Peer-to-Peer Networking and Applications (2021) 14:1431–1445
https://doi.org/10.1007/s12083-020-01032-y

Three-stage Stackelberg game based edge computing resource management for mobile blockchain

Yuqi Fan[1] · Zhifeng Jin[1] · Guangming Shen[1] · Donghui Hu[1] · Lei Shi[1] · Xiaohui Yuan[2]

Received: 29 July 2020 / Accepted: 12 November 2020 / Published online: 5 January 2021

Abstract

Blockchain can provide a dependable environment for mobile applications. Mining, as an important component in blockchain, requires a lot of computing resources, and hence resource limited moblie devices are unable to perform the mining. Offloading mining computation tasks to an edge computing service provider (ESP) or a cloud computing service provider (CSP) is considered as a feasible solution to mobile blockchain mining. However, the computing resources of the ESP are not unlimited. Therefore, rational edge computing resource management is critical to maximizing the utilities of the ESP and the miners. Most of the existing work assumes the computation is offloaded to either the CSP or the ESP which serves the terminal devices. However, an ESP can also offload the computation to the other ESPs, when the ESP is overloaded. In this paper, we construct a computation offloading model composed of multiple miners, multiple ESPs, and a CSP, where an overloaded ESP can offload the mining tasks to the CSP or the other ESPs or both. We propose a three-stage Stackelberg game for optimal pricing-based edge computing resource management. We analyze the existence and uniqueness of Stackelberg game equilibrium and derive the optimal amount of computing resource requests from the miners. We then propose a simple yet effective golden section based Stackelberg game equilibrium searching algorithm SES for resource pricing. We conduct experiments through simulations. Experimental results show that the proposed computing offloading model and algorithm can achieve high unit service utilities of both the ESPs and the terminal devices.

Keywords Edge computing · Mobile blockchain mining · Resource management · Stackelberg game

1 Introduction

In recent years, blockchain has been receiving extensive attention from industry and academia. Blockchain has been increasingly used in areas such as Bitcoin, financial services, Internet of Things (IoT), smart grids power systems, etc. The annual revenue of the enterprise applications of blockchain is estimated to increase to approximately $19.9 billion by 2025 [1].

Initially, blockchain was designed as a distributed shared ledger, and the data in the blockchain network can be saved and accessed by any node in the blockchain network. Consensus is the core of the blockchain. In the blockchain system, each node must make its ledger consistent with the ledger of other nodes through the consensus. The consensus algorithm guarantees the consistency and correctness of each transaction on all the nodes and enables the blockchain for efficient collaborative work on a large scale without relying on a centralized third party. Some consensus algorithms, such as Proof-of-Work (PoW), require a large amount of computation. Users (miners) win rewards through mining, where the users need to solve a computationally challenging problem. The first miner who successfully solves the computation problem and reaches an agreement with other miners is considered as the winner of the competition, and the winner will receive a reward for successful mining.

Blockchain can provide a dependable environment for mobile applications [2, 3]. However, the computing resources required to solve the compute-intensive mining

This article belongs to the Topical Collection: *Special Issue on Privacy-Preserving Computing*
Guest Editors: Kaiping Xue, Zhe Liu, Haojin Zhu, Miao Pan and David S.L. Wei

This work was partly supported by the National Natural Science Foundation of China under Grant U1836102.

✉ Yuqi Fan
yuqi.fan@hfut.edu.cn

Extended author information available on the last page of the article.

problem are prohibitively high for mobile devices [4]. Edge computing is considered as a viable solution to blockchain mining in mobile environments [5], where the terminal devices offload the mining tasks to an edge computing service provider (ESP). The ESP makes profits by allocating the computing resources for the mining tasks offloaded by the terminal devices. Nevertheless, the computing resources of the ESP are not unlimited and hence are competed by the terminal devices. Therefore, careful resource management is critical for the ESP and terminal devices to maximize their utilities.

Most of the existing work assumes the computation is offloaded to the CSP or the ESP which is serving the terminal devices. However, an ESP can also offload the computation to the other ESPs when the ESP is overloaded. In this paper, we deal with the problem of resource management of the ESPs for mobile blochain, assuming an overloaded ESP can offload the mining tasks to the CSP or the other ESPs or both. The main contributions of this paper are as follows:

1) We construct a computation offloading model consisting of a CSP, multiple ESPs, and multiple mobile miners/end devices. When an ESP has no enough capacity to process the service requests from the terminal devices, the mining tasks of the terminal devices can be offloaded to the CSP or the other ESPs or both.
2) We design the utility functions of the miners, ESPs and CSP to capture the intrinsic relationship among them. We formulate a three-stage Stackelberg game for optimal pricing-based edge computing resource management. In the first stage, the CSP decides the resource price. In the second stage, each ESP decides the resource price. In the third stage, the miners determine the amount of resources to purchase. We prove the existence and uniqueness of the Stackelberg game equilibrium and derive the miners' optimal amount of computing resources to purchase. We then propose an efficient golden section based Stackelberg game equilibrium searching algorithm SES for resource pricing.
3) We conduct simulations to evaluate the performance of the proposed model and algorithm. Simulation results demonstrate the proposed model and algorithm can achieve high unit service utilities of the ESPs and the terminal devices.

The rest of the paper is organized as follows. Section 2 introduces the related work. The system model is formulated and analyzed in Section 3. Section 4 presents the proposed golden section based Stackelberg game equilibrium searching algorithm. The simulations are given in Section 5, and Section 6 concludes the paper.

2 Related work

Blockchain has the characteristics of security, reliability, immutability, decentralization, etc., and it can provide a reliable environment for IoT devices. For example, Roberto et al. proposed an architecture based on blockchain and edge computing to improve the quality of IoT data and false data detection [6]. Chamarajnagar et al. designed a decentralized architecture using blockchain technology, in order to promote the distributed collaboration among mobile IoT devices to share their services and redundant computing resources [7]. In vehicle edge networks, Kang et al. proposed a reputation-based data sharing scheme which introduces consortium blockchain and smart contract technology to implement secure data storage and prevent data sharing without authorization [8]. Kim et al. proposed an edge computing architecture based on blockchain technology to ensure the availability, scalability, and integrity of edge computing; blockchain structure and protocols were modified to support the execution of complex programs [9].

Some research on resource management for mobile blockchain mining has been conducted. Kroll et al. proposed a game model consisting of miners for the mining process of blockchain, and each miner made a decision on which branch of the blockchain to mine on [10]. Sompolinsky et al. proposed a cooperative game model to solve the mining pool problem; in this model, the miners form an alliance to finish the computing power accumulation and share stable reward [11]. Houy organized the mining process as a speed game between miners with different computational powers, and analytically found the Nash equilibrium in the two-player case [12]. Xiong et al. adopted a two-stage Stackelberg game to jointly maximize the profit of the ESP and the individual utilities of different miners [13]. Zhang et al. proposed a joint optimization framework of Fog nodes (FNs), data service operators (DSOs), and data service subscribers (DSSs), which implemented the optimal resource allocation scheme in a distributed manner [14]. Chiu et al. formalized the PoW protocol into a Cournot game in which users compete to update the blockchain for rewards [15]. Luong proposed an optimal auction based on deep learning for the edge resource allocation, which used valuations of the miners as the training data to adjust the parameters of the neural networks [16]. Jiao et al. constructed an auction-based market model which achieved an efficient allocation of computing resources [17]. Liu et al. modeled the joint optimization problem of mining task offloading and block cryptographic hash cache, and proposed an alternating direction multiplier method for the problem [18]. Wu et al. proposed efficient distributed algorithms for the mobile terminals to individually determine their optimal computational resources acquired from different edge servers, so as to maximize the total net return of the mobile terminals while maintaining the fairness between mobile terminals [19]. Fan et al. constructed a computation offloading model consisting of multiple miners, an ESP, and a CSP, where the ESP and the CSP are independent of each other; a Stackelberg game was formulated with the ESP as the leader and the miners as the followers for optimal pricing based edge computing resource management [20].

