C8$. Then we get $\mu_j > 0$. From $\mu_j(t_j - T_j) = 0$ in (24) and $\mu_j > 0$, it can be deduced that

$$t_j = T_j. \tag{28}$$

Let $d_j = \frac{a_j}{c_j B T_j}$, $g_j = \frac{c_j B h^2}{a_j \sigma \ln 2}$. Taking t_j, d_j, g_j into (27), we have

$$\lambda_j^2 \cdot 2^{d_j \lambda_j} = g_j \gamma. \tag{29}$$

Equation (29) represents the relationship between bandwidth allocation and the γ multiplier. Changing both sides of Eq. (29) into the form of logarithmic function $\ln(x)$ and dividing by 2, we obtain

$$\ln \lambda_j + \frac{d_j \ln 2}{2} \cdot \lambda_j = \frac{\ln(g_j \gamma)}{2}. \tag{30}$$

We introduce exponential function $e^{(x)}$ and transform the Eq.(30) into the following equivalent one

$$e^{\ln \lambda_j} \cdot e^{\frac{d_j \ln 2}{2} \cdot \lambda_j} = e^{\frac{\ln(g_j \gamma)}{2}}. \tag{31}$$

After simplification, we get

$$\lambda_j \cdot e^{\frac{d_j \ln 2}{2} \cdot \lambda_j} = \sqrt{g_j \gamma}. \tag{32}$$

Then, we transform the Eq.(32) into the following forms

$$\left(\frac{d_j \ln 2}{2} \cdot \lambda_j\right) \cdot e^{\left(\frac{d_j \ln 2}{2} \cdot \lambda_j\right)} = \frac{d_j \ln 2 \sqrt{g_j \gamma}}{2}. \tag{33}$$

Finally, the expression of λ_j by using γ is obtained as

$$\lambda_j = \frac{2W\left(\frac{d_j \ln 2 \sqrt{g_j \gamma}}{2}\right)}{d_j \ln 2}, \tag{34}$$

where $W(x)$ is the principal branch of the Lambert W function defined as the solution for $W(x)e^W(x) = x$ [35].

To facilitate the search, we use the constraint $C11$ and the Eq. (16) to deduce that the maximum of λ_j is:

$$\lambda_j^l = \frac{\log_2\left(\frac{P_j^{max} h^2}{\sigma} + 1\right)}{d_j}. \tag{35}$$

According to formula (29), when $\lambda_j \geq 0$, γ and λ_j are proportional, so the range of γ can be derived from the constraint $C8$ and the Eq. (35).

Algorithm 2 Energy-efficient Offloading Scheme

Require: $\mathbf{a}$, B, n, T, $\mathbf{f} = \{f_1, ...f_j, ..., f_n\}$.

Ensure: Optimal offloading time allocation strategy $\mathbf{t}$ and bandwidth allocation strategy $\boldsymbol{\lambda}$.

1: Initialization: $\lambda_j^l = 1$, $\lambda_j^h = \frac{\log_2\left(\frac{P_j^{max} h^2}{\sigma} + 1\right)}{d_j}$, $\gamma_l = min\{\frac{\lambda_j^{l\,2} \cdot 2^{d_j \lambda_j^l}}{g_j}\}$, $\gamma_h = max\{\frac{\lambda_j^{h\,2} \cdot 2^{d_j \lambda_j^h}}{g_j}\}$, $sum_l = \sum_{j=1}^{N} 1/\lambda_j^l$, $sum_h = \sum_{j=1}^{N} 1/\lambda_j^h$, $j \in \{1, 2, ..., n\}$.
2: Based on (28), obtain $\mathbf{t} = \{T - \frac{a_j}{f_j}\}$, $j \in \{1, 2, ..., n\}$.
3: **while** $sum_l < 1$ and $sum_h > 1$ **do**
4: $\quad \gamma_m = 1/2(\gamma_h + \gamma_l)$
5: $\quad$ Update $\boldsymbol{\lambda}$ using equation (34)
6: $\quad sum_\lambda = \sum_{j=1}^{N} 1/\lambda_j$
7: $\quad$ **if** $sum_\lambda < 1$ **then**
8: $\quad\quad \gamma_h = \gamma_m$
9: $\quad\quad sum_h = sum_\lambda$
10: $\quad$ **else if** $sum_\lambda > 1$ **then**
11: $\quad\quad \gamma_l = \gamma_m$
12: $\quad\quad sum_l = sum_\lambda$
13: $\quad$ **end if**
14: **end while**
15: **return** $\boldsymbol{\lambda}$

Based on the analysis above, the energy-efficient offloading scheme is presented as Algorithm 2.

