Delay-limited Computation Offloading for Mobile Blockchain Networks in Edge Computing

Yiping Zuo, Shi Jin, *Senior Member, IEEE,* Shengli Zhang, *Senior Member, IEEE,* and Kai-Kit Wong, *Fellow, IEEE*

Abstract

In this paper, we present new mobile blockchain networks with the help of a mobile edge computing (MEC) server, where all mobile users participate in the proof-of-work (PoW) mining process. To maintain a stable block time of mobile blockchain networks, we formulate delay-limited computation offloading strategies of PoW-based mining tasks as a non-cooperative game with maximizing the individual revenue in the MEC-assisted mobile blockchain networks. Then we analyze specifically the sub-game optimization problem and prove the existence of Nash equilibrium (NE) of this non-cooperative game. Moreover, we design an alternating iterative algorithm based on continuous relaxation and greedy rounding to acquire NE of this game. Given optimal computation offloading strategies, we also derive the optimal transmission power for the individual user within the maximum mining delay range. Our proposed algorithm can efficiently attain optimal delay-limited computation offloading and transmit power strategies for all users. The individual transmission power extends accordingly with the optimal computation resource allocation strategies for all users. Different parameters, such as the number of users, block size, and CPU frequency of the MEC server, have great impacts on the system performance of our proposed delay-limited mobile blockchain networks.

Index Terms

Blockchain, mobile edge computing, computation offloading, mining delay, non-cooperative game

Y. Zuo and S. Jin are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: zuoyiping@seu.edu.cn; jinshi@seu.edu.cn).

S. Zhang is with the College of Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: zsl@szu.edu.cn).

K.-K Wong is with the Department of Electronic and Electrical Engineering, University College London, London, United Kingdom. (e-mail: kaikit.wong@ucl.ac.uk).

I. INTRODUCTION

Essentially, blockchain is a distributed database [1]–[4], where transactions are connected by blocks without the need for a third-party intermediary institution to justify securely legitimacy of transaction information. Blockchain technology naturally has advantages of decentralization, distributed, and tamper-proof. In recent years, scholars and industrial engineers pay great attention to blockchain researches. Significantly, many researchers have done a large number of works on the combination of blockchain and wireless communication technologies, for example, internet of things (IoT) [5]–[7], spectrum allocation [8]–[10], and interference management [11].

In particular, the mining process of verifying transactional legitimacy requires a large amount of intensive computing, which leads to some plights such as heavy equipment and fixed access nodes in traditional blockchain systems. To break these barriers of traditional blockchain systems, mobile blockchain networks deployed many mobile devices have been proposed in [12]–[14], so that various mobile devices, such as mobile phones, iPads, and laptops, can all participate in the blockchain mining process. This proposal of all devices joining mobile blockchain networks in validating transactional legitimacy without any third-party intermediaries can accelerate transaction data processing and ensure data privacy. In the mobile blockchain network, mobile devices have low computation, storage, energy, and communication resources, while the mining task of public blockchains is computation-intensive. Resource-limited mobile device is the bottleneck problem of the development of mobile blockchain networks. Meanwhile, mobile edge computing (MEC) is a computing model that provides computing capabilities for wireless access networks by deploying cloud computing servers at the network edge [15]–[17]. The core idea of MEC is to migrate storage and computation capabilities from cloud to edge of the network, resulting in the consequence that mobile users' computing tasks can be processed directly on their devices or offloaded to nearby service nodes (such as BS and AP) over the wireless channel. As mentioned above, we can see that MEC can solve the bottleneck problem in the mobile blockchain network and provide storage and computation services for mobile devices. Motivated by this, the research of MEC-assisted mobile blockchain networks have been investigated as a promising approach for alleviating the resource limitation of mobile users [12]–[14].

Many existing kinds of research have studied that MEC servers aid mobile users to complete computation-intensive mining tasks of mobile blockchains [12]–[14], [18]–[23]. The authors of

[12] formulated the price-based computation resource allocation for offloading proof-of-work (PoW) mining tasks to cloud/fog providers as a two-stage Stackelberg game model, which jointly maximizes revenues of cloud/fog providers and utilities of each mobile user. In [13], to effectively manage computation resources, the authors proposed an auction-based market model to solve the social welfare optimization problem and design two auction mechanisms in the constant-demand and multi-demand schemes for all mobile users. The work in [14] considered a novel MEC-enabled wireless blockchain framework, where PoW mining tasks are offloaded to nearby APs or a group of D2D devices and transactional content of block can be cached in the MEC server. Resource-constrained mobile users cannot support the computation-intensive PoW mining process, so the reputation-based consensus mechanism was presented to take place of PoW in [18]. The MEC-assisted mobile blockchain was also applied to UAV caching for ultra-reliable communication, and a novel neural-blockchain network was proposed in [19]. Especially, the optimal resource allocation designs in [20] were obtained to minimize the delay of data transmission and computation. Mobile blockchain ensures the authenticity of users' priorities in the MEC networks for healthcare applications. Note that, the simple combination of MEC and mobile blockchain in [12]–[14], [18]–[20] may not yield optimal network performances. As a consequence, some other works have considered deep reinforcement learning [21], [22] and federated learning [23], [24] in the mobile MEC-assisted blockchain networks to obtain optimal system performances.

In these previous works, many pieces of research treated the mining task as a general computing demand, which cannot really reflect the characteristics of the blockchain mining task. Moreover, the traditional blockchain network adjusts the difficulty target value to meet the demand for keeping the block time stable [25]. For mobile blockchain networks, the adjusted difficulty value needs to be widely advertised to every mobile device, which results in too much communication costs. To the best of authors' knowledge, few works focus on the problem mentioned above. To tackle this problem, we consider the mining delay characteristic when designing a new MEC-assisted mobile blockchain system to avoid frequently adjusting the difficulty target value. Different from previous works, this paper jointly optimizes the individual user's nonce length strategy and offloading ratio to the MEC server and analyzes the computation offloading and transmit power allocation in the MEC-aided mobile blockchain networks. Our main contributions of this paper are listed as follows.

- We propose a new MEC-assisted mobile blockchain system and derive the individual delay and revenue of mining a new block for three computation offloading schemes, which are local computation scheme, full computation offloading scheme, and hybrid computation offloading scheme. The analysis of the individual delay and revenue shows that there is a trade-off problem between the individual delay and revenue.
- We formulate computation resource management strategies as a non-cooperative game with maximizing the individual revenue and considering the mining delay limitation. Then we analyze the sub-game optimization problem and prove the existence of Nash equilibrium (NE) of this game. In addition, we propose an alternating iterative algorithm based on continuous relaxation and greedy rounding (CRGR) to obtain optimal computation offloading strategies for all mobile users and derive the optimal individual transmission power within the maximum mining delay range.
- We provide numerical results to validate the convergence performance of our proposed CRGR-based alternating iterative algorithm. Then we demonstrate the effects of different system parameters on the computation resource and transmit power allocation strategies in our proposed MEC-assisted mobile blockchain networks. Numerical results validate the feasibility of our proposed new systems for computational offloading and resource management.

The rest of the paper is organized as follows. Section II represents the MEC-assisted mobile blockchain system model and analyzes the individual delay and revenue of mining a new block in three computation offloading schemes. In Section III, we discuss a trade-off problem between the individual delay and revenue and formulate computation offloading strategies as a non-cooperative game in the MEC-assisted mobile blockchain networks. In Section IV, we prove the existence of NE of the non-cooperative game, propose a CRGR-based alternating iterative algorithm to obtain NE of this game, and derive the optimal individual transmitted power. Simulations are presented in Section V to confirm the analytical results. At last, we conclude the main results of the work in Section VI.

II. System Model

In this section, we first introduce the system model of MEC-assisted mobile blockchain. Then we present the individual delay and revenue of mining a new block in three computation offloading schemes.