Most of the existing work assumes the computation is offloaded to the CSP or the ESP which is serving the terminal devices. However, an ESP can also offload the computation to the other ESPs, when the ESP is overloaded. In this paper, we study the cloud/edge computing resource allocation and pricing problem for mobile blockchain under the computing offloading framework consisting of a cloud, multiple edge servers, and multiple users.

3 The system model

Figure 1 illustrates the system model which is divided into three layers. The bottom layer consists of multiple miners, the middle layer includes multiple ESPs, and the top layer is a CSP. Each terminal device conducts mining for reward, and the resource-limited terminal devices offload the mining tasks to the ESPs who make profits by providing the computing resources to the miners. The CSP typically has enough processing capacity to execute the mining tasks, while the miners will experience an unpredictable Internet communication latency with the CSP, which restricts the performance of running the tasks at the CSP. The ESPs are close to the miners so that the network latency between the ESPs and the terminal devices is low. However, due to the limited computing resources of ESPs, when an ESP has no computing resources to accommodate all the service requests, the ESP has three modes to offload the tasks. The first is to offload the mining tasks to the other ESPs, the second is to send the mining tasks to the CSP, and the third is to offload the mining tasks to both the other ESPs and the CSP. When an ESP is not overloaded, the ESP can provide its computing resources to the terminal devices or the other ESPs. The symbols and notations used in the paper are listed in Table 1.

Table 1 Symbols and notations

Attributes	Description
B	the set of miners/terminal devices
K	the number of ESPs
e_n	the n-th ESP
M_n	the number of miners under ESP e_n
$b_{n,i}$	the i-th miner served by ESP e_n
$q_{n,i}$	the amount of services/computing resources purchased miner $b_{n,i}$
$Q_{n,i}$	the maximum amount of services/computing resources that $b_{n,i}$ will purchase
w	the unit service reward of miners
p_n^c	the unit service price of the CSP corresponding to ESP e_n
p_n^e	the unit service price of ESP e_n
C_n	the computing capacity of ESP e_n
R^c	the transfer rate of between the ESP and the CSP
R^e	the transfer rate of between the miner and the ESP
f^c	the CSP's computing power
f_n^e	the computing power of ESP e_n
τ	the unit time cost of service provisioning
α	the unit service provisioning cost of the ESPs
$\beta_{n,i}$	the weight of service response time for miner $b_{n,i}$
δ_n	indicating the fraction of excessive tasks at ESP e_n to be offloaded to the other ESPs
θ_n	indicating whether ESP e_n is overloaded (=1) or not (=0)
μ_n	the amount of computing resources sold to the other ESPs by ESP e_n
$E_{n,i}$	the utility of miner $b_{n,i}$
E_n	the utility of ESP e_n
E_n^c	the utility of the cloud gained from ESP e_n

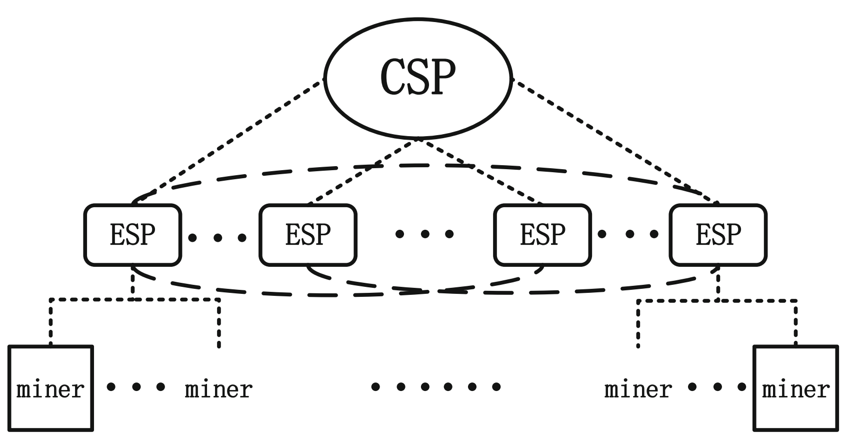


Fig. 1 System model

3.1 Analysis of the offloading mode

An overloaded ESP has three modes to execute the tasks that are beyond the computing capaicity of the ESP. One is to send the mining tasks to the CSP (mode 0), and the second is to offload the mining tasks to the other ESPs (mode 1), and the third is to offload the mining tasks to the other ESPs and the CSP (mode 2). We define the excessive tasks of ESP e_n as

$$A = \sum_{i=1}^{M_n} q_{n,i} - C_n \tag{1}$$

3.1.1 Offloading to the CSP (mode 0)

In this mode, the overloaded ESP offloads the excessive mining tasks to the CSP. The delay with mode 0 is

$$t^c(A) = t_U^c(A) + t_E^c(A) = \frac{\sum_{i=1}^{M_n} q_{n,i} - C_n}{R^c} + \frac{(\sum_{i=1}^{M_n} q_{n,i} - C_n)\gamma(A)}{f^c} \tag{2}$$

where $t_U^c(A)$, $t_E^c(A)$, and $t^c(A)$ are the transfer delay between the ESP and the cloud, the execution time at the cloud, and the total delay of offloading A to the cloud, respectively; $\gamma(A)$ is the unit workload of A.

3.1.2 Offloading to the other ESPs (mode 1)

In this mode, the overloaded ESP offloads the excessive mining tasks to the other ESPs. The delay with mode 1 is

$$\begin{aligned} t^e(A) &= \max_{n=1..K} \rho_n (t_U^e(A,n) + t_E^e(A,n)) \\ &= \max_{n=1..K} \rho_n \left(\frac{\sum_{i=1}^{M_n} q_{n,i} - C_n}{R^e} + \frac{(\sum_{i=1}^{M_n} q_{n,i} - C_n)\gamma(A)}{f_n^e} \right) \end{aligned} \tag{3}$$

where $t_U^e(A,n)$, $t_E^e(A,n)$, and $t^e(A)$ are the transfer delay between the ESPs, the execution time at ESP e_n, and the total delay of offloading A to the other ESPs, respectively; $\rho_n \in [0,1] (n \in \{1,2,\ldots,K\})$ is the ratio of the tasks offloaded to ESP e_n to A, and $\sum_{n=1}^{K} \rho_n = 1$.

3.1.3 Offloading to both the CSP and the other ESPs (mode 2)

In this mode, the overloaded ESP offloads the excessive mining tasks to the CSP and the other ESPs. The delay with mode 2 is

$$t(A) = (1-\delta_n)t^c(A) + \delta_n t^e(A) \tag{4}$$

where $t^c(A)$ and $t^e(A)$ are the transfer delay between the ESP and the cloud, and the delay between the ESPs, respectively.

3.2 Resource allocation and pricing of miners, ESPs and CSP

ESP's unit service price p_n^e has an important impact on the ESP's utility. A higher p_n^e leads to a higher ESP's utility for a given amount of services purchased by the miners. A low p_n^e will attract the miners to purchase a large amount of computing resources. Note that service provisioning also imposes cost on the ESP, where the cost includes power, equipment loss, etc. As the total amount of computing resources sold increases, the ESP's cost will also increase. If the total demand of computing resources from the miners exceed the ESP's computing capacity, the ESP has to purchase computing resources from the CSP at price p_n^c or from another ESP e_k at price $p_k^e (k \neq n)$, which brings computing resource purchase cost to the ESP. If $p_n^e < p_n^c$, the ESP will experience a loss by providing computing resources to the miners, and hence the ESP is not willing to provide services to the terminal devices. Therefore, we set $p_n^e \leq p_n^c$.