Remark 1 (Energy-efficient Offloading Scheme): According to the obtained range of λ_j and γ, initialize the maximum and minimum values of them. Then calculate the range of the total bandwidth allocated $sum_\lambda = \sum_{j=1}^{N} 1/\lambda_j$ through the range of λ_j. sum_l and sum_h represent the values of sum_λ when λ_j is minimum and maximum. When the bandwidth is underutilized or the total bandwidth allocated exceeds the available bandwidth($sum_l < 1$ or $sum_h > 1$), update γ_m, the middle value of the range of γ, expressed as $\gamma_m = 1/2(\gamma_h + \gamma_l)$. Put γ_m into Eq. (34) to calculate λ_j, Then update sum_λ. If the available bandwidth is just completely allocated($sum_\lambda = 1$), the energy consumption is minimal. Otherwise, update the range of γ and continue the iteration.

Remark 2 (Low-complexity algorithm): The traditional method for solving problem P5 is the block-coordinate descending algorithm which performs iterative optimization of the two sets of variables, $\{t_j\}$ and $\{\lambda_j\}$, resulting in high computation complexity. In contrast, the proposed solution approach, described in Algorithm 2, needs to perform only a one-dimension search for γ, reducing the computation complexity from $O(n^2)$ to $O(\log_2 n)$.

5 Simulation experiment and result analysis

In this section, the proposed resource-sharing strategy is analyzed by simulations. We consider the main factors that affect users' utility and equipment energy consumption and verify that the proposed algorithm can effectively minimize equipment energy consumption under the premise of the high utility of the user. In each group of experiments, we perform the corresponding random simulations 100 times and average the results. The experimental equipment is with a 3.40 GHz I7 Intel processor and 20 GB RAM.

5.1 Parameter settings

The computing resource sharing is model of $n(n \in \{2, 3, ..., 12\})$ CRPs and one CRR, where $J = \{1..., j, ...n\}$ denote the index set of CRPs. Referring to [21, 36–39], the parameter settings for resource trading and computation offloading are presented in Table 1.

5.2 Performance of resource trading

To validate the efficacy of the proposed trading scheme, we conduct simulations under different numbers of CRPs, and obtain the utility of users under the three resource pricing strategies: fixed low pricing ($p = 1$), fixed high pricing ($p = 3$), and the pricing of the proposed trading scheme.

As shown in Figs. 3 and 4, the low pricing makes the CRR pay fewer resource coins for completing the task, thereby improving its utility. Conversely, the utility of CRPs is reduced because low pricing leads to low income. Conversely, high pricing satisfies the utility of CRPs but ignores the utility of the CRR. The pricing strategy of the proposed trading scheme can satisfy both the utility of the CRR and the utility of CRPs in a changing trading environment.

Then we observe the impact of different resource allocation strategies on the utility of users. We conduct the resource allocation strategy of the proposed trading scheme(PTS), the strategy with the maximum transaction volume(MaxTV), and the strategy with the minimum transaction volume(MinTV).

From Fig. 5, the resource allocation strategy of the proposed trading scheme can maximize the utility of the CRR, because our strategy is determined based on the bids of the CRPs, and the CRR can get the maximum utility all the time. When the resource allocation strategy is fixed, the CRR cannot adjust the resource allocation to obtain maximum utility. From Fig. 6, it is concluded that the resource allocation strategy with MaxTV enables resource providers to obtain high utility. In resource trading, the greater the

Table 1 Parameters in resource trading

Parameter	Value	Parameter	Value
d_{min}	$\{5, 6, ..., 10\}$	κ	10^{-28}
d_{max}	$\{10, 12, ..., 20\}$	B	10 MHz
n_1	[0.005, 0.025]	T	[1, 3]s
n_2	(0, 1]	f_l^{max}	$[3 \times 10^8, 30 \times 10^8]$ cycles/s
z_j	$\{0.001, 0.002, ..., 0.009\}$	c_j	[500, 1500] cycles/bit
$\bar{p}$	[3, 3.5]	σ	10^{-9} W
		h_j	$[10^{-8.6}, 10^{-9.75}]$