A. MEC-assisted Mobile Blockchain Network

In this paper, we consider a MEC-aided mobile blockchain framework, where computation-intensive PoW mining tasks of mobile users are offloaded to the MEC server. As shown in Fig. 1, we assume that there are one MEC server and N mobile users running blockchain application denoted by $\mathcal{N} = \{1, 2, ..., N\}$ in the network. Furthermore, we consider that the MEC server not only has powerful computing capability, but also has storage capacity. Then each mobile user has certain computing capabilities locally. In Fig. 1, we see that the mobile user n offloads own computation-intensive PoW mining tasks of public blockchains and transmits own block header content to the MEC server for $n \in \mathcal{N}$. For simplicity, let $H(\cdot)$denote the hash function, and

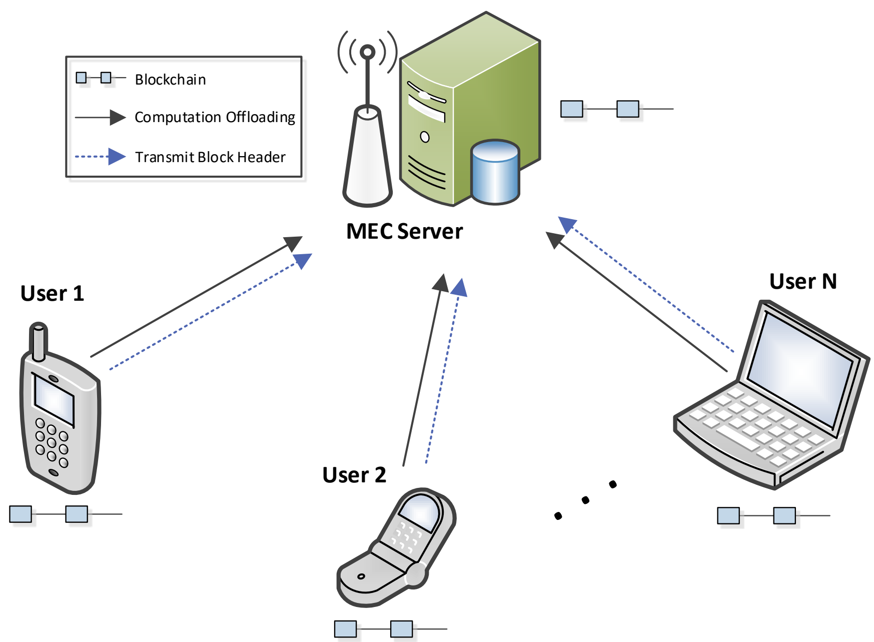


Fig. 1. Computation offloading for MEC-assisted mobile blockchain networks

X denotes the candidate block header information except nonce. The hash value of the block header bh, which concatenates X and nonce, is smaller than the difficulty target value $V(h)$

$$bh = H(X||nonce) \leq V(h), \tag{1}$$

then we have

$$V(h) = 2^{L-h} = \frac{2^L}{D(h)}, \tag{2}$$

where L denotes the fixed length of bits, determining the searching space of the hash function, i.e., all $nonce \in \left[0, 2^L - 1\right]$, and $D(h)$ is the blockchain's difficulty value. If the obtained hash value of the block header is smaller than the difficulty target value, the PoW mining is successful. Otherwise, the nonce is incremented by 1, and the above PoW mining process is repeated until the correct nonce is found.

Due to different transaction contents of the block header for all mobile users, each user independently selects nonces for hash computing. Even though each user chooses the same nonce, the corresponding hash is different. For mobile users, the computation-intensive PoW mining demands of public blockchains are too high (if the SHA-256 hash function is used, then all $nonce \in [0, 2^{32}]$), and given the cost of renting MEC's service and mining delay, the final user's revenue can be reduced or even negative. Thus, we assume that each user only chooses some nonces to do hash computing. We consider a nonce ordering mechanism in the untrusted MEC PoW scheme [25] in this paper to guarantee fairer MEC resource allocation for all mobile users. Each user transmits their own block header content and selected nonce sequence to the MEC server. Then the block header content is stored in the MEC server and N nonce sequences for all mobile users will be mapped into a merged sequence. After preprocessing, the MEC server provides nonce hash computing services for the merged nonce sequence.

The MEC-aided mobile blockchain system discussed in this paper has only one MEC server, where the fork and orphaning probabilities will not be considered. We represent the block size vector by $\mathrm{s} = (s_1, ..., s_N)$, which is the number of transactions packaged into the block. The first user, which successfully mines a block and achieves agreement in the whole network, can get a reward. The reward is composed of a fixed bonus B \$ for mining a new block, and a flexible transaction fee determined by the size of its collected transactions s and the transaction fee rate r \$. Therefore, the n-th mobile user's expected reward regardless of cost can be expressed by

$$F_n = (B + rs_n) \tfrac{M_n}{M_{\mathcal{N}}}, \ \forall n \in \mathcal{N}, \tag{3}$$

then the total number of nonces for all mobile users is given by

$$M_{\mathcal{N}} = \sum_{n \in \mathcal{N}} M_n, \tag{4}$$

where M_n is the selected nonce number/length of user n. However, equation (3) does not reflect the effect of blockchain's target difficulty on the PoW mining process. Even though the MEC server provides all selected nonces with hash computing services, all mobile users may fail to mine a new block. Nonce hash computing is a memoryless searching process, and the searching probability is only related to the target difficulty value $D(h)$, regardless of the size of this searching space. For a given difficulty value $D(h)$, each nonce hash computing is i.i.d. Bernoulli trial with a successful probability as

$$P_D = 2^{-h}. \tag{5}$$

With this effect in mind, the n-th user's expected reward without considering cost gets discounted by P_D, becoming

$$F_n = (B + rs_n)\, 2^{-h} \frac{M_n}{M_{\mathcal{N}}},\ \forall n \in \mathcal{N}. \tag{6}$$

B. Local Computation Scheme

In the following scheme, mobile user n only performs hash computing of PoW mining tasks locally. We assume that the CPU frequency of user n is f_n (cycles/s). The number of nonces selected by user n is M_n and the corresponding CPU cycles required for hash computing are $\alpha_1 M_n$ (cycles), where α_1 is a fixed constant, which is CPU cycles required for once nonce hash computing. Consequently, the total mining delay of performing hash computing locally for user n is

$$T^L_{total,n} = \frac{\alpha_1 M_n}{f_n},\ \forall n \in \mathcal{N}. \tag{7}$$

Then the revenue of only performing hash computing locally for user n is

$$F^L_n = (B + rs_n)\, 2^{-h} \frac{M_n}{M_{\mathcal{N}}},\ \forall n \in \mathcal{N}. \tag{8}$$

C. Full Computation Offloading Scheme

In this full computation offloading scheme, mobile user n offloads the full computation-intensive mining tasks to the MEC server. Firstly, we consider that the time of transmitting the block header content and nonces selected by user n to the MEC server. By Shannon theorem, we will obtain the transmit rate of user n as

$$R_n = W \log_2 \left(1 + \frac{p_n h_n}{N_0}\right),\ \forall n \in \mathcal{N}, \tag{9}$$

where W is the amount of wireless bandwidth, p_n represents the transmission power of user n, h_n denotes the channel gain of user n, and N_0 is the noise power. We assume the data size of the block header except nonce is denoted as $\mathcal{S}$ and the data size of a nonce is denoted as a. As a result, the transmission time required to transfer the block header content and selected nonces to the MEC server is given by

$$T_{t,n}^{E}=\frac{\mathcal{S}+aM_n}{W\log_2\left(1+\frac{p_n h_n}{N_0}\right)},\forall n\in\mathcal{N}. \tag{10}$$

Secondly, we assume that the CPU frequency of the MEC server is f_E (cycles/s) and $f_E \gg f_n$. CPU cycles required for KL divergence between two N-element vectors are $\alpha_2 N$, where α_2 is a fixed constant, and CPU cycles for nonce ordering by the nonce ordering mechanism in our previous work [25] are $\alpha_2 N^2 M_{\mathcal{N}}$. Hence, we consider the ordering time of all nonces on the MEC server as

$$T_{o,n}^{E}=\frac{\alpha_2 N^2 M_{\mathcal{N}}}{f_E},\forall n\in\mathcal{N}. \tag{11}$$

Finally, we consider the execution time required to offload to the MEC server for hash computing is

$$T_{e,n}^{E}=\frac{\alpha_1 M_n}{f_E},\forall n\in\mathcal{N}. \tag{12}$$

The total mining delay of offloading to the MEC server, including the transmission time of block header data, the ordering time of nonces, and the execution time of hash computing, is given by

$$\begin{aligned}&T_{total,n}^{E}\\&=T_{t,n}^{E}+T_{o,n}^{E}+T_{e,n}^{E}\\&=\tfrac{\mathcal{S}+aM_n}{W\log_2\left(1+\frac{p_n h_n}{N_0}\right)}+\tfrac{\alpha_2 N^2 M_{\mathcal{N}}}{f_E}+\tfrac{\alpha_1 M_n}{f_E},\forall n\in\mathcal{N}.\end{aligned} \tag{13}$$

Here, the communication of downlink is just MEC server publishing PoW mining results to all mobile users, where the delay is too small to be ignored. Then the revenue of offloading all PoW mining tasks to the MEC server for user n is

$$F_n^{E}=\left(B+rs_n\right)2^{-h}\frac{M_n}{M_{\mathcal{N}}}-cM_n,\forall n\in\mathcal{N}, \tag{14}$$

where c is the unit price for once hash computing service and cM_n is the computation service cost paid by user n to the MEC server.