The utility of ESP e_n, E_n, is defined as follows:

$$\begin{aligned} E_n = {} & p_n^e \sum_{i=1}^{M_n} q_{n,i} + (1-\theta_n)\mu_n p_n^e - \alpha \sum_{i=1}^{M_n} q_{n,i} \\ & -\theta_n \Bigg\{ (1-\delta_n)[p_n^c A + \tau t^c(A)] \\ & + \delta_n \left[\sum_{k=1}^{K} \rho(k) A p_k^e + \tau t^e(A) \right] \Bigg\} \end{aligned} \tag{5}$$

The objective of the ESP is to maximize

$$E_n \tag{6}$$

subject to:

$$0 \leq p_n^e \leq p_n^c \tag{7}$$

The utility of miner $b_{n,i}$ is determined by ESP's unit service price p_n^e and the amount of computing resources purchased by the ESP. If the total number of service requests exceeds the ESP's computing capacity, the ESP needs to run the offloading tasks at the cloud or the other ESPs, which increases the latency of executing mining tasks and degrades the performance of task execution. The utility of miner $b_{n,i}$, $E_{n,i}$, is defined as follows:

$$E_{n,i} = wq_{n,i} - p_n^e q_{n,i} - \frac{\beta_{n,i} q_{n,i}^2}{C_n} \tag{8}$$

The objective of each miner $b_{n,i}$ is to maximize

$$E_{n,i} = wq_{n,i} - p_n^e q_{n,i} - \frac{\beta_{n,i} q_{n,i}^2}{C_n} \tag{9}$$

subject to:

$$0 \leq q_{n,i} \leq Q_{n,i} \tag{10}$$

We regard the CSP as being composed of different small parts, each of which plays a game with an ESP. The CSP's revenue comes from the computing resources purchased by the ESPs. The CSP's utility is defined as:

$$E_n^c = (1 - \delta_n) \left[p_n^c \left(\sum_{i=1}^{M_n} q_{n,i} - C_n \right) \right] \tag{11}$$

The objective of the CSP is to maximize E_n^c. During the interaction between the ESP and the miners, the ESP acts first to set unit service price p_n^e, and then the miners respond to the price by deciding the amount of computing resources to purchase. Therefore, the interaction between the ESP and the users can be formalized as a Stackelberg game with a single leader and multiple followers, where the leader is the ESP and each follower is a miner. During the interaction between the ESP and the CSP, the CSP acts first to set unit service price p_n^c, and then the ESP responds to the price based on whether it is overloaded and whether it offloads the tasks to the CSP. Therefore, the interaction between the ESP and the CSP can also be formalized as a Stackelberg game, where the leader is CSP and each follower is an ESP.

4 Resource management based on golden section search

In this section, we first prove the existence and uniqueness of equilibrium of the Stackelberg game between the ESP and the miners, and between the ESP and the CSP, respectively. We then derive the optimal amount of computing resources to be purchased by the miners. We also propose a Stackelberg game equilibrium search algorithm based on the golden section search (SES) for resource pricing.

4.1 Analysis of the Stackelberg game

Lemma 1 *There is a unique equilibrium in miner sub-game.*

Proof During miner sub-game, i.e. the third phase of the Stackelberg game, each miner $b_{n,i}$ determines $q_{n,i}$, the amount of computing resources to purchase, with the goal of maximizing the utility at given resource price p_n^e. The miner's utility function defined in Eq. 8 is continuous, and the second derivative of the function is calculated as follows:

$$\frac{\partial^2 E_{n,i}}{\partial q_{n,i}^2} = -\frac{2\beta_{n,i}}{C_n} \tag{12}$$

□

We can get $\frac{\partial^2 E_{n,i}}{\partial q_{n,i}^2} \leq 0$, since $\beta_{n,i} \geq 0$ and $C_n > 0$. Therefore, miner's utility $E_{n,i}$ is a strict concave function of variable $q_{n,i}$, and there exists a unique equilibrium in miner sub-game.

Lemma 2 *At given computing resource price p_n^e, the optimal amount of computing resources purchased by miner $b_{n,i}$ is calculated as*

$$q_{n,i}^* = \min\left(\frac{(w - p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i} \right) \tag{13}$$

Proof At a given p_n^e, miner $b_{n,i}$ decides $q_{n,i}$ by making the first derivative of Eq. 8 equal to 0 as Eq. 14.

$$\frac{\partial E_{n,i}}{\partial q_{n,i}} = w - p_n^e - \frac{2\beta_{n,i}}{C_n} q_{n,i} = 0. \tag{14}$$

□

Note that $q_{n,i} \leq Q_{n,i}$. Therefore, the lemma is proven.

Theorem 1 *The unique equilibrium exists in the Stackelberg game between the ESP and the miners.*

Proof According to Lemma 1, there is a unique equilibrium in the third phase of the Stackelberg game. Next, we consider the second phase of the Stackelberg game during which p_n^e is determined by the ESP. We can recalculate ESP utility E_n based on $q_{n,i}^*$ obtained by Lemma 2 as Eq. 15.

$$
\begin{aligned}
E_n = {} & p_n^e \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) + (1-\theta_n)\mu_n p_n^e \\
& -\alpha \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) \\
& -\theta_n \Bigg\{ (1-\delta_n) \Bigg[p_n^c \left(\sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n\right) \\
& +\tau \left(\frac{\sum_{i=0}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n}{R^c} + \frac{\left(\sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n\right)\gamma(A)}{f^c} \right) \Bigg] \\
& +\delta_n \Bigg[\sum_{k=0}^{K} \rho(k) \left(\sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n\right) p_k^e + \tau \max_{k \neq n} \rho(k) \\
& \left(\frac{\sum_{i=0}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n}{R^e} + \frac{\left(\sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n\right)\gamma(A)}{f_n^e} \right) \Bigg] \Bigg\}
\end{aligned}
\tag{15}
$$

□

Next we discuss different cases of whether the ESP is overloaded.

(1) Case 1: $\theta_n = 0$; that is, the ESP is not overloaded.

The utility of ESP e_n becomes:

$$
\begin{aligned}
E_n = {} & p_n^e \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) \\
& +\mu_n p_n^e - \alpha \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right)
\end{aligned}
\tag{16}
$$

The second derivative of the function is calculated via Eq. 17.

$$
\frac{\partial^2 E_n}{\partial {p_n^e}^2} = \sum_{i=1}^{M_n} \begin{cases} 0, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} \geq Q_{n,i} \\ -\frac{C_n}{\beta_{n,i}}, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} < Q_{n,i} \end{cases}
\tag{17}
$$

We can get $\frac{\delta^2 E_n}{\delta P_{n,e}^2} \leq 0$, since $w > 0$, $\beta_{n,i} \geq 0$, and $C_n > 0$.

(2) Case 2: $\theta_n = 1$; that is, the ESP is overloaded.

In this case, the ESP has three modes to offload the excessive tasks.

Mode 0: the ESP chooses to offload the excessive tasks to the CSP. The utility of the ESP becomes:

$$
\begin{aligned}
E_n = {} & p_n^e \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - \alpha \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) \\
& - \Bigg\{ \Bigg[p_n^c \left(\sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n\right) \\
& +\tau \left(\frac{\sum_{i=0}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n}{R^c} + \frac{(\sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n)\gamma(A)}{f^c} \right) \Bigg] \Bigg\}
\end{aligned}
\tag{18}
$$

The first derivative of the function is calculated via Eq. 19.

$$\frac{\partial E_n}{\partial p_n^e} = \sum_{i=1}^{M_n} \min\left(\frac{(w-2p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - \alpha \sum_{i=1}^{M_n} \min\left(-\frac{C_n}{2\beta_{n,i}}, 0\right) - \left[p_n^c \sum_{i=1}^{M_n} \min\left(-\frac{C_n}{2\beta_{n,i}}, 0\right) + \tau\left(\frac{\sum_{i=1}^{M_n}\min\left(-\frac{C_n}{2\beta_{n,i}},0\right)}{R^c} + \frac{\left(\sum_{i=1}^{M_n}\min\left(-\frac{C_n}{2\beta_{n,i}},0\right)\right)\gamma(A)}{f^c}\right)\right] \tag{19}$$

The second derivative of the function is calculated via Eq. 20.

$$\frac{\partial^2 E_n}{\partial {p_n^e}^2} = \sum_{i=1}^{M_n} \begin{cases} 0, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} \geq Q_{n,i} \\ -\frac{C_n}{\beta_{n,i}}, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} < Q_{n,i} \end{cases} \tag{20}$$

We can get $\frac{\delta^2 E_n}{\delta P_{n,e}^2} \leq 0$, since $w > 0$, $\beta_{n,i} \geq 0$, and $C_n > 0$.