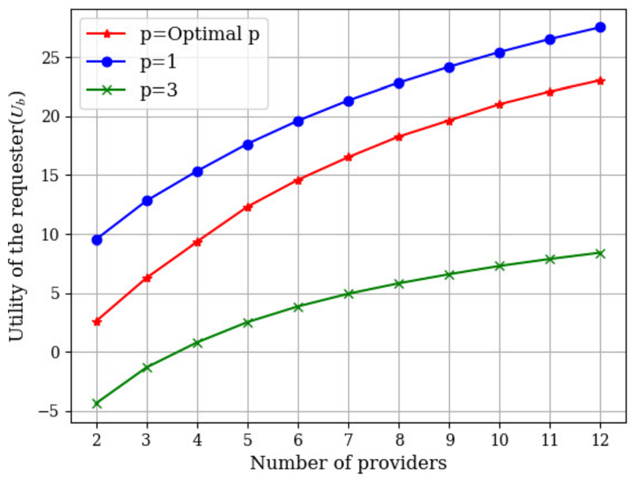

Fig. 3 The requester utility versus the different pricing schemes

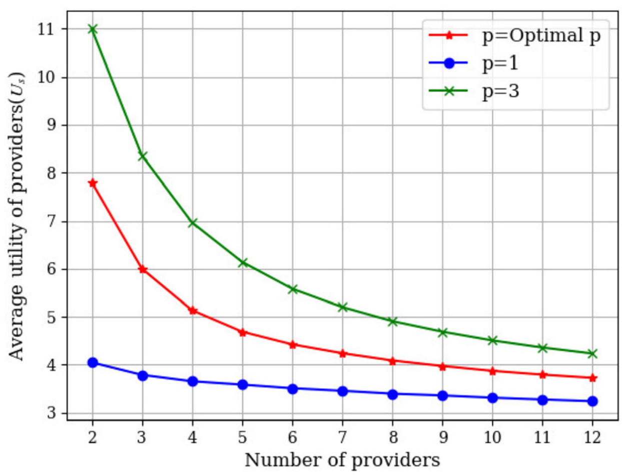

Fig. 4 The provider utility versus the different pricing schemes

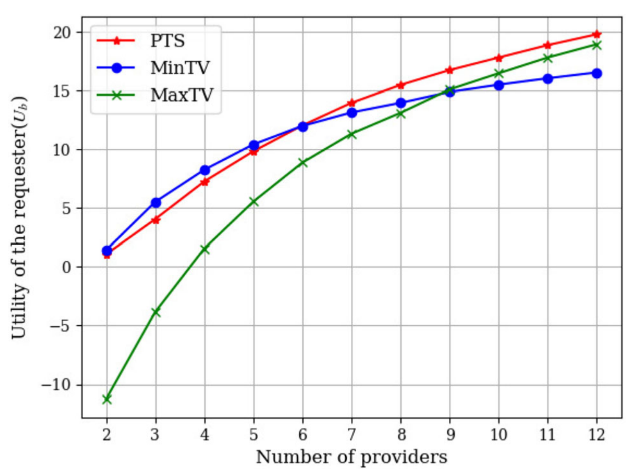

Fig. 5 The requester utility versus the different resource allocation schemes

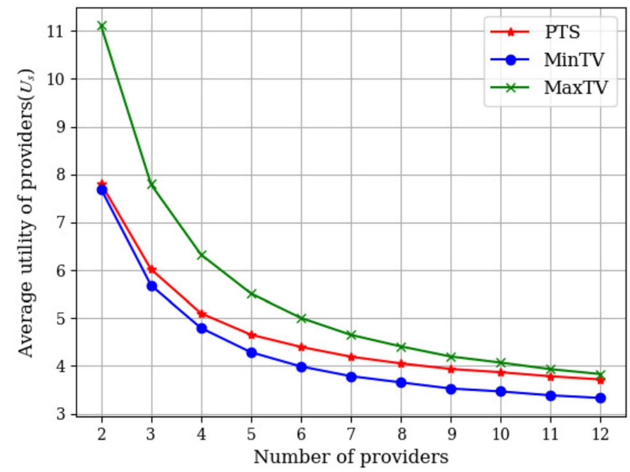

Fig. 6 The provider utility versus the different resource allocation schemes

resource demand, CRPs can sell more computing resources to obtain more revenue.

5.3 Performance of computation offloading

This group of simulations is conducted to verify the efficiency of the proposed computation offloading scheme and the performance advantage of our offloading algorithm under the premise of satisfying the utility of users.

We substitute the result of resource allocation into the energy consumption model. Then compare the energy consumption obtained through the different computation offloading strategies: the strategy of the proposed computation offloading scheme(COS), the random local calculation frequency strategy(CFS), the random offloading time allocation strategy(TAS), and the random offloading bandwidth allocation strategy(BAS).

CFS: The offloading time allocation and offloading bandwidth allocation are obtained by COS, and the local computational frequency is set to be a value randomly drawn from $\left[\frac{d_{max}-\sum_{j=1}^{N} a_j}{T}, f_{max}\right]$.