D. Hybrid Computation Offloading Scheme

In the following scheme, mobile user n partially has hash computing locally and partially offloads to the MEC server. The mining delay in this scheme is defined as the maximum spending time under two computation offloading schemes of partially local and offloading to the MEC server. Then the total hybrid mining delay is obtained as

$$\begin{aligned} &T_{total,n}^{H} \\ &= \max\left\{T_{total,n}^{H/L}, T_{total,n}^{H/E}\right\} \\ &= \max\left\{\tfrac{\alpha_1(1-\beta_n)M_n}{f_n}, \tfrac{S+a\beta_n M_n}{W\log_2\left(1+\frac{p_n h_n}{N_0}\right)} + \tfrac{\alpha_2 N^2(\beta_n M_n+\sum\beta_{-n}M_{-n})}{f_E} + \tfrac{\alpha_1\beta_n M_n}{f_E}\right\}, \forall n\in\mathcal{N}, \end{aligned} \tag{15}$$

where β_n is the proportion of offloading hash computing tasks to MEC server for user n, β_{-n} and M_{-n} are the other users' offloading ratio and selected nonce length, respectively, and $\sum\beta_{-n}M_{-n}$ is the summation of nonce length, which is offloaded to the MEC server for all users except user n. Then the revenue of partially doing hash computing locally and partially offloading to the MEC server is given by

$$F_n^H = (B + rs_n)\, 2^{-h}\frac{M_n}{M_{\mathcal{N}}} - c\beta_n M_n, \forall n\in\mathcal{N}. \tag{16}$$

III. Non-cooperative Game Formulation

In this section, we first discuss a trade-off problem between the individual delay and revenue in our proposed MEC-assisted mobile blockchain networks. Then we formulate computation resource allocation strategies as a non-cooperative game. Moreover, we represent the sub-game optimization problem to jointly optimize nonce length selection and offloading ratio to the MEC server with the objective of maximizing revenue for each user and considering the limitation of the mining delay. Note that, we only analyze the system performance under the hybrid computation offloading scheme, because the other two computation offloading schemes are special cases of the hybrid scheme.

First of all, we analyze the properties of the delay and revenue of the individual user in three computation offloading schemes, as shown in the following *Remark* 1 and *Remark* 2, respectively.

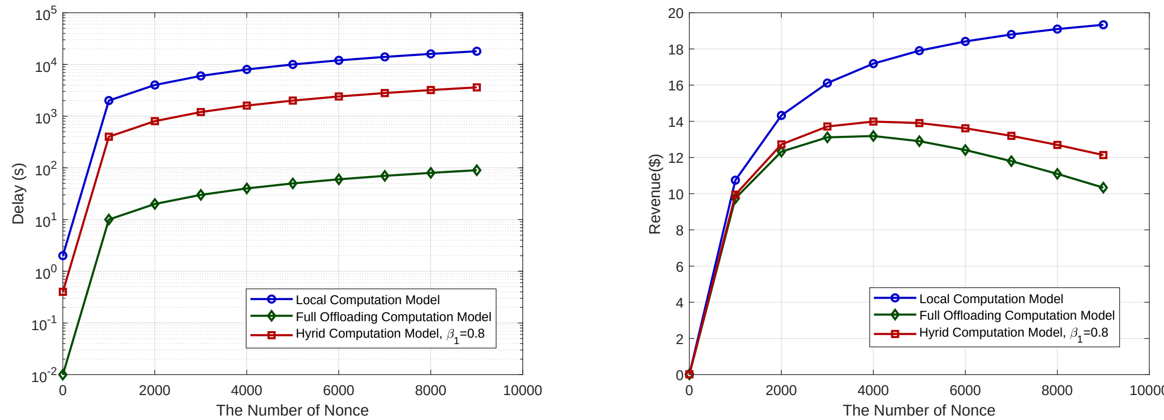
Remark 1 (Properties of delay): From the mining delay in three computation offloading schemes described as (7), (13), and (15), a few observations are in order.

- From (7), we can see that $T^L_{total,n}$ expands with M_n and decreases with f_n. That is, the larger the number of nonces and smaller CPU frequency of user n, the more mining delay of user n can be obtained in the local computation scheme.
- If all mining tasks are offloaded to the MEC server, we easily see that $T^E_{total,n}$ ascends not only with M_n but also with $\sum M_{-n}$, and descends with f_E and p_n. Namely, other users' strategies also effect on the mining delay of user n in the full computation offloading scheme. Note that, different from $T^L_{total,n}$ in (7), $T^E_{total,n}$ in (13) also considers the transmit delay between user n and MEC server, and nonce ordering delay.
- If user n and MEC server complete mining tasks together, from (15), we shall see that $T^{H/L}_{total,n}$ decreases and $T^{H/E}_{total,n}$ increases with the increment of β_n, which means that the mining delay in the hybrid computation offloading scheme does not change linearly with β_n.

Remark 2 (Properties of revenue): From the revenue of the individual user in three computation offloading schemes described as (8), (14), and (16), we easily see that revenues in three schemes, i.e., F^L_n, F^E_n, and F^H_n, are all concave functions of M_n, which implies that we can take optimal nonce selection strategies of user n to achieve maximum individual revenue. In addition, the revenue of user n decreases with β_n.

Fig. 2 plots the individual delay and revenue of user 1 versus the number of nonces. From Fig. 2(a), we can see that the individual delay extends with the number of nonces and the individual delay in the local computation scheme is two to three orders of magnitude greater than that of full computation offloading scheme. The individual delay in the hybrid computation offloading scheme is between the other two schemes. As shown in Fig. 2(b), the individual revenue is a concave function of the number of nonces, and the revenue of the local scheme is higher than that of the other two computation offloading schemes. Combining with Fig. 2(a), we can conclude that more individual revenue needs more delay. Consequently, we need to consider a trade-off problem between the individual delay and revenue. Within the specified mining delay range, how to solve the computation resource allocation problem is the focus of the discussion and analysis below.

The interaction between mobile users can be modeled as a non-cooperative game for computation resource allocation of each mobile user in the MEC-aided mobile blockchain networks. The



(a) Mining Delay of User 1 (b) Mining Revenue of User 1

Fig. 2. Individual delay and revenue versus the number of nonce. $N = 2$, $M_2 = 1000$, $\beta_1 = 0.8$, $\beta_2 = 1$, $s_1 = s_2 = 1000$, $f_1 = f_2 = 0.5GHz$ and $p_1 = p_2 = 0.1W$. The settings of other parameters are consistent with Table I. Here, we only choose two mobile users in the system, and given the computation resource allocation of user 2, then we compare the delay and revenue of user 1 with the number of nonce in three computation offloading schemes.

mobile users compete with each other to maximize their own utility by choosing their individual computation offloading demand A_n, which includes nonce length selection M_n and offloading ratio to the MEC server β_n. The non-cooperative game $\mathcal{G} = \left\{N, (A_n)_{n\in\mathcal{N}}, (u_n)_{n\in\mathcal{N}}\right\}$ is described as follows:

Players: Each user is one player and there are N mobile users selecting the nonce length and offloading ratio;

Strategies: The computation resource allocation $A_n = \{M_n, \beta_n\}$ is the strategy of user n, $\boldsymbol{A} = (A_1, A_2, \cdots, A_N)$ is the computation resource allocation profile for all mobile users;

Utility function: The utility function u_n in (17) is denoted as the revenue of user n.