Mode 1: The ESP chooses to offload the excessive tasks to the other ESPs. The utility function of the ESP becomes:

$$E_n = p_n^e \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - \alpha \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - \left\{\left[\sum_{k=0}^{K} \rho(k)\left(\sum_{i=1}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n\right)p_k^e + \tau \max_{k\neq n} \rho_k \left(\frac{\sum_{i=1}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n}{R^e} + \frac{(\sum_{i=1}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n)\gamma(A)}{f_n^e}\right)\right]\right\} \tag{21}$$

The second derivative of the function is calculated via Eq. 22.

$$\frac{\partial^2 E_n}{\partial {p_n^e}^2} = \sum_{i=1}^{M_n} \begin{cases} 0, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} \geq Q_{n,i} \\ -\frac{C_n}{\beta_{n,i}}, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} < Q_{n,i} \end{cases} \tag{22}$$

We can get $\frac{\delta^2 E_n}{\delta P_{n,e}^2} \leq 0$, since $w > 0$, $\beta_{n,i} \geq 0$, and $C_n > 0$.

Mode 2: The ESP offloads the excessive tasks to both the CSP and the other ESPs. The utility function of the ESP becomes:

$$E_n = p_n^e \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - \alpha \sum_{i=1}^{M_n} \min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - \left\{(1-\delta_n)\left[p_n^c(\sum_{i=1}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n) + \tau\left(\frac{\sum_{i=0}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n}{R^c} + \frac{(\sum_{i=1}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n)\gamma(A)}{f^c}\right)\right] + \delta_n\left[\sum_{k=0}^{K}\rho(k)\left(\sum_{i=1}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n\right)p_k^e + \tau\max_{k\neq n}\rho(k) \left(\frac{\sum_{i=0}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n}{R^e} + \frac{(\sum_{i=1}^{M_n}\min\left(\frac{(w-p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - C_n)\gamma(A)}{f_n^e}\right)\right]\right\} \tag{23}$$

The second derivative of the function is calculated via Eq. 24.

$$\frac{\partial^2 E_n}{\partial {p_n^e}^2} = \sum_{i=1}^{M_n} \begin{cases} 0, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} \geq Q_{n,i} \\ -\frac{C_n}{\beta_{n,i}}, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} < Q_{n,i} \end{cases} \tag{24}$$

We can get $\frac{\delta^2 E_n}{\delta P_{n,e}^2} \leq 0$, since $w > 0$, $\beta_{n,i} \geq 0$, and $C_n > 0$. Therefore, E_n is a strict concave function of p_n^e $(0 \leq p_n^e \leq p_n^c)$. That is, the ESP can find the optimal resource price p_n^e to maximize the ESP's utility, and the Stackelberg game has a unique equilibrium.

Theorem 2 *The unique equilibrium exists in the Stackelberg game between the ESP and the CSP.*

Proof According to Lemma 1 and Theorem 1, there is a unique equilibrium in the second and the third phases of the Stackelberg game. Next, we consider the first phase of the Stackelberg game during which p_n^c is determined by the CSP. Only when the ESP is overloaded and the task is offloaded to the CSP, it can benefit from the service provisioning.

In other cases, the CSP's utility is 0. Therefore, we can recalculate CSP's utility E_n^c as Eq. 25. □

$$E_n^c = \begin{cases} p_n^c\left(\sum_{i=1}^{M_n} q_{n,i} - C_n\right), & 0 \le \delta_n < 1 \,\&\, \theta_n = 1 \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

When the ESP does not choose to offload the excessive tasks to the CSP, the utility of CSP is 0, and the second derivative of E_n^c must be equal to 0. Therefore, we only consider the case that the tasks are offloaded to the CSP. We can get the CSP's utility E_n^c as Eq. 26.

$$E_n^c = p_n^c\left(\sum_{i=1}^{M_n} q_{n,i} - C_n\right) \tag{26}$$

According to Eq. 13, we can get the CSP's utility function as:

$$E_n^c = \begin{cases} p_n^c\left(\sum_{i=1}^{M_n} Q_{n,i} - C_n\right), & \frac{(w-p_n^e)C_n}{2\beta_{n,i}} \ge Q_{n,i} \\ p_n^c\left(\sum_{i=1}^{M_n} \frac{(w-p_n^e)C_n}{2\beta_{n,i}} - C_n\right), & \frac{(w-p_n^e)C_n}{2\beta_{n,i}} < Q_{n,i} \end{cases} \tag{27}$$

At a given p_n^e, the CSP decides p_n^c by making the first derivative of Eq. 19 equal to 0.

$$\begin{aligned}\frac{\partial E_n}{\partial p_n^e} &= \sum_{i=1}^{M_n} \min\left(\frac{(w-2p_n^e)C_n}{2\beta_{n,i}}, Q_{n,i}\right) - \alpha \sum_{i=1}^{M_n} \min\left(-\frac{C_n}{2\beta_{n,i}}, 0\right) \\ &- \left[p_n^c \sum_{i=1}^{M_n} \min\left(-\frac{C_n}{2\beta_{n,i}}, 0\right)\right. \\ &\left. + \tau\left(\frac{\sum_{i=1}^{M_n} \min\left(-\frac{C_n}{2\beta_{n,i}}, 0\right)}{R^c} + \frac{\left(\sum_{i=1}^{M_n} \min\left(-\frac{C_n}{2\beta_{n,i}}, 0\right)\right)\gamma(A)}{f^c}\right)\right] = 0\end{aligned} \tag{28}$$

After getting p_n^e, we put p_n^e into Eq. 27 and then get the CSP's utility function as:

$$E_n^c = \begin{cases} p_n^c\left(\sum_{i=1}^{M_n} Q_{n,i} - C_n\right), & \frac{(w-p_n^e)C_n}{2\beta_{n,i}} \ge Q_{n,i} \\ p_n^c\left\{\left[\sum_{i=1}^{M_n} \frac{wC_n}{2\beta_{n,i}} + \frac{\alpha}{2}\sum_{i=1}^{M_n}\left(-\frac{C_n}{2\beta_{n,i}}\right) + \frac{p_n^c}{2}\sum_{i=1}^{M_n}\left(-\frac{C_n}{2\beta_{n,i}}\right)\right.\right. \\ \quad + \frac{\tau\sum_{i=1}^{M_n}\left(-\frac{C_n}{2\beta_{n,i}}\right)}{2R^c} + \frac{\tau\gamma(A)\sum_{i=1}^{M_n}\left(-\frac{C_n}{2\beta_{n,i}}\right)}{f^c} \\ \quad \left.\left. + \sum_{i=1}^{M_n}\left(-\frac{wC_n}{4\beta_{n,i}}\right)\right] - C_n\right\}, & \frac{(w-p_n^e)C_n}{2\beta_{n,i}} < Q_{n,i} \end{cases} \tag{29}$$

The first derivative of the function is calculated via Eq. 30.

$$\frac{\partial E_n^c}{\partial p_n^c} = \begin{cases} \sum_{i=1}^{M_n} Q_{n,i} - C_n, & \frac{(w-p_n^e)C_n}{2\beta_{n,i}} \ge Q_{n,i} \\ \left[\sum_{i=1}^{M_n} \frac{wC_n}{2\beta_{n,i}} + \frac{\alpha}{2}\sum_{i=1}^{M_n}\left(-\frac{C_n}{2\beta_{n,i}}\right) + p_n^c\sum_{i=1}^{M_n}\left(-\frac{C_n}{2\beta_{n,i}}\right)\right. \\ \quad + \frac{\tau\sum_{i=1}^{M_n}\left(-\frac{C_n}{2\beta_{n,i}}\right)}{2R^c} + \frac{\tau\gamma(A)\sum_{i=1}^{M_n}\left(-\frac{C_n}{2\beta_{n,i}}\right)}{f^c} \\ \quad \left. + \sum_{i=1}^{M_n}\left(-\frac{wC_n}{4\beta_{n,i}}\right)\right] - C_n, & \frac{(w-p_n^e)C_n}{2\beta_{n,i}} < Q_{n,i} \end{cases} \tag{30}$$

The second derivative of the function is calculated via Eq. 31.

$$\frac{\partial^2 E_n^c}{\partial {p_n^c}^2} = \sum_{i=1}^{M_n} \begin{cases} 0, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} \ge Q_{n,i} \\ -\frac{C_n}{2\beta_{n,i}}, & \frac{(w-2p_n^e)C_n}{2\beta_{n,i}} < Q_{n,i} \end{cases} \tag{31}$$

We can get $\frac{\delta^2 E_n^c}{\delta P_{n,c}^2} \le 0$, since $w > 0$, $\beta_{n,i} \ge 0$, and $C_n > 0$. Therefore, E_n^c is a strict concave function. That is, the CSP can find the optimal p_n^c to maximize the CSP's utility, and the Stackelberg game has a unique equilibrium.