TAS: Take the local computational frequency and offloading bandwidth allocation in COS. Then set the offloading time range of provider j to $\left[\frac{\lambda_j a_j}{c_j B \cdot \log_2(\frac{P_j^{max} \cdot c^2}{\sigma}+1)}, T-\frac{a_j}{f_j}\right]$.

BAS: Take the local computational frequency and offloading time allocation in COS. Allocate the bandwidth to n CRPs randomly, and obtain the bandwidth ratio η_j of provider j. We define $\lambda_j = 1/\eta_j$ and

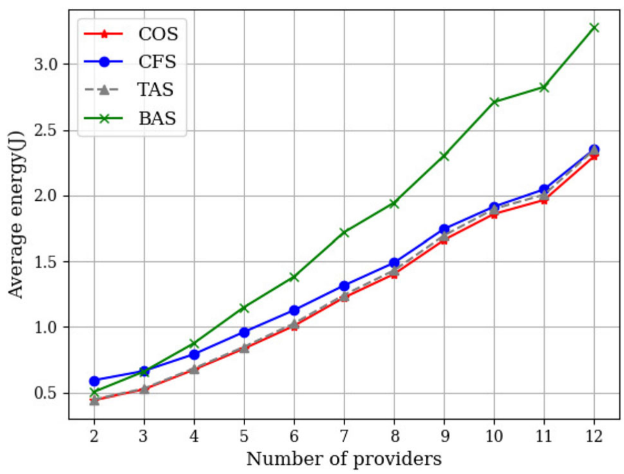

Fig. 7 Energy consumption of different computation offloading strategies

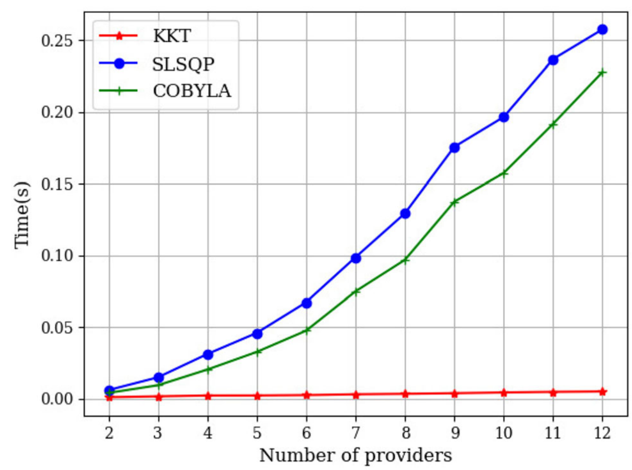

Fig. 9 The time cost of the three algorithms

$\lambda_j^{max} = \frac{c_j B t_j}{a_j} \cdot \log_2(\frac{P_j^{max} \cdot c^2}{\sigma} + 1)$. If $\lambda_j > \lambda_j^{max}$, let $\lambda_j = \lambda_j^{max}$, and finally get the energy consumption.

Figure 7 shows that the offloading strategy obtained by our algorithm always minimizes energy consumption, which proves the effectiveness of the proposed computation offloading scheme.

Then we compare the results of optimal value and time cost in the proposed computation offloading scheme with the other two traditional optimization algorithms: SLSQP [40] and COBYLA [41].

Figures 8 and 9 respectively show the growth of energy consumption and algorithm execution time as the number of providers increases. The optimal values obtained by the three algorithms are roughly similar, but our algorithm obtains the optimal solution in the shortest time. This is because we reduce the iteration variables through derivation and simplification. The computation time of our algorithm hardly increases with the increase in the number of providers. The superiority of the proposed algorithm is demonstrated through comparison with other algorithms.

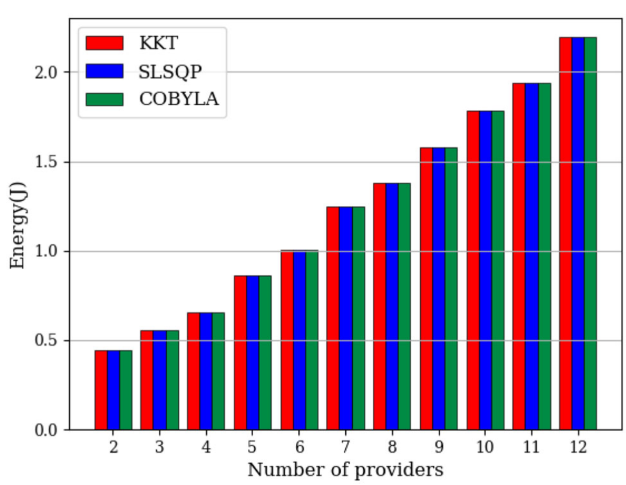

Fig. 8 The optimal energy consumption value obtained by three algorithms

5.4 Performance of joint trading and offloading

To prove the advantages of joint trading and computation offloading, we compare the optimal combination strategy with other strategies.