Given the fee charged by the MEC server for once nonce hash computing c and other users' computation resource allocation strategies, the expected utility u_n is given as

$$\begin{aligned} & u_n\left(A_n, \boldsymbol{A}_{-n}\right) \\ & = u_n\left(M_n, \boldsymbol{M}_{-n}, \beta_n, \boldsymbol{\beta}_{-n}\right), \end{aligned} \tag{17}$$

where $\boldsymbol{A}_{-n} = \{A_1, \cdots, A_{n-1}, A_{n+1}, \cdots, A_N\}$, $\boldsymbol{M}_{-n} = \{M_1, \cdots, M_{n-1}, M_{n+1}, \cdots, M_N\}$ and

$\boldsymbol{\beta}_{-n} = \{\beta_1, \cdots, \beta_{n-1}, \beta_{n+1}, \cdots, \beta_N\}$ are denoted as the computation resource allocation strategy profile, nonce length selection strategy profile, and offloading ratio strategy profile for all other mobile users except user n, respectively. By substituting (16) into (17), the expected utility u_n is rewritten as follows

$$\begin{aligned} & u_n\left(A_n, \boldsymbol{A}_{-n}\right) \\ & = \left(B + r s_n\right) 2^{-h} \frac{M_n}{M_n + \sum M_{-n}} - c\beta_n M_n, \forall n \in \mathcal{N}. \end{aligned} \tag{18}$$

Given a maximum mining delay $\bar{T}$, we formulate the sub-game optimization problem to jointly optimize nonce length selection and offloading ratio to the MEC server with the objective of maximizing revenue of user n for $n \in \mathcal{N}$. This sub-game optimization problem of user n can be expressed as follows

$$\begin{aligned} \mathcal{P}1: \quad & \max_{M_n, \beta_n} \; u_n = \left(B + r s_n\right) 2^{-h} \frac{M_n}{M_n + \sum M_{-n}} - c\beta_n M_n \\ & s.t. \quad C1: \; M_n \geq 0 \\ & \qquad\;\; C2: \; 0 \leq \beta_n \leq 1 \\ & \qquad\;\; C3: \; \left(B + r s_n\right) 2^{-h} \frac{M_n}{\sum\limits_{n=1}^{N} M_n} - c\beta_n M_n \geq 0 \\ & \qquad\;\; C4: \; \max\left\{ \frac{\alpha_1 (1-\beta_n) M_n}{f_n}, \frac{\mathcal{S} + a\beta_n M_n}{W \log_2\left(1 + \frac{p_n h_n}{N_0}\right)} + \frac{\alpha_2 N^2 (\beta_n M_n + \sum \beta_{-n} M_{-n})}{f_E} + \frac{\alpha_1 \beta_n M_n}{f_E} \right\} \leq \bar{T}, \end{aligned} \tag{19}$$

where constraint $C1$ guarantees the non-negativity of nonce length for user n. Constraint $C2$ means the change range of the offloading ratio to the MEC server. Constraint $C3$ ensures the non-negativity of the individual revenue for user n. In order to meet the delay requirement, constraint $C4$ makes sure that the mining delay for hash computing cannot exceed the maximum mining delay limit $\bar{T}$. Our proposed non-cooperative game problem will be divided into N sub-game optimization problems to obtain final solutions for all users. Here, the sub-game optimization problem can be formulated as $\mathcal{P}1$ under certain mining delay constraints, which maximizes the revenue function of the individual revenue and wants to find the optimal computing resource allocation strategy of user n for $n \in \mathcal{N}$.

IV. Delay-limited Computation Offloading and Transmit Power Analysis

In this section, we first simplify the sub-game optimization problem $\mathcal{P}1$ by using continuous relaxation and variable substitution. Then we prove the existence of NE of the non-cooperative game. Furthermore, we propose a CRGR-based alternating iterative algorithm to achieve the optimal nonce length selection and offloading ratio for all users. In the end, we derive the optimal transmission power allocation for each mobile user, which is based on the above optimal computation resource allocation.

A. NE Analysis for Computation Resource Allocation

For simplicity, we define the feasible solution set of $\mathcal{P}1$ as $\mathcal{J}$ ($\mathcal{J} \neq 0$). We consider NE as the solution for this non-cooperative game model $\mathcal{G} = \left\{N, (A_n)_{n\in\mathcal{N}}, (u_n)_{n\in\mathcal{N}}\right\}$, the definition of NE is denoted as:

Definition 1: Let $\boldsymbol{M}^*$ and $\boldsymbol{\beta}^*$ denote the optimal nonce length selection and offloading ratio vectors to the MEC server of all mobile users, respectively. Then a strategy profile $\boldsymbol{A}^* = (\boldsymbol{M}^*, \boldsymbol{\beta}^*)$ is a NE if the following conditions

$$u_n\left(M_n^*, \boldsymbol{M}_{-n}^*, \beta_n^*, \boldsymbol{\beta}_{-n}^*\right) \geq u_n\left(M_n, \boldsymbol{M}_{-n}^*, \beta_n, \boldsymbol{\beta}_{-n}^*\right), \forall \left(M_n, \beta_n\right) \in \mathcal{J}, \forall n \in \mathcal{N}, \qquad (20)$$

is satisfied, where $\boldsymbol{M}_{-n}^*$ and $\boldsymbol{\beta}_{-n}^*$ are the best response nonce length selection and offloading ratio vectors to the MEC server for all mobile users except user n.

From the above NE definition, we can see that NE has self-stability property. When other players' strategies are given, no player has the incentive to deviate to another strategy. This property is critical to the non-cooperative computation resource allocation problem, since each mobile user is selfish to mine new blocks in their own interests. Observing $\mathcal{P}1$, we can find that the n-th user's computation resource allocation strategy $\{M_n, \beta_n\}$ is associated with other users' strategies $\{\boldsymbol{M}_{-n}, \boldsymbol{\beta}_{-n}\}$. As such, we first investigate the sub-game optimization problem $\mathcal{P}1$, and then iteratively solve the sub-game problem to obtain the NE of the entire game $\mathcal{G}$. Next, we will analyze the sub-game problem first. Sub-game optimization problem $\mathcal{P}1$ is difficulty to tackle due to the following two aspects:

- Since M_n has the non-negative integer constraint and β_n is a continuous variable, making the problem $\mathcal{P}1$ become a mixed integer programming problem.
- The inequality function of constraint C4 is a non-smooth function and non-convex function with regards to variables (M_n, β_n), so the feasible set of $\mathcal{P}1$ is not convex.

In brief, $\mathcal{P}1$ is a mixed discrete and non-convex optimization problem, which is also well known as NP-hard problem. As a result, some transformation and simplification of the original sub-game optimization problem are necessary. To tackle the difficulty of the original optimization problem, we can make the following transformation and relaxation:

1) ***Continuous Relaxation of Target Variable***

   We first continuously relax the target variable M_n and find the optimal solution of this continuous variable. Then we can get the optimal nonce length strategies of user n by rounding this continuous solution.

2) ***Constraint Relaxation***

   In order to solve the non-smoothness of the inequality function of constraint $C4$, we relax the constraint $C4$ to make the inequality function smooth. Constraint $C4$ can be relaxed to the following two inequality constraints

$$\begin{cases} C4'\!-\!1: \ \frac{\alpha_1(1-\beta_n)M_n}{f_n} \leq \bar{T}, \forall n \in \mathcal{N}, \\ C4'\!-\!2: \ \frac{\mathcal{S}+a\beta_n M_n}{W\log_2\left(1+\frac{p_n h_n}{N_0}\right)} + \frac{\alpha_2 N^2(\beta_n M_n + \sum \beta_{-n} M_{-n})}{f_E} + \frac{\alpha_1 \beta_n M_n}{f_E} \leq \bar{T}, \forall n \in \mathcal{N}. \end{cases} \tag{21}$$

   The reason is that the maximum delay function is less than or equal to the maximum mining delay threshold $\bar{T}$, which is equivalent to each value in the maximum function (i.e., $T^{H/L}_{total,n}$ and $T^{H/E}_{total,n}$) being less than or equal to this threshold. The original sub-game optimization problem of user n for $n \in \mathcal{N}$ can be transformed into

$$\begin{aligned} \mathcal{P}2: \quad & \max_{M_n,\beta_n} u_n = (B + rs_n)\, 2^{-h} \frac{M_n}{M_n + \sum M_{-n}} - c\beta_n M_n \\ s.t. \quad & C1: \ M_n \geq 0 \\ & C2: \ 0 \leq \beta_n \leq 1 \\ & C3: \ (B + rs_n)\, 2^{-h} \frac{M_n}{\sum\limits_{n=1}^{N} M_n} - c\beta_n M_n \geq 0 \\ & C4'\!-\!1: \ \frac{\alpha_1(1-\beta_n)M_n}{f_n} \leq \bar{T} \\ & C4'\!-\!2: \ \frac{\mathcal{S}+a\beta_n M_n}{W\log_2\left(1+\frac{p_n h_n}{N_0}\right)} + \frac{\alpha_2 N^2(\beta_n M_n + \sum \beta_{-n} M_{-n})}{f_E} + \frac{\alpha_1 \beta_n M_n}{f_E} \leq \bar{T} \end{aligned} \tag{22}$$