4.2 Stackelberg game equilibrium search algorithm

We consider the sequential decision process of all users, the ESPs, and the CSP. First, the CSP decides the resource price. Second, the ESP plays a game with the users to decide the ESP's resource price p_n^e. All the users will decide the amount of computing resources to purchase based on the observed resource price of the ESP. If the ESP is overloaded and offloads the excessive tasks to the CSP, it will adjust the resource price p_n^c based on the ESP's resource purchase amount decision to increase the CSP's utility. The ESP then plays the game with the users.

The miners can calculate the optimal amount of computing resources to purchase with Eq. 13 when the ESP determines computing resource price p_n^e, so we need to find the optimal p_n^e to maximize the ESP's utility. The equilibrium prediction of the Stackelberg sub-game between the ESP and the miners in this paper is a one-dimensional single-peak concave function extreme value searching problem, since the range of p_n^e is $[0, p_n^c]$. Similarly, the ESP can calculate the optimal ESP's utility with Eq. 11 when the CSP determines computing resource price p_n^c, and the equilibrium prediction of the Stackelberg sub-game between the ESP and the CSP is also a one-dimensional single-peak concave function extreme value searching problem.

The analysis of the Stackelberg game shows that resource pricing is the key to the edge computing resource management in mobile blockchain. Therefore, we propose an efficient golden section based Stackelberg game equilibrium searching algorithm SES to decide the resource prices of the CSP and the ESPs, which is shown in Algorithm 1. Two points t_1 and t_2 are inserted in the searching interval of $[t_0, t_3]$ based on the ratio of 0.618, where t_0 and t_3 are initialized as 0 and p_n^c, respectively. For points t_1 and t_2, we calculate the optimal amount of computing resources to purchase via Eq. 13. We can then obtain the ESP's utility function values at the two inserted points, i.e. E_n' and E_n'' and obtain the CSP's utility function values at the two inserted points, i.e. $E_{n,c}'$ and $E_{n,c}''$. The searching interval is divided into three segments by the inserted points. We compare the values of E_n' and E_n'' according to the nature of the single-peak function, and one of the segments, either $[t_0, t_1]$ or $[t_2, t_3]$, is deleted to reduce the original searching interval. The algorithm proceeds iteratively following the process of narrowing the searching interval until the size of searching interval $[t_0, t_3]$ is less than the predefined precision threshold ϵ. The middle value of the final searching interval is returned as the approximate maximum value of the ESP's computing resource price.

Algorithm 1 SES.

Input: M_n, C_n, w, $Q_{n,i}$, δ_n, τ, α, $\beta_{n,i}$, μ_n, ϵ, ρ, R^c, R^e, f^c, f_n^e, $\gamma(A)$.

Output: Optimal computing resource price p_n^e and p_n^c.

Initialize $t_0 = 0$, $t_3 = p_n^c$;
while $t_3 - t_0 \geq \epsilon$ **do**
  $t_1 = t_0 + 0.382(t_3 - t_0)$, $t_2 = t_0 + 0.618(t_3 - t_0)$, $t = t_1$;
  while $p_n^c - t \geq \epsilon$ **do**
    Calculate CSP's utility $E_n^{c\prime}$ at price p_n^c with Eq. 11;
    Calculate CSP's utility $E_n^{c\prime\prime}$ at price t with Eq. 11;
    if $E_n^{c\prime} < E_n^{c\prime\prime}$ **then**
      $p_n^c = t + 0.618(p_n^c - t)$;
    else if then
      $t = t + 0.382(p_n^c - t)$;
    end if
  end while
  $p_n^c = \frac{p_n^c + t}{2}$;
  Calculate ESP's utility E_n' at price t_1 with Eq. 5;
  Calculate ESP's utility E_n'' at price t_2 with Eq.5;
  if $E_n' > E_n''$ **then**
    $t_3 = t_2$;
  else if then
    $t_0 = t_1$;
  end if
end while
return p_n^c, $p_n^e = \frac{t_0 + t_3}{2}$.

5 Performance evaluation

In this section, we evaluate the performance of the proposed algorithm SES through simulations. There are 5 ESPs in the system. The unit time cost of service provisioning is 5, and the maximum amount of services that each miner will purchase is 50. The transfer rates of task uploading to the ESPs and the CSP are 10 and 1, respectively. The computing power of the ESPs and the CSP is 1 and 10, respectively. The simulation parameters are listed in Table 2.

Table 2 Simulation parameters

Attributes	Value
the number of ESPs K	5
the unit time cost of service provisioning	5
the maximum amount of services that each miner will purchase	50
the transfer rate of task uploading to CSP R^c	1
the transfer rate of task uploading to ESP R^e	10
the CSP computing power f^c	10
the ESP computing power f_n^e	1
the unit workload of excessive task A, $\gamma(A)$	1.5
the precision threshold ε of algorithm SES	0.01

We also investigate the impact of important parameters, i.e. the number of miners, the cloud service price, the unit service provisioning cost of the ESPs, the weight of service response time for the miners, the unit service reward of miners, the ESPs' computing capacity, etc., on the proposed algorithm.

5.1 Unit service utility of ESPs

Figure 2 shows the unit service utility of ESPs versus different numbers of miners under mode 0 (offloading the excessive tasks to the CSP), assuming $w = 10$, $\alpha = 1$, each $C_n = 200$, $\delta_n = 0$, and $\beta_{n,i} = 20$. The unit service utility of ESPs initially increases and then decreases with increasing number of miners. A large number of miners leads to a large number of computing resource requests, and hence the ESP increases the computing resource price p_n^e to earn more income and control the total amount of

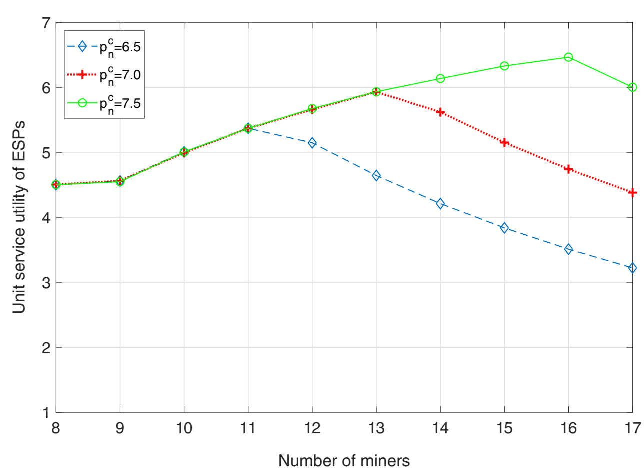


Fig. 2 Unit service utility of ESPs versus the number of miners (mode 0)

services sold. As a result, the unit service utility of ESPs initially shows an increasing trend. When the total number of services purchased by the miners exceeds the ESP's computing capacity, high p_n^e makes the miners unwilling to buy computing resources, resulting in a decreasing trend in the unit service utility of ESPs. A high cloud service price p_n^c makes the ESPs take a large increase of p_n^e to control the total number of services purchased by the miners. Therefore, the unit service utility of ESPs starts to decline with a higher p_n^c later than that with a lower p_n^c.

Figure 3 demonstrates the unit service utility of ESPs with different numbers of miners under mode 1 (offloading the excessive tasks to the other ESPs), assuming $w = 10$, $\alpha = 1$, each $C_n = 200$, $\delta_n = 1$, and $\beta_{n,i} = 20$. The unit service utility of ESPs also initially increases and then decreases as the number of miners increases. The increase of miners potentially leads to more task offloading requests, which increases the ESPs' unit service utility. However, with the increasing amount of service requests, more tasks need to be offloaded to the other ESPs, which incurs more service provisioning cost and hence decreases the ESPs' unit service utility.