We investigate the utility of random resource allocation strategy(RAS), the proposed optimal combination strategy(OCS), and the low energy resource allocation strategy(ERAS) respectively. Then we compare the energy consumption in random offloading strategy(ROS), OCS, and ERAS.

RAS: Select demands from CRPs randomly under the conditions of constraints $C1$ and $C2$ to get the utility.

ERAS: After obtaining the combination strategy of trading and computation computing, optimize resource allocation strategy to minimize energy consumption. Then, calculate the utility based on the obtained resource allocation strategy.

ROS: The local computational frequency, offloading time allocation, and offloading bandwidth allocation are selected referring to CFS, TAS, and BAS in Sect. 5.3.

Figures 10 and 11 show that ERAS, which only considers reducing energy without considering utility, has lower utility for CRR and CRPs than OCS, so ERAS cannot motivate users to participate in trading. When the number of CRPs is small, ROS reduces the utility of CRR significantly, which causes CRR to be reluctant to participate in the transaction. OCS can make both CRR and CRPs have high utility with different numbers of providers. In Fig. 12, ROS, which only considers utility without energy consumption, consumes significantly more energy than OCS in computation offloading. Therefore, the

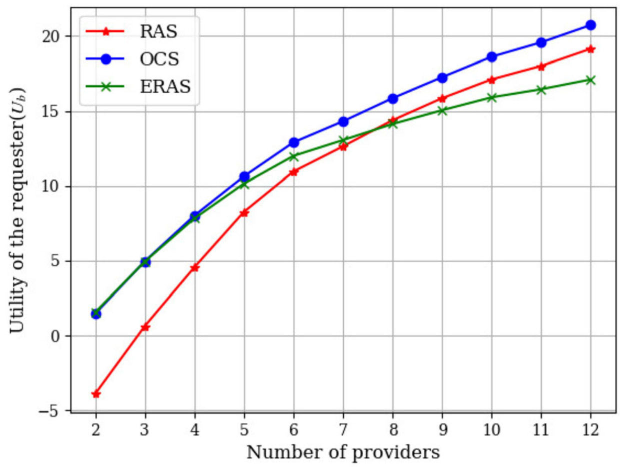

Fig. 10 The relationship between resource allocation and requester utility

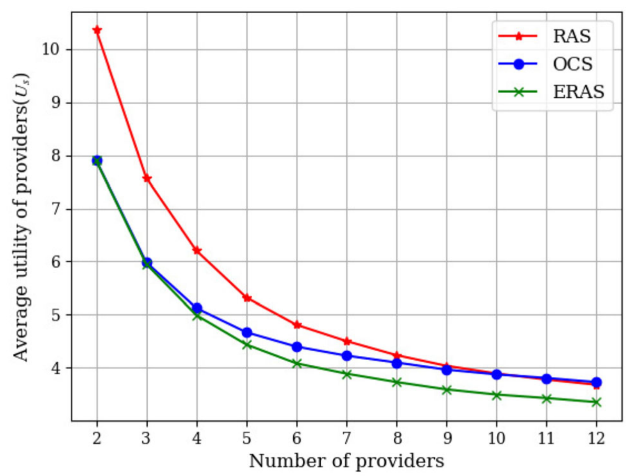

Fig. 11 The relationship between resource allocation and provider utility

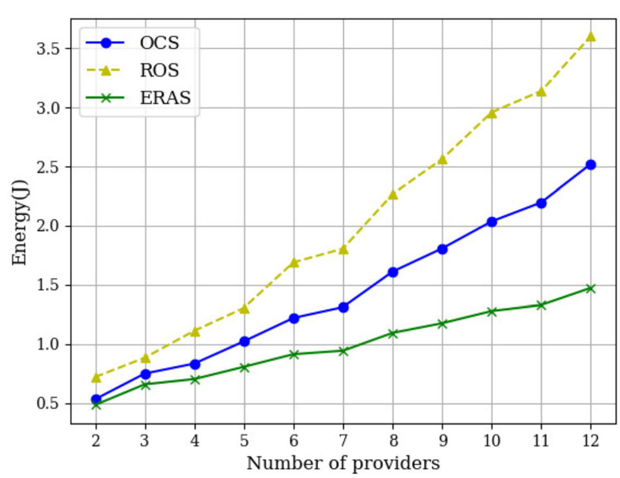

Fig. 12 The relationship between resource allocation policies and device energy consumption

combination strategy of resource trading and computation offloading OCS we proposed can satisfy the utility of both the CRR and CRPs, which motivates users to participate in the trading and implement energy-efficient computation offloading.