3) ***Variable Substitution***

   After the relaxation of target variable M_n and constraint $C4$, the objective function of the optimization problem $\mathcal{P}2$ is a concave function. Then constraints of $C1-C3$ and $C4'-2$ are convex sets with respect to all the target variables M_n and β_n. However, constraint $C4'-1$ is concave set to the optimization variables M_n and β_n, so the transformed $\mathcal{P}2$ problem is still a non-convex problem, which is a difference of convex programming problem. Here, we introduce an auxiliary variable $Z_n = 1 - \beta_n$, which is the proportion of computation-intensive PoW mining tasks on the mobile user n locally. Then we reformulate constraint $C4'{-}1$ as follows

$$C4''{-}1: \ \frac{\alpha_1 Z_n M_n}{f_n} \leq \bar{T}, \forall n \in \mathcal{N}. \tag{23}$$

Combining the above target variable relaxation, constraint relaxation, and the auxiliary variable Z_n, the original sub-game optimization problem of user n for $n \in \mathcal{N}$ can be equivalent to

$$\begin{aligned} \mathcal{P}3: \quad & \max_{M_n,\beta_n,Z_n} u_n = \left(B + rs_n\right) 2^{-h\frac{M_n}{M_n + \sum M_{-n}}} - c\beta_n M_n \\ s.t. \quad & C1: \ M_n \geq 0 \\ & C2: \ 0 \leq \beta_n \leq 1 \\ & C3: \ \left(B + rs_n\right) 2^{-h\frac{M_n}{M_n + \sum M_{-n}}} - c\beta_n M_n \geq 0 \\ & C4''{-}1: \ \tfrac{\alpha_1 Z_n M_n}{f_n} \leq \bar{T} \\ & C4'{-}2: \ \tfrac{\mathcal{S} + a\beta_n M_n}{W\log_2\left(1 + \frac{p_n h_n}{N_0}\right)} + \tfrac{\alpha_2 N^2\left(\beta_n M_n + \sum \beta_{-n} M_{-n}\right)}{f_E} + \tfrac{\alpha_1 \beta_n M_n}{f_E} \leq \bar{T} \\ & C5: Z_n + \beta_n = 1, \end{aligned} \tag{24}$$

where constraint $C5$ ensures that the sum of the proportion of the n-th user's mining tasks offloaded to the MEC server and user n is 1. The original sub-game optimization problem $\mathcal{P}1$ is ultimately equivalent to the optimization problem $\mathcal{P}3$.

Next, we will analyze the existence of NE in the non-cooperative game $\mathcal{G} = \left\{N, \left(A_n\right)_{n\in\mathcal{N}}, \left(u_n\right)_{n\in\mathcal{N}}\right\}$, which is described by the following theorem.

Theorem 1: The non-cooperative game $\mathcal{G} = \left\{N, \left(A_n\right)_{n\in\mathcal{N}}, \left(u_n\right)_{n\in\mathcal{N}}\right\}$ has at least one pure-strategy NE, i.e., the existence of NE.

Proof: After the above target variable relaxation, constraint relaxation, and variable substitution, constraint $C5$ is a closed set, constraint $C3$ is a convex set, and all other constraints are linear with regards to variables M_n, β_n, and Z_n. Obviously, the utility function $u_n\left(M_n, \beta_n\right)$ is a continuous function in convex sets of constraints $C1 - C5$. Then we take the second order derivatives of (24) with respect to M_n and β_n to obtain the Hessian matrix, which can be written as follows

$$\mathcal{H}_n = \begin{bmatrix} \frac{\partial u_n^2}{\partial M_n^2} & \frac{\partial u_n^2}{\partial M_n \beta_n} \\ \frac{\partial u_n^2}{\partial \beta_n M_n} & \frac{\partial u_n^2}{\partial \beta_n^2} \end{bmatrix} = \begin{bmatrix} \left(B + rs_n\right) 2^{-h} \frac{-2\sum M_{-n}}{\left(M_n + \sum M_{-n}\right)^3} & -c \\ -c & 0 \end{bmatrix} \preceq 0, \forall n \in \mathcal{N}. \tag{25}$$

Therefore, we proved the objective function u_n is a concave function with respect to variables M_n and β_n. Thus, problem $\mathcal{P}3$ is a convex optimization problem. Accordingly, the NE exists (see [26]-Theorem 3.2) in the non-cooperative game $\mathcal{G} = \left\{N, (A_n)_{n\in\mathcal{N}}, (u_n)_{n\in\mathcal{N}}\right\}$. The proof is now completed. ■

At this time, the sub-game optimization problem $\mathcal{P}3$ is a convex optimization problem of all optimization variables M_n, β_n, and Z_n. The traditional optimization methods, such as the interior point method and standard gradient projection method, can be used to find the n-th user's optimal solution of $\mathcal{P}3$. Based on Theorem 1, we can propose an effective algorithm, which achieves optimal computing resource allocation strategies for all mobile users as shown in the following subsection.

B. Algorithm Proposal for Delay-limited Computation Resource Allocation

In this subsection, we design an alternating iterative algorithm to achieve optimal numerical solutions of computation resource allocation strategies for all mobile users in the MEC-assisted mobile blockchain networks. In our proposed alternating iterative algorithm, a greedy rounding method is applied. Specifically, for each sub-game optimization problem $\mathcal{P}3$, the optimal solution $\{M_n^+, \beta_n^+, Z_n^+\}$ is rounded down greedily to $\left\{M_{n,d}^*, \beta_{n,d}^*, Z_{n,d}^*\right\}$ and rounded up to $\left\{M_{n,u}^*, \beta_{n,u}^*, Z_{n,u}^*\right\}$. Then we compare revenues of these two rounded solutions and take the one with the larger revenue as the optimal solution with the integer constraint. For the detailed process, our proposed CRGR-based alternating iterative algorithm to obtain NE of the non-cooperative game $\mathcal{G}$ is summarized in Algorithm 1.

Our proposed algorithm is based on alternating iterations, and its core idea is to achieve optimal delay-limited computation offloading strategy of the individual user in a distributed method when other users' computation resource allocation strategies remain fixed. Since the NE of the non-cooperative game $\mathcal{G} = \left\{N, (A_n)_{n\in\mathcal{N}}, (u_n)_{n\in\mathcal{N}}\right\}$ exists, which enables each user can reach a relatively stable state. As such, our proposed CRGR-based alternating iterative algorithm is convergent. The computing pressure of Algorithm 1 is primarily engrossed in line 3 and line 12, and the other parts are only some basic algorithm operations. At line 3, we require to calculate the optimal solution $\{M_n^+, \beta_n^+, Z_n^+\}$ of user n for $n \in \mathcal{N}$. Specifically, the computational complexity of one iteration using the interior point method is $\mathcal{O}\left(\max\left\{K_1^3, K_1^2 K_2\right\}\right)$ [27], where $K_1 = 3$ and $K_2 = 7$ are numbers of optimization variables and constraints of the sub-game optimization

Algorithm 1 Alternating Iterative Algorithm Based on Continuous Relaxation and Greedy Rounding (CRGR) of the non-cooperative game $\mathcal{G}$

1: **Initialization:** input data $(N, \alpha_1, \alpha_2, f_E, \mathcal{S}, a, W, N_0)$, set (f_n, h_n) for all users, $n = 1$ and choose $\mathbf{M}_{-1}= [2000, 2000, \cdots, 2000]_{1\times(N-1)}, \boldsymbol{\beta}_{-1} = [0.8, 0.8 \cdots 0.8]_{1\times(N-1)}$.

2: **repeat**

3: Fix other users' nonce lengths and offloading ratios $\boldsymbol{M}_{-n} = (M_1, M_2, \cdots, M_{n-1}, M_{n+1}, \cdots, M_N)$, $\boldsymbol{\beta}_{-n} = (\beta_1, \beta_2, \cdots, \beta_{n-1}, \beta_{n+1}, \cdots, \beta_N)$, and solve sub-game optimization problem $\mathcal{P}3$ using the interior point method to obtain the optimal solution $\{M_n^+, \beta_n^+, Z_n^+\}$.

4: Let $M_{n,d}^* = \lfloor M_n^+ \rfloor$, $\beta_{n,d}^* = \frac{\lfloor \beta_n^+ M_n^+ \rfloor}{\lfloor M_n^+ \rfloor}$, and $Z_{n,d}^* = 1 - \beta_{n,d}^*$.