Figure 4 shows the unit service utility of ESPs versus different numbers of miners under three different offloading modes of the ESPs, assuming $w = 10$, $\alpha = 1$, each $C_n = 200$, $p_n^c = 8$, and $\beta_{n,i} = 20$. The ESP offloads half of the excessive tasks to the CSP in mode 2, i.e. $\delta_n = 0.5$. The unit service utility of ESPs also initially increases and then decreases as the number of miners increases. However, the unit service utility of ESPs with mode 0 declines faster than that with mode 1, and the unit service utility of ESPs with mode 1 declines faster than that with mode 2. In the following simulations, we assume an overloaded ESP will offload the excessive tasks to the CSP and the other ESP (mode 2). That is, the overloaded ESP can obtain more utilities by

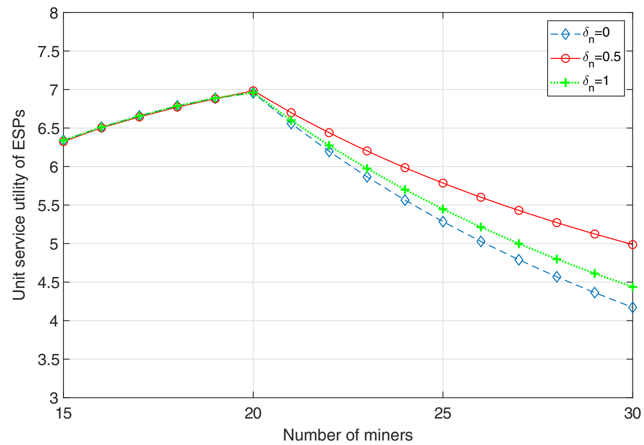


Fig. 4 Unit service utility of ESPs versus the number of miners under three different offloading modes of the ESPs

offloading some of the excessive tasks to the other ESPs than to the CSP. However, it is most beneficial for the overloaded ESP to offload some of the excessive tasks to the other ESPs and offload the other tasks to the CSP at the same time.

Figure 5 illustrates the unit service utility of ESPs with different w, the unit rewards of miners, assuming each $M_n = 10$, $C_n = 200$, $\delta_n = 0.5$, $p_n^c = 8$, and $\beta_{n,i} = 25$. The unit service utility of ESPs increases as w increases. With the increase in the unit service reward w, the number of services that the miners are willing to purchase will continue to increase. For the ESPs, increasing the unit service price p_n^e can obtain higher service revenue, but the increase in the number of services also leads to more ESPs' service provisioning cost and more services purchased from the cloud. With a specific unit service reward w, the unit service utility of ESPs decreases as the unit service provisioning cost α increases.

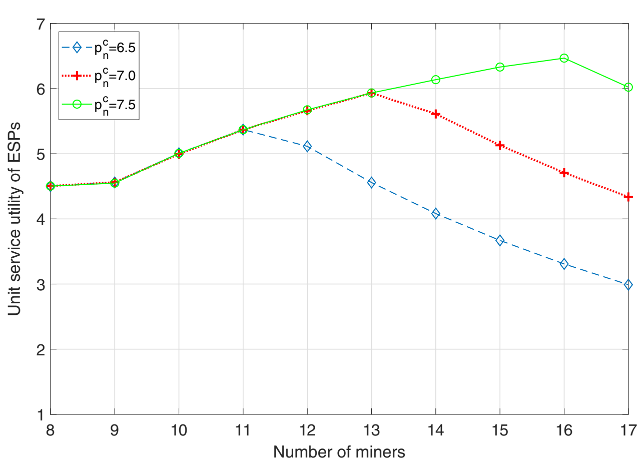


Fig. 3 Unit service utility of ESPs versus the number of miners (mode 1)

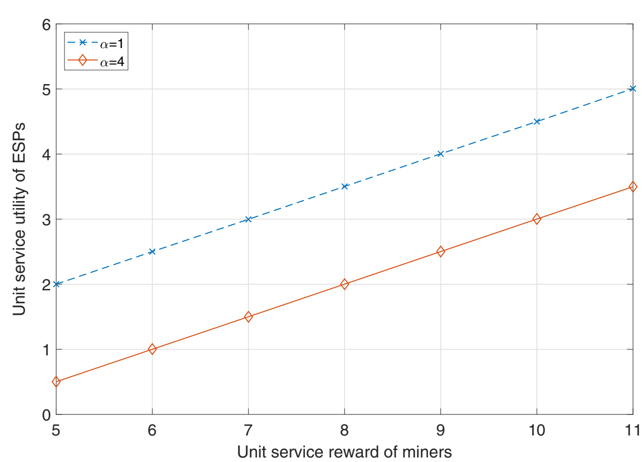


Fig. 5 Unit service utility of ESPs with different unit service rewards of miners

Figures 2–5 demonstrate that the unit service utility of ESPs increases and then decreases as the number of miners increases, the unit service utility of ESPs increases with the increase of the unit service reward of miners and decreases with the increasing unit service provisioning cost, and it is better for the overloaded ESP to offload the mining tasks to both the CSP and the other ESPs.

5.2 Unit service price of ESPs

Figure 6 shows the unit service price of ESPs versus different ESPs' computing capacities, assuming $w = 10$, $\alpha = 1$, each $M_n = 10$, $\delta_n = 0.5$, and $p_n^c = 8$. The unit service price of ESPs increases with the increase of the ESPs' computing capacity. A large ESPs' computing capacity enables a big number of tasks to be executed at the ESPs, such that the ESPs can increase the price to increase the ESPs' utilities. It can be observed that a large β, the weight range of task response time for miners, results in a low unit service utility of ESPs, since the negative impact of executing the tasks at the cloud or at the other ESPs increases with the increase of β.

Figure 7 depicts the unit service price of ESPs by varying the weight of service response time for miners, assuming each $w = 10$, $\alpha = 1$, each $M_n = 10$, $\delta_n = 0.5$, and $p_n^c = 8$. We can observe that as the weight of service response time for miners continues to increase, the unit service price of ESPs first decreases and then keeps stable at a low point. For the ESPs, in order to increase the utility, it is necessary to stimulate the miners to purchase more services through price reduction. With a specific weight of service response time for miners, more ESPs' computing capacities lead to more tasks to be executed at the ESPs, such that the ESPs can increase the unit service price.

Figures 6 and 7 demonstrate that the unit service price of ESPs increases with the increase of ESPs' computing capacity and decreases with the increasing weight of service response time.

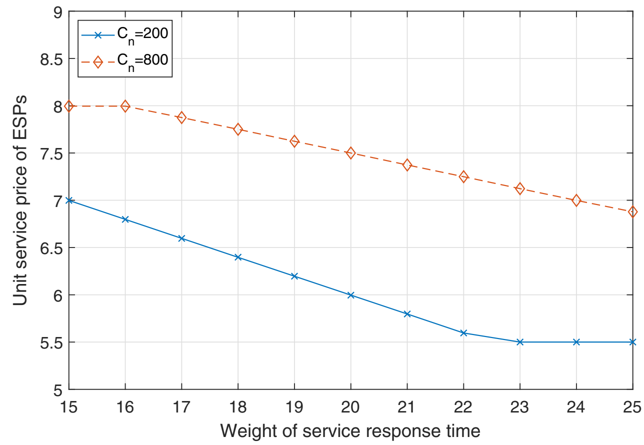


Fig. 7 Unit service price of ESPs with different weights of service response time for miners

5.3 Unit service utility of miners

Figure 8 shows the unit service utility of miners versus different α, the unit service provisioning cost of ESPs, assuming each $M_n = 10$, $\beta_{n,i} = 20$, $\delta_n = 0.5$, $C_n = 200$, and $p_n^c = 8$. The unit service utility of miners decreases as α increases. The increase of α indicates that the ESPs take an increasing cost to provide the services to the miners. Therefore, the miners do not want to buy more services, which leads to a decrease in the unit service utility of miners. With the same α, the increase of unit service reward of miners leads to more rewards of miners through mining, which increases the unit service utility of miners.