6 Conclusion

In this paper, we propose a joint scheme of computing resource trading and computation offloading based on blockchain in D2D-assisted MEC to motivate users to participate in resource sharing. First, to guarantee security and privacy, we introduce blockchain into the resource sharing system. Then, the problem of resource trading is modeled as a Stackelberg game. We obtain the optimal resource trading strategy to maximize the utility of CRR and CRPs by using the proposed resource trading algorithm. To achieve the minimum energy consumption under the premise of high utility, we substitute the resource allocation strategy into the energy consumption model and design an energy-efficient computation offloading scheme. Finally, by conducting the simulations, we have evaluated the performance of the proposed scheme, which motivates users to participate in resource sharing effectively. In future work, we will extend our work to the scenario of resource sharing between multiple CRRs and multiple CRPs.

Author contributions All authors contributed to the study's conception and design. Material preparation, data collection and analysis were performed by WJ, XF, PL and HS. The first draft of the manuscript was written by XF and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This research was partially supported by National Natural Science Foundation of China (32171777), Fundamental Research Funds for the Central Universities (2572017PZ04), Heilongjiang Province Applied Technology Research and Development Program Major Project (GA20A301).

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest All authors disclosed no relevant relationships.

Informed consent Written informed consent for publication of this paper was obtained from the Northeast Forestry University and all authors.