5: Let $M_{n,u}^* = \lceil M_n^+ \rceil$, $\beta_{n,u}^* = \frac{\lceil \beta_n^+ M_n^+ \rceil}{\lceil M_n^+ \rceil}$, and $Z_{n,u}^* = 1 - \beta_{n,u}^*$.

6: **if** $u_n\left(M_{n,d}^*,\ \beta_{n,d}^*,\ Z_{n,d}^*\right) \geq u_n\left(M_{n,u}^*,\ \beta_{n,u}^*,\ Z_{n,u}^*\right)$ **then**

7: The optimal solution is $\{M_n^*,\ \beta_n^*,\ Z_n^*\} = \left\{M_{n,d}^*,\ \beta_{n,d}^*,\ Z_{n,d}^*\right\}$

8: **else**

9: The optimal solution is $\{M_n^*,\ \beta_n^*,\ Z_n^*\} = \left\{M_{n,u}^*,\ \beta_{n,u}^*,\ Z_{n,u}^*\right\}$

10: **end if**

11: $n \leftarrow n + 1$

12: Then update other users' sets $\boldsymbol{M}_{-n} =(M_1, M_2, ..., M_{n-1}^*,\ M_{n+1}, ..., M_N)$ and $\boldsymbol{\beta}_{-n} = \left(\beta_1, \beta_2, \cdots, \beta_{n-1}^*, \beta_{n+1}, \cdots, \beta_N\right)$, then go to line 3.

13: **until** the optimal vectors $\boldsymbol{M}^* = \{M_1^*, M_2^*, ..., M_N^*\}$, $\boldsymbol{\beta}^* = (\beta_1^*, \beta_2^*, \cdots, \beta_N^*)$, and $\boldsymbol{Z}^* = (Z_1^*, Z_2^*, \cdots, Z_N^*)$ are obtained.

problem $\mathcal{P}_3$, respectively. As a result, we can easily attain the computational complexity of the entire Algorithm 1 as $\mathcal{O}\left(I_I I_O \max\left\{K_1^3, K_1^2 K_2\right\}\right)$, where I_I represents the number of iterations required for the convergence of the interior point method in the inner loop, which is usually relatively small, such as $I_I \leq 100$. And $I_O = N + 1$ indicates the number of iterations of the outer loop, later numerical experiments will also prove that Algorithm 1 only needs $N + 1$ iterations to achieve good convergence performance. As a result, our proposed alternating iterative algorithm described in Algorithm 1 can acquire the NE point of the non-cooperative game $\mathcal{G}$ in the polynomial time.

C. Transmission Power Allocation

By using Algorithm 1, the proposed non-cooperative game $\mathcal{G} = \left\{N, (A_n)_{n\in\mathcal{N}}, (u_n)_{n\in\mathcal{N}}\right\}$ can obtain NE, that is, the computation resource allocation of each user can finally reach a stable state. At present, we guarantee the maximum revenue of the individual user within a certain mining delay range, we consider saving the transmit power of each user. In order to obtain the optimal transmission power of the individual user, we give the following theorem.

Theorem 2: Given the optimal computing resource allocation $\{\boldsymbol{M}^*, \boldsymbol{\beta}^*\}$, we can get the optimal transmission power allocation for user n as

$$p_n^* = \frac{N_0}{h_n}\left(2^{\frac{\mathcal{S}+a\beta_n^* M_n^*}{W\left(\bar{T}-\frac{\alpha_2 N^2\left(\beta_n^* M_n^* + \sum \beta_{-n}^* M_{-n}^*\right)}{f_E}-\frac{\alpha_1 \beta_n^* M_n^*}{f_E}\right)}} - 1\right), \forall n \in \mathcal{N}, \tag{26}$$

where β_{-n}^* and M_{-n}^* are the optimal others' offloading ratio and selected nonce length, respectively, and $\sum \beta_{-n}^* M_{-n}^*$ is the summation of optimal nonce length, which is offloaded to the MEC server for all mobile users except user n.

Proof: The NE of the non-cooperative game $\mathcal{G}$ is obtained through our proposed CRGR-based alternating iterative algorithm. Given the optimal computing resource allocation $\{\boldsymbol{M}^*, \boldsymbol{\beta}^*\}$, the mining delay of the MEC server meets the maximum delay requirement, which means the following inequality holds

$$\frac{\mathcal{S}+a\beta_n^* M_n^*}{W\log_2\left(1+\frac{p_n h_n}{N_0}\right)} + \frac{\alpha_2 N^2\left(\beta_n^* M_n^* + \sum \beta_{-n}^* M_{-n}^*\right)}{f_E} + \frac{\alpha_1 \beta_n^* M_n^*}{f_E} \leq \bar{T}, \forall n \in \mathcal{N}. \tag{27}$$

After some simple operations on (27), we can easily achieve the transmitted power of user n as

$$p_n \geq \frac{N_0}{h_n}\left(2^{\frac{\mathcal{S}+a\beta_n^* M_n^*}{W\left(\bar{T}-\frac{\alpha_2 N^2\left(\beta_n^* M_n^* + \sum \beta_{-n}^* M_{-n}^*\right)}{f_E}-\frac{\alpha_1 \beta_n^* M_n^*}{f_E}\right)}} - 1\right), \forall n \in \mathcal{N}. \tag{28}$$

In order to save the transmitted power as much as possible, we can obtain the optimal or minimum individual transmission power as shown in (26). This completes the proof. ■

Note that the result in (26) can achieve a closed-form expression of the optimal transmission power of the individual mobile user, given the delay-limited computation offloading strategies for all mobile users. Based on Theorem 2, we investigate the relationship between the optimal individual transmission power and its own optimal computation resource allocation strategy as the following proposition.

Proposition 1: The optimal individual transmission power p_n^* increases monotonically with M_n^* and β_n^*.

Proof: Firstly, we analyze the relationship between p_n^* and M_n^* and we take the first order derivative of (26) with respect to M_n as follows

$$\begin{aligned} &\frac{\partial p_n^*}{\partial M_n^*} \\ &= \frac{N_0 \ln 2\left(a\beta_n^*\phi + (S + a\beta_n^* M_n^*)\left(\frac{\alpha_2 N^2 \beta_n^*}{f_E} + \frac{\alpha_1 \beta_n^*}{f_E}\right)\right)}{h_n W \phi^2} \cdot 2^{\frac{S + a\beta_n^* M_n^*}{W\phi}}, \forall n \in \mathcal{N}, \end{aligned} \tag{29}$$

where

$$\phi = \bar{T} - \frac{\alpha_2 N^2 \left(\beta_n^* M_n^* + \sum \beta_{-n}^* M_{-n}^*\right)}{f_E} - \frac{\alpha_1 \beta_n^* M_n^*}{f_E}. \tag{30}$$

Meanwhile, the optimal computation resource allocation for user n satisfies the constraint condition $C4' - 2$ in the problem $\mathcal{P}3$, that is, the following inequality holds

$$\bar{T} - \frac{\alpha_2 N^2 \left(\beta_n^* M_n^* + \sum \beta_{-n}^* M_{-n}^*\right)}{f_E} - \frac{\alpha_1 \beta_n^* M_n^*}{f_E} \geq \frac{\mathcal{S} + a\beta_n^* M_n^*}{W \log_2\left(1 + \frac{p_n h_n}{N_0}\right)} \geq 0, \forall n \in \mathcal{N}. \tag{31}$$

As such, we proved $\frac{\partial p_n^*}{\partial M_n^*} > 0$, which means that p_n^* increases monotonically with M_n^*. Similarly, we can get $\frac{\partial p_n^*}{\partial \beta_n^*} > 0$, which also proves p_n^* increases monotonically with β_n^*. The proof of Proposition 1 is now completed. ■

According to Proposition 1, we see that the optimal individual transmission power of user n needs to strengthen when its own optimal computation resource allocation $A_n^* = \{M_n^*, \beta_n^*\}$ raises for $n \in \mathcal{N}$. That is, the larger number of selected nonces and offloading ratio of user n to the MEC server, the more individual transmission power can be consumed for $n \in \mathcal{N}$. Next, we examine the relationship between the optimal individual transmitted power and other users' computation resource allocation strategies as the following proposition.

Proposition 2: The optimal individual transmission power p_n^* also increases monotonically with $\sum \beta_{-n}^* M_{-n}^*$.