Figure 9 depicts the unit service utility of miners by varying w, the unit service reward of miners, assuming $\alpha = 1$, each $M_n = 10$, $\beta_{n,i} = 10$, $\delta_n = 0.5$ and $p_n^c = 8$. The unit service

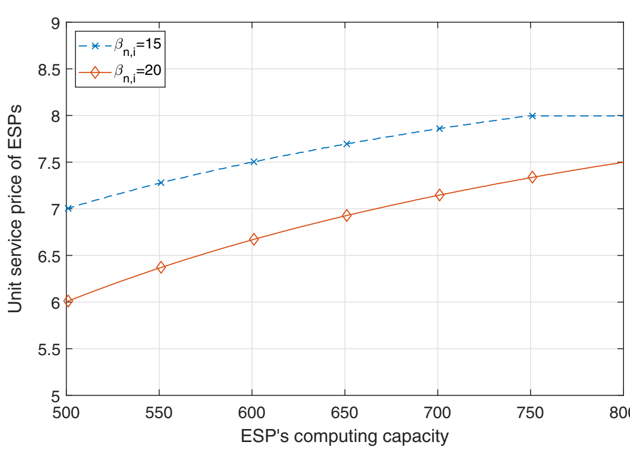


Fig. 6 Unit service price of ESPs with different ESPs' computing capacities

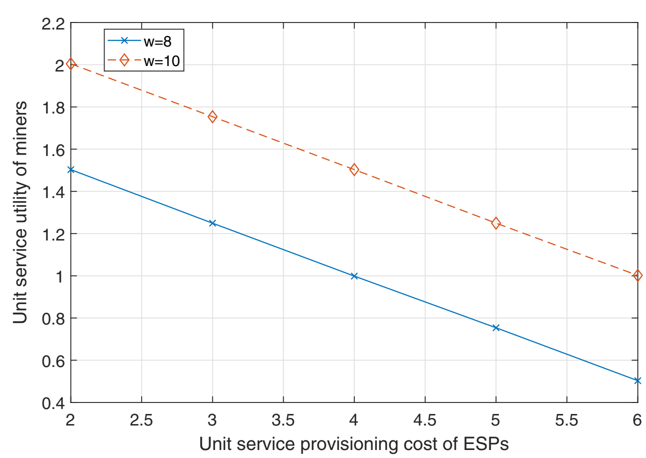


Fig. 8 Unit service utility of miners versus unit service provisioning cost of ESPs

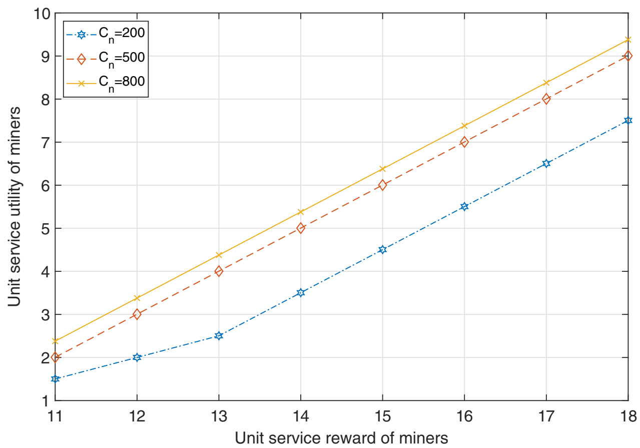


Fig. 9 Unit service utility of miners with different unit service rewards of miners

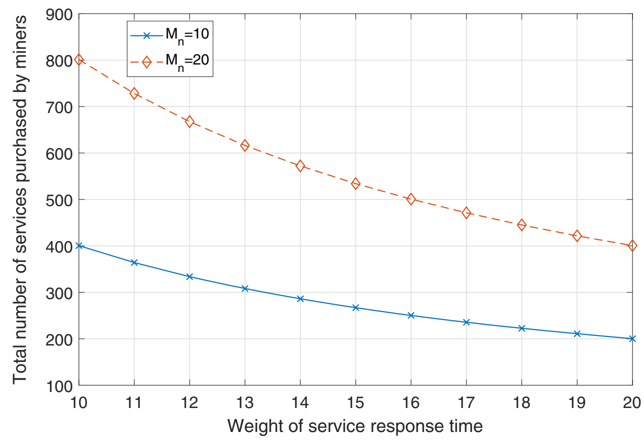


Fig. 11 Total number of services purchased by miners with different weights of service response time for miners

utility of miners increases as w increases. The increase of w leads to more rewards of the miners, such that the miners are willing to purchase more computing resources. With a small ESPs' computing capacity, the miners' competition for edge resources is intense so that a big number of tasks are executed at the CSP, which decreases the miners' utilities.

Figures 8 and 9 demonstrate that the unit service utility of miners increases with the increase of unit service reward of miners and the ESPs' computing capacities, and decreases with the increase of unit service provisioning cost of ESPs.

5.4 Total number of services purchased by miners

Figure 10 shows the total number of services purchased by the miners versus different ESPs' computing capacities, assuming $\alpha = 1$, each $M_n = 10$, $\beta_{n,i} = 15$, $\delta_n = 0.5$, and $p_n^c = 8$. The total number of services purchased by the miners increases as the ESPs' computing capacity increases. More computing resources enable the ESPs to reduce the purchase price to attract more miners to purchase computing resources. Similarly, when the unit service reward is higher, the miners are willing to purchase more services.

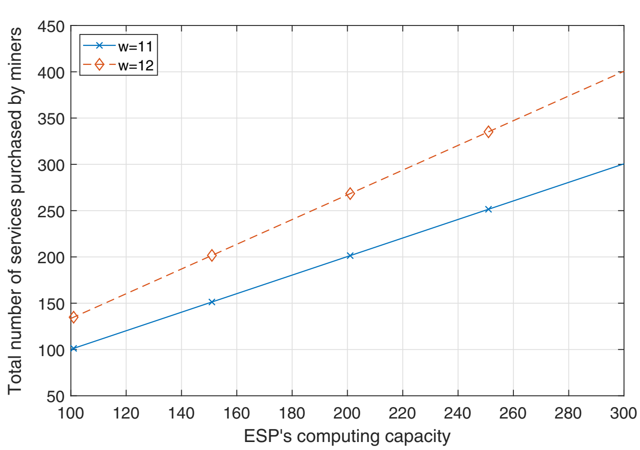


Fig. 10 Total number of services purchased by miners versus ESPs' computing capacities

Figure 11 depicts the total number of services purchased by the miners by varying the weight of service response time for miners, assuming $w = 12$, $\alpha = 1$, each $C_n = 200$, $\delta_n = 0.5$, and $p_n^c = 8$. The total number of services purchased by the miners decreases as the weight of service response time for miners increases. The increase in the weight of service response time indicates that the miners care more about the tasks completion time, such that the miners prefer to make the tasks be executed by the ESPs. Therefore, the total number of resources purchased by the miners will decrease so that the offloaded tasks are within the capacities of the ESPs. With the same weight of service response time, the increase in the number of miners will lead to an increase in the service requests, such that the total number of services purchased by miners also increases.

Figures 10 and 11 demonstrate that the total number of services purchased by the miners increases with the increase of unit service reward of miners and the ESPs' computing capacities, and decreases with the increase of the weight of service response time and the number of miners.

6 Conclusions

Offloading mining computation tasks to an edge computing service provider (ESP) or a cloud computing service provider (CSP) is considered as a feasible solution to blockchain mining in mobile environments. In this paper, we constructed a mining offloading model which includes multiple miners, multiple ESPs, and a CSP. We formulated a three-stage Stackelberg game for optimal pricing-based edge computing resource management. In the first stage, the CSP decides the resource price. In the second stage, each ESP decides the resource price. In the third stage, the miners determine the amount of resources to purchase. We proved the existence and uniqueness of the Stackelberg game equilibrium and derived the miners' optimal amount of computing resources to purchase. We then proposed an efficient golden section based Stackelberg game equilibrium searching algorithm SES for resource pricing. Simulation results showed: (1) the unit service utility of ESPs increases and then decreases as the number of miners increases, and the unit service utility of ESPs increases with the increase of the unit service reward of miners and decreases with the increasing unit service provisioning cost; (2) the unit service price of ESPs increases with the increase of ESPs' computing capacity and decreases with the increasing weight of service response time; (3) the unit service utility of miners increases with the increase of unit service reward of miners and the ESPs' computing capacities, and decreases with the increase of unit service provisioning cost of ESPs; (4) the total number of services purchased by the miners increases with the increase of unit service reward of miners and the ESPs' computing capacities, and decreases with the increase of the weight of service response time and the number of miners; (5) it is better for the overloaded ESP to offload the mining tasks to both the CSP and the other ESPs.