References

1. Zhang, Z., Chu, D., Chen, X., Moscibroda, T.: Mobile motion gaming: enabling a new class of phone-to-phone action games on commodity phones. IEEE Trans. Mobile Comput. **12**, 1487–1501 (2013)
2. Luo, C., Min, G., Yu, F.R., Chen, M., Yang, L.T., Leung, V.C.M.: Energy-efficient distributed relay and power control in cognitive radio cooperative communications. IEEE J. Sel. Areas Commun. **31**, 2442–2452 (2013)
3. Agiwal, M., Roy, A., Saxena, N.: Next generation 5G wireless networks: a comprehensive survey. IEEE Commun. Surv. Tutor. **18**(3), 1617–1655 (2016). https://doi.org/10.1109/COMST.2016.2532458
4. Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T.: Mobile edge computing: a survey. IEEE Internet Things J. **5**, 450–465 (2018)
5. Chen, M., Hao, Y.: Task offloading for mobile edge computing in software defined ultra-dense network. IEEE J. Sel. Areas Commun. **36**, 587–597 (2018)
6. Luo, C., Salinas, S., Li, M., Li, P.: Energy-efficient autonomic offloading in mobile edge computing. In: Proceedings of IEEE DASC'17, (Orlando, FL USA), pp. 581-588 (2017)
7. Wang, R., Yan, J., Wu, D., Wang, H., Yang, Q.: Knowledge-centric edge computing based on virtualized d2d communication systems. IEEE Commun. Mag. **56**, 32–38 (2018)
8. Wen, J., Ren, C., Sangaiah, A.K.: Energy-efficient device-to-device edge computing network: An approach offloading both traffic and computation. IEEE Commun. Mag. **56**, 96–102 (2018)
9. Qiao, G., Leng, S., Chai, H., Asadi, A., Zhang, Y.: Blockchain Empowered Resource Trading in Mobile Edge Computing and Networks. In: ICC 2019–2019 IEEE International Conference on Communications (ICC), pp. 1–6 (2019). https://doi.org/10.1109/ICC.2019.8761664.
10. Aitzhan, N.Z., Svetinovic, D.: Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. Dependable Secure Comput. **15**(5), 840–852 (2018). https://doi.org/10.1109/TDSC.2016.2616861
11. Zarrin, J., Wen Phang, H., Babu Saheer, L., et al.: Blockchain for decentralization of internet: prospects, trends, and challenges. Cluster Comput. **24**(4), 2841–2866 (2021)
12. Qiang, Q., Nurgaliev, I., Muzammal, M., Jensen, C.S., Fan, J.: On spatio-temporal blockchain query processing. Future Gener. Comput. Syst. **98**, 208–218 (2019)
13. Tseng, L., Yao, X., Otoum, S., et al.: Blockchain-based database in an IoT environment: challenges, opportunities, and analysis. Cluster Comput. **23**, 2151–2165 (2020)
14. Muzammal, Muhammad, Qiang, Qu., Nasrulin, Bulat: Renovating blockchain with distributed databases: an open source system. Future Gener. Comput. Syst. **90**, 105–117 (2019)
15. Li, H., Pei, L., Liao, D., et al.: BDDT: use blockchain to facilitate IoT data transactions. Cluster Comput. **24**, 459–473 (2021)
16. Swan, M.: Blocakchain: blueprint for a new economy. O'Reilly Media, Newton (2015)
17. Liu, Y., Yu, F.R., Li, X., Ji, H., Leung, V.C.M.: Decentralized resource allocation for video transcoding and delivery in blockchain-based system with mobile edge computing. IEEE Trans. Vehicular Technol. **68**(11), 11169–11185 (2019). https://doi.org/10.1109/TVT.2019.2937351
18. Li, M., Hu, D., Lal, C., Conti, M., Zhang, Z.: Blockchain-enabled secure energy trading with verifiable fairness in industrial internet of things. IEEE Trans. Ind. Inform. **16**(10), 6564–6574 (2020). https://doi.org/10.1109/TII.2020.2974537
19. Fernando, P., Gunawardhana, L., Rajapakshe, W., Dananjaya, M., Gamage, T., Liyanage, M.: Blockchain-Based Wi-Fi Offloading Platform for 5G. In: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6 (2020). https://doi.org/10.1109/ICCWorkshops49005.2020.9145369
20. Chen, T., Khan, A.S., Zheng, G., Lambotharan, S.: Blockchain secured auction-based user offloading in heterogeneous wireless networks. IEEE Wirel. Commun. Lett. **9**(8), 1141–1145 (2020). https://doi.org/10.1109/LWC.2020.2982634
21. Li, Z., Yang, Z., Xie, S.: Computing resource trading for edge-cloud-assisted internet of things. IEEE Trans. Ind. Inform. **15**(6), 3661–3669 (2019). https://doi.org/10.1109/TII.2019.2897364
22. Liu, M., Yu, F.R., Teng, Y., Leung, V.C.M., Song, M.: Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing. IEEE Trans. Wirel. Commun. **18**(1), 695–708 (2019). https://doi.org/10.1109/TWC.2018.2885266
23. Zhang, Z., Hong, Z., Chen, W., Zheng, Z., Chen, X.: Joint computation offloading and coin loaning for blockchain-empowered mobile-edge computing. IEEE Internet Things J. **6**(6), 9934–9950 (2019). https://doi.org/10.1109/JIOT.2019.2933445
24. Liu, K., Chen, W., Zheng, Z., Li, Z., Liang, W.: A novel debt-credit mechanism for blockchain-based data-trading in internet of vehicles. IEEE Internet Things J. **6**(5), 9098–9111 (2019). https://doi.org/10.1109/JIOT.2019.2927682
25. De Souza, A.B., et al.: Computation offloading for vehicular environments: a survey. IEEE Access **8**, 198214–198243 (2020). https://doi.org/10.1109/ACCESS.2020.3033828
26. Hassija, V., Saxena, V., Chamola, V.: A mobile data offloading framework based on a combination of blockchain and virtual voting. Softw. Pract. Exp. **51**(12), 2428–2445 (2020)
27. Hassija, V., Chamola, V., Gupta, V., Chalapathi, G.S.S.: A Blockchain based Framework for Secure Data Offloading in Tactile Internet Environment. In: 2020 International Wireless Communications and Mobile Computing (IWCMC), pp. 1836–1841 (2020). https://doi.org/10.1109/IWCMC48107.2020.9148559
28. Sheng, M., Wang, Y., Wang, X., Li, J.: Energy-efficient multiuser partial computation offloading with collaboration of terminals, radio access network, and edge server. IEEE Trans. Commun. **68**(3), 1524–1537 (2020). https://doi.org/10.1109/TCOMM.2019.2959338
29. Wen, J., Ren, C., Sangaiah, A.K.: Energy-efficient device-to-device edge computing network: an approach offloading both traffic and computation. IEEE Commun. Mag. **56**(9), 96–102 (2018). https://doi.org/10.1109/MCOM.2018.1701054
30. Chen, J., Zhao, Y., Xu, Z., Zheng, H.: Resource allocation strategy for D2D-assisted edge computing system with hybrid energy harvesting. IEEE Access **8**, 192643–192658 (2020). https://doi.org/10.1109/ACCESS.2020.3032033
31. Anoh, K., Maharjan, S., Ikpehai, A., Zhang, Y., Adebisi, B.: Energy peer-to-peer trading in virtual microgrids in smart grids: a game-theoretic approach. IEEE Trans. Smart Grid **11**(2), 1264–1275 (2020). https://doi.org/10.1109/TSG.2019.2934830
32. Xiong, Z., Kang, J., Niyato, D., Wang, P., Poor, H.V.: Cloud/edge computing service management in blockchain networks: multi-leader multi-follower game-based ADMM for pricing. IEEE Trans. Services Comput. **13**(2), 356–367 (2020). https://doi.org/10.1109/TSC.2019.2947914
33. Yao, H., Mai, T., Wang, J., Ji, Z., Jiang, C., Qian, Y.: Resource trading in blockchain-based industrial internet of things. IEEE Trans. Ind. Inform. **15**(6), 3602–3609 (2019). https://doi.org/10.1109/TII.2019.2902563
34. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)