Proof: From the closed-form expression (26), we can certainly work on the relationship between p_n^* and $\sum \beta_{-n}^* M_{-n}^*$ and take the first order derivative of (26) with respect to $\sum \beta_{-n}^* M_{-n}^*$ as follows

$$\begin{aligned} &\frac{\partial p_n^*}{\partial\left(\sum \beta_{-n}^* M_{-n}^*\right)} \\ &= \frac{N_0 \ln 2 (S + a\beta_n^* M_n^*) \alpha_2 N^2}{h_n f_E W \phi^2} \cdot 2^{\frac{S + a\beta_n^* M_n^*}{W\phi}}, \forall n \in \mathcal{N}. \end{aligned} \tag{32}$$

According to the proof process of Proposition 1, we can easily get $\phi \geq 0$. On that account, we proved $\frac{\partial p_n^*}{\partial \left(\sum \beta_{-n}^* M_{-n}^*\right)} > 0$, which indicates that p_n^* also extends monotonically with $\sum \beta_{-n}^* M_{-n}^*$. The proof of Proposition 2 is finished completely. ∎

Here, $\sum \beta_{-n}^* M_{-n}^*$ is meant to the summation of optimal nonce lengths offloaded to the MEC server for all users except user n. From Proposition 2, we note that the n-th user's optimal transmission power also elevates when other users' total number of nonces offloaded to the MEC server raises. The main reason for this phenomenon is that the increment of other users' offloading tasks to the MEC server may stimulate the more computation offloading of user n to the MEC server for $n \in \mathcal{N}$. Combined with Proposition 1, we conclude that the individual transmission power of user n will extend accordingly with the optimal computation resource allocation strategies for all mobile users.

V. Numerical Results

In our numerical results, we first perform the convergence of the proposed CRGR-based alternating iterative algorithm. Then we present the impact of different parameters on the system performance of our proposed MEC-assisted mobile blockchain networks. Moreover, we use computer simulations to evaluate the delay-limited computation resource management and the optimal individual transmission power in the proposed mobile blockchain networks with the need of the MEC server. The main simulation parameters are listed in Table I.

A. The Convergence of the CRGR-based Alternating Iterative Algorithm

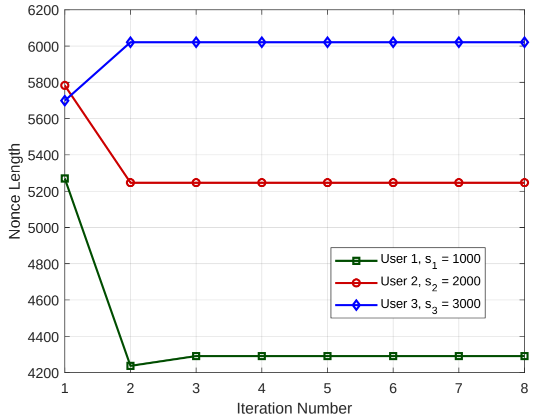
Fist of all, we present the convergence performance of our proposed CRGR-based alternating iterative algorithm described in Algorithm 1. Without loss of generality, we assume only three mobile users in this MEC-assisted mobile blockchain network. CPU frequencies of all users and MEC server are set at $f_1 = f_2 = f_3 = 0.5$ GHz and $f_E = 100$ GHz , respectively. The all users' original transmitted power is $0.1W$ and block sizes of all users are $s_1 = 1000$, $s_2 = 2000$, and $s_3 = 3000$.

In Fig. 3, we plot the individual nonce length, offloading ratio to the MEC server, and revenue versus the number of iteration to depict the convergence of the outer loop of Algorithm 1 in Fig. 3(a), Fig. 3(b) and Fig. 3(c), respectively. There is no simulation about the convergence of the inner loop of Algorithm 1, because the interior-point method used for the inner loop

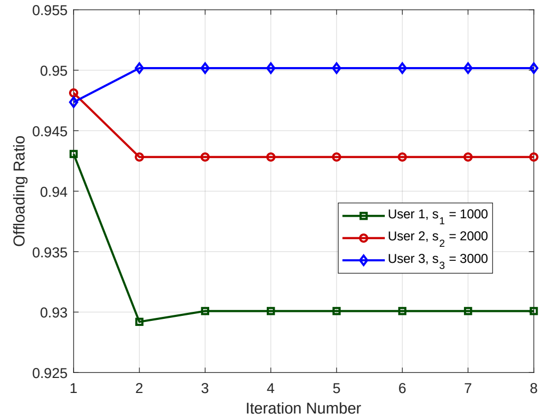
TABLE I. Simulation Parameters

Parameter	**Value**
CPU cycles required for one nonce hash computing α_1	1000 Mega cycles
CPU cycles required for KL divergence between two one-dimensional vectors α_2	10^{-10} cycles
CPU frequency of user n, f_n	$0.5-1$ GHz
CPU frequency of the MEC server f_E	$50-150$ GHz
The transmit bandwidth W	20 MHz
The channel gain of user n, h_n	10^{-8}
The power of noise N_0	-100 dBm
The data size of the block header except nonce $\mathcal{S}$	$76*8=608$ bit
The data size of one nonce a	$4*8=32$ bit
The threshold of the maximum mining delay $\bar{T}$	600 s
The fixed bonus for mining a new block B	$2*10^4$ \$
The transaction fee rate r	2 \$
The transaction size of user n, s_n	$500-3000$
The fee charged by the MEC server for one nonce hash computing c	0.001 \$
The adjustable difficulty parameter h	10

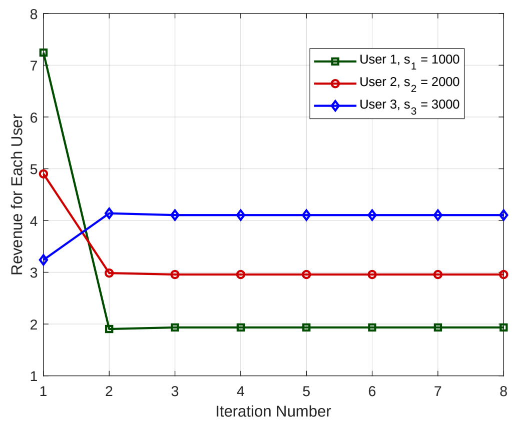
usually requires only a few iterations to converge, such as within 100 iterations [28]. Observing from Fig. 3, we can observe that the individual nonce length, offloading ratio, and revenue of Algorithm 1 are all unstable in the first 4 iterations for three users. As shown in Fig. 3, when the iteration is larger than 4, the proposed CRGR-based alternating iterative algorithm reaches a stable state. We can also observe that when computing resource allocation strategies including nonce lengths and offloading ratios of all three users tend to stabilize in Fig. 3(a) and Fig. 3(b), the individual revenue of all users will converge in Fig. 3(c). This is because the individual revenue is greatly affected by computing resource allocation strategies in our proposed MEC-assisted mobile blockchain networks. Hence, we can see that Algorithm 1 has a significantly fast convergence rate and the number of iteration is always no more than 4. This also shows that our proposed Algorithm 1 can achieve the NE of the non-cooperative game $\mathcal{G}$ in the polynomial time.



(a) Nonce Length



(b) Offloading Ratio

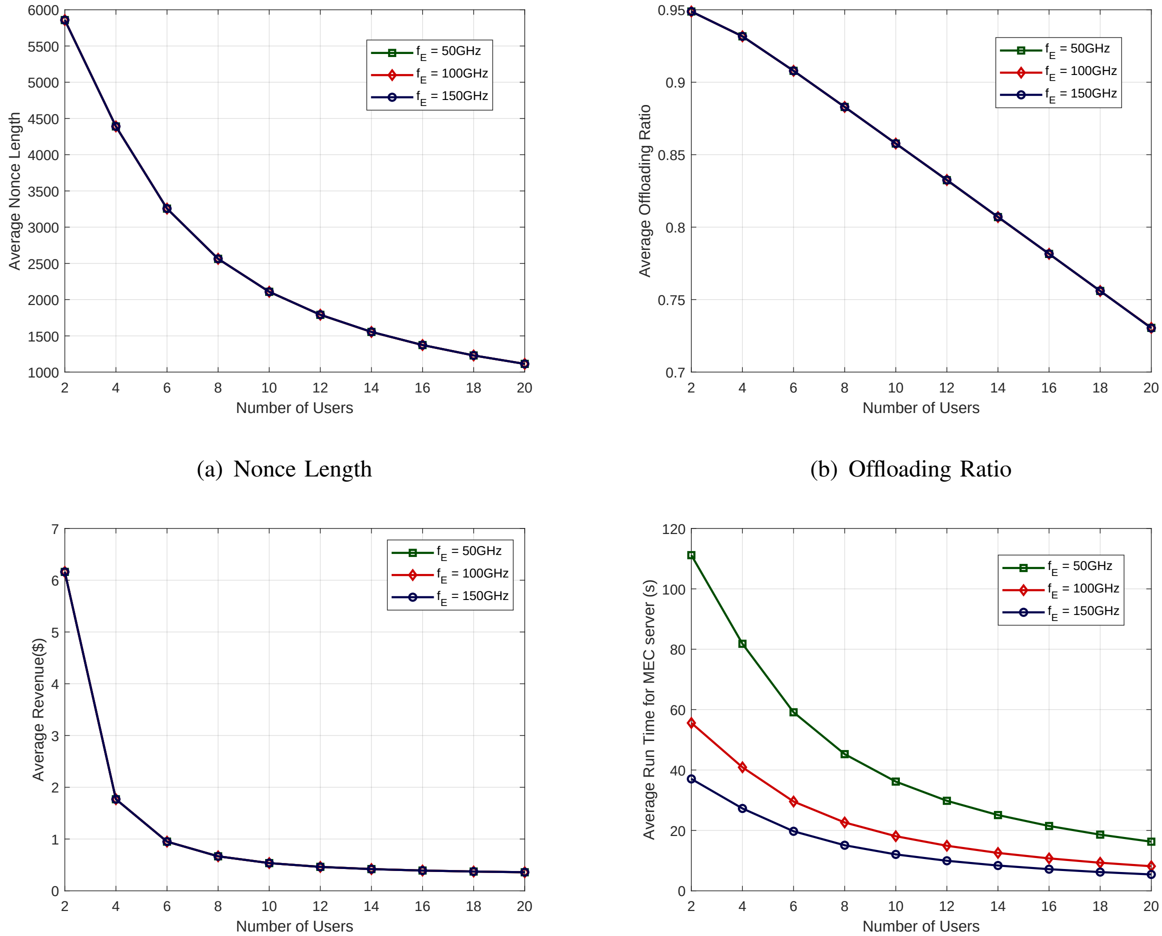


(c) Individual Revenue

Fig. 3. Convergence performance of the proposed CRGR-based alternating iterative algorithm. CPU frequencies of all three users and MEC server are $f_1 = f_2 = f_3 = 0.5$ GHz and $f_E = 100$ GHz. The block sizes of all three users are $s_1 = 1000$, $s_1 = 2000$, and $s_3 = 3000$, respectively. (a) The individual nonce length versus the number of iteration. (b) The offloading ratio to the MEC server versus the number of iteration. (c) The individual revenue versus the number of iteration.

B. Impacts of Different Parameters

In this subsection, we chiefly examine that effects of several parameters including the number of users, CPU frequency of the individual user and the MEC server, and the block size on



(a) Nonce Length

(b) Offloading Ratio

(c) Average Revenue

(d) Delay for MEC Server

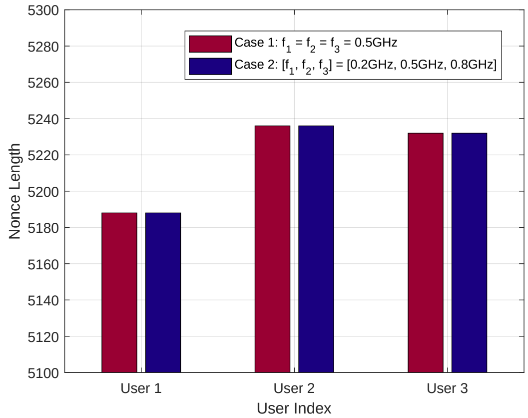
Fig. 4. The effect of the number of users and CPU frequency of the MEC server. CPU frequency and the block size of all users are $f_1 = f_2 = ... = f_N = 0.5$ GHz and $s_1 = s_2 = ... = s_N = 2000$. CPU frequency of the MEC server takes 50 GHz, 100 GHz, and 150 GHz, respectively. (a) The average nonce length versus the number of users. (b) The average offloading ratio to the MEC server versus the number of users. (c) The average revenue versus the number of users. (d) The average run time for the MEC server versus the number of users.

computation offloading and transmit power resource allocation are represented in Fig. 4, Fig. 5, and Fig. 6, respectively.

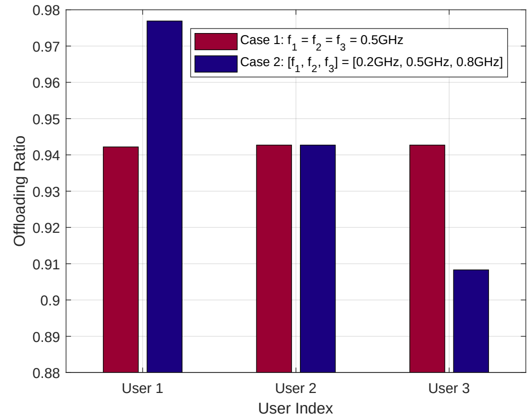
1) Impacts of the Number of Users & CPU Frequency of MEC Server: Fig. 4 discusses the effects of the number of users and CPU frequency of MEC server on the average nonce length, offloading ratio, revenue, and mining delay of the MEC server in Fig. 4(a), Fig. 4(b), Fig. 4(c) and Fig. 4(d), respectively. Here, we assume that CPU frequency and the block size of all users are 0.5 GHz and 2000. We take the CPU frequency of the MEC server as 50 GHz, 100GHz, 150GHz, respectively. Observing Fig. 4, we find that the average computation resource allocation (i.e. nonce length and offloading ratio), revenue, and mining delay of MEC server all decrease with the increment of N. Meanwhile, the average revenue of all users tends towards stability with the continuous growth of N. The reason is that more users in the MEC-aided mobile blockchain systems, the mining competition between users becomes more fierce, and the corresponding allocated computation resources are going to reduce. The mining computation task assigned to the MEC server decreases and the delay cuts down accordingly.

Fig. 4(c) shows that the downward trend of the average revenue is getting slower, when the number of users is larger. Fig. 4(d) indicates an obvious result that the mining delay of the MEC server decreases with the increment of f_E. We can also observe from Fig. 4(d) that the decline of the mining delay is slowing down with the increase of f_E. For example, when $N = 2$, the mining delay drops from 106.16s of $f_E = 50$ GHz to 53.08s of $f_E = 100$ GHz. We can easily get that the mining delay is going to drop by has 53.08s when f_E increases from 50 GHz to 100 GHz. From Fig. 4(d), the mining delay changes from 53.08s of $f_E = 100$ GHz to 35.39s of $f_E = 150$ GHz. Then we can obtain that the mining delay will decrease by 17.69s when f_E increases from 100 GHz to 150 GHz. From Fig. 4(a)-4(c), we can observe that the computation resource allocation and revenue are independent of the CPU frequency of the MEC server. The phenomenon is relevant to our proposed MEC-assisted mobile blockchain system model, and we consider only one MEC server, which has no competition with other MEC servers. Thus, the change of f_E does not substantially transform the computation resource allocation and revenue of our proposed MEC-assisted mobile blockchain systems in this paper.

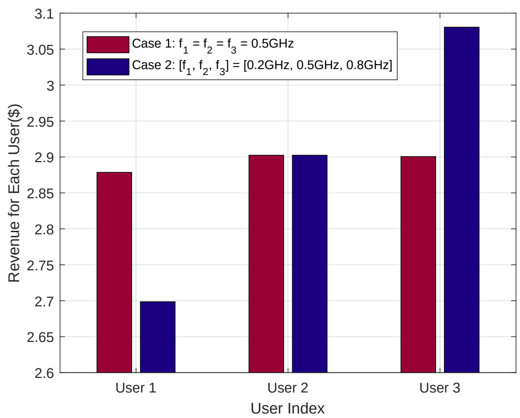
2) Impact of CPU Frequency of Individual User: Fig. 5 shows effects of CPU frequency of the individual user on the computation resource management, revenue and transmit power for each user, where the number of users is set to be $N = 3$, CPU frequency of MEC server is 100 GHz, and the block size of all users is $s_1 = s_2 = s_3 = 2000$. As shown in Fig. 5(a), the nonce length in Case 1 and Case 2 is the same for all three users, which means that the nonce length selection strategy has nothing to do with the CPU frequency of the individual user. However, under the same conditions, the nonce length strategy of users 1 and user 3 is slightly