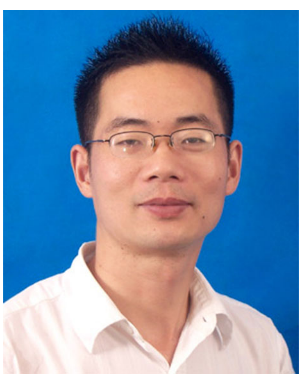
References

1. Blockchain for enterprise applications. https://www.tractica.com/research/blockchain-for-enterprise-applications(2018)
2. Dorri A, Kanhere SS, Jurdak R (2016) Blockchain in Internet of Things: Challenges and solutions. arXiv:1608.05187
3. Rawat DB, Alshaikhi A (2018) Leveraging distributed blockchain-based scheme for wireless network virtualization with security and QoS constraints. In: 2018 International Conference on Computing, Networking and Communications (ICNC). IEEE, pp 332–336
4. Suankaewmanee K, Hoang DT, Niyato D, Sawadsitang S, Wang P, Han Z (2018) Performance analysis and application of mobile blockchain. In: 2018 International Conference on Computing, Networking and Communications (ICNC). IEEE, pp 642–646
5. Jiao Y, Wang P, Niyato D, Xiong Z (2018) Social welfare maximization auction in edge computing resource allocation for mobile blockchain. In: 2018 IEEE International Conference on Communications (ICC), pp 1–6
6. Casado-Vara R, de la Prieta F, Prieto J, Corchado JM (2018) Blockchain framework for IoT data quality via edge computing. In: Proceedings of the 1st Workshop on blockchain-enabled networked sensor systems, pp 19–24
7. Chamarajnagar R, Ashok A (2018) Opportunistic mobile IoT with blockchain based collaboration. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp 1–6
8. Kang J, Yu R, Huang X, Wu M, Maharjan S, Xie S, Zhang Y (2019) Blockchain for secure and efficient data sharing in vehicular edge computing and networks. IEEE Internet of Things Journal 6(3):4660–4670
9. Kim JY, Moon SM (2018) Blockchain-based edge computing for deep neural network applications. In: Proceedings of the workshop on intelligent embedded systems architectures and applications, pp 53–55
10. Kroll JA, Davey IC, Felten EW (2013) The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In: Proceedings of WEIS, p 11
11. Sompolinsky Y, Lewenberg Y, Bachrach Y, Zohar A, Rosenschein JS (2015) Bitcoin mining pools: A cooperative game theoretic analysis. In: Proceedings of the 2015 International conference on autonomous agents and multiagent systems, pp 919–927
12. Houy N (2016) The Bitcoin mining game. Ledger 1:53–68
13. Xiong Z, Feng S, Niyato D, Wang P, Han Z (2018) Optimal pricing-based edge computing resource management in mobile blockchain. In: 2018 IEEE International Conference on Communications (ICC), vol 05, pp 1–6
14. Zhang H, Xiao Y, Bu S, Niyato D, Yu FR, Han Z (2017) Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching. IEEE Internet of Things J 4(5):1204–1215
15. Chiu J, Koeppl T (2019) Incentive compatibility on the blockchain. In: Social design. Springer, pp 323–335
16. Luong NC, Xiong Z, Wang P, Niyato D (2018) Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach. In: 2018 IEEE International Conference on Communications (ICC). IEEE, pp 1–6
17. Jiao Y, Wang P, Niyato D, Suankaewmanee K (2019) Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks. In: IEEE Transactions on parallel and distributed systems
18. Liu M, Yu FR, Teng Y (2018) Computation offloading and content caching in wireless blockchain networks with mobile edge computing. IEEE Trans Veh Technol 67(11):11008–11021
19. Wu Y, Chen X, Shi J, Ni K, Qian L, Huang L, Zhang K (2018) Optimal computational power allocation in multi-access mobile edge computing for blockchain. Sensors 18(10):3472
20. Fan Y, Shen G, Jin Z, Hu D, Shi L, Yuan X (2020) Stackelberg game based edge computing resource management for mobile blockchain. ACM Turing Celebration Conference - China (ACM TURC'20)

Yuqi Fan is an associate professor at School of Computer Science and Information Engineering at Hefei University of Technology, China. He received his Ph.D. in Computer Science and Engineering from Wright State University in 2009. He received both B.S. and M.S. degrees in computer science and engineering from Hefei University of Technology in 1999 and 2003, respectively. His research interests include blockchain, computer networks, cloud computing, cyber-physical systems, etc.

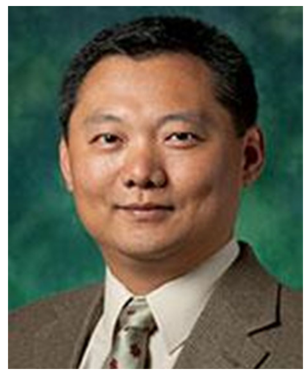
Zhifeng Jin received the B.S. degree in Computer Science and Technology from Hefei University of Technology, China, in 2019. He is currently a graduate student in Computer Science at School of Computer Science and Information Engineering at Hefei University of Technology, Hefei, China. His research interests include blockchain, computer networks, cloud computing, etc.

Guangming Shen received the B.S. degree in Computer Science from Zhejiang Normal University, China, in 2018. He is currently a graduate student in Computer Science at School of Computer and Information Engineering at Hefei University of Technology, Hefei, China. His research interests include blockchain, deep learning, etc.



Donghui Hu received the B.S. degree from Anhui Normal University, Wuhu, China, in 1995, the M.S. degree in computer science and technology from University of Science and Technology of China, Hefei, China, in 2004, and Ph.D. degree in information security from Wuhan University, Wuhan, China, 2010. He was a Visiting Researcher with UNCC IN 2013-2014. Currently, he is a Professor in the College of Computer Science and Information Engineering at the Hefei University of Technology. His research interests include network and information security.

Lei Shi received his B.S. degree in 2002, M.S. degree in 2005, and Ph.D. degree in 2012, all from Hefei University of Technology, Hefei, Anhui, China. He is currently an associate professor in School of Computer Science and Information Engineering, Hefei University of Technology. His main research area lies in edge computing and wireless network optimization.



Xiaohui Yuan received the B.S. degree in electrical engineering from the Hefei University of Technology, China, in 1996, and the Ph.D. degree in computer science from Tulane University in 2004. After his graduation, he was with the National Institutes of Health until 2006. He is currently an Associate Professor with the University of North Texas. His research interests include computer vision, data mining, machine learning, and arti cial intelligence. He was a recipient of the Ralph E. Powe Junior Faculty Enhancement Award in 2008 and the Air Force Summer Faculty Fellowship in 2011, 2012, and 2013. University of North Texas, 3940 N. Elm, Denton, TX 76207, xiaohui.yuan@unt.edu.

Affiliations

Yuqi Fan[1] · **Zhifeng Jin**[1] · **Guangming Shen**[1] · **Donghui Hu**[1] · **Lei Shi**[1] · **Xiaohui Yuan**[2]

Zhifeng Jin
2019170960@hfut.edu.cn

Guangming Shen
2018170910@hfut.edu.cn

Donghui Hu
hudh@hfut.edu.cn

Lei Shi
shilei@hfut.edu.cn

Xiaohui Yuan
xiaohui.yuan@unt.edu

[1] Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui 230601, China

[2] Department of Computer Science and Engineering, University of North Texas, Denton, TX 76203, USA