35. Corless, R.M., Gonnet, G.H., Hare, D.E.G., Jeffrey, D.J., Knuth, D.E.: On the Lambert W function. Adv. Comput. Math. **5**(1), 329–359 (1996)
36. Wang, Y., Sheng, M., Wang, X., Wang, L., Li, J.: Mobile-edge computing: partial computation offloading using dynamic voltage scaling. IEEE Trans. Commun. **64**(10), 4268–4282 (2016)
37. You, C., Huang, K., Chae, H., Kim, B.: Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Trans. Wirel. Commun. **16**(3), 1397–1411 (2017). https://doi.org/10.1109/TWC.2016.2633522
38. Ti, N.T., Le, L.B.: Computation offloading leveraging computing resources from edge cloud and mobile peers. IEEE Int. Conf. Commun. (ICC) **2017**, 1–6 (2017). https://doi.org/10.1109/ICC.2017.7997138
39. You, C., Huang, K., Chae, H.: Energy efficient mobile cloud computing powered by wireless energy transfer. IEEE J. Select. Areas Commun. **34**(5), 1757–1771 (2016)
40. Kraft, D.: A software package for sequential quadratic programming. Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt (1988)
41. Powell, M.: A direct search optimization method that models the objective and constraint functions by linear interpolation. Adv. Opt. Numer. Anal. **275**, 51–67 (1994)

Weipeng Jing Weipeng Jing (Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology, China. He is currently an Associate Professor with Northeast Forestry University, China. He has published over 50 research articles in refereed journals and conference proceedings, such as CPC, PUC, and FGCS. His research interests include modeling and scheduling for distributed computing systems, fault tolerant computing and system reliability, cloud computing, and spatial data mining.

Xinyu Fu Xinyu Fu received the B.E. degree in information management and information system from Northeast Forestry University, China, in 2019. She is currently pursuing the M.E. degree in computer system architecture with Northeast Forestry University. Her current research interests include mobile edge computing, blockchain, and the Internet of Things.

Peng Liu College of Information and Computer Engineering, Northeast Forestry University, Harbin, China Peng Liu received the M.S. degree from the College of Computer Science and Technology, Jilin University, China, in 2016, and the Ph.D. degree, in 2019. Currently, he is a Lecturer with the College of Information and Computer Engineering, Northeast Forestry University, Harbin, China. His main research interests include mobile cloud computing, mobile edge computing, blockchain, and the Internet of Things.

Houbing Song Houbing Song (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in August 2012, and the M.S. degree in civil engineering from The University of Texas at El Paso, El Paso, TX, USA, in December 2006.,In August 2017, he joined the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, where he is currently an Assistant Professor and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Laboratory). He has served on the faculty for West Virginia University from August 2012 to August 2017. In 2007, he was an Engineering Research Associate with the Texas A &M Transportation Institute. He is the author of more than 100 articles. His research interests include cyber-physical systems, cybersecurity and privacy, the Internet of Things, edge computing, big data analytics, unmanned aircraft systems, connected vehicle, smart and connected health, and wireless communications and networking. He is also a Senior Member of ACM. He was a recipient of the prestigious Air Force Research Laboratory's Information Directorate (AFRL/RI) Visiting Faculty Research Fellowship in 2018, and the very first recipient of the Golden Bear Scholar Award and the highest campus-wide recognition for research excellence at the West Virginia University Institute of Technology (WVU Tech) in 2016. He also serves as an Associate Technical Editor for IEEE Communications Magazine. He is also the Editor of six books, including Big Data Analytics for Cyber-Physical Systems: Machine Learning for the Internet of Things (Elsevier, 2019), Smart Cities: Foundations, Principles and Applications (Hoboken, NJ, USA: Wiley, 2017), Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications (Chichester, U.K.: Wiley-IEEE Press, 2017), Cyber-Physical Systems: Foundations, Principles and Applications, (Boston, MA, USA: Academic Press, 2016), and Industrial Internet of Things: Cybermanufacturing Systems (Cham, Switzerland: Springer, 2016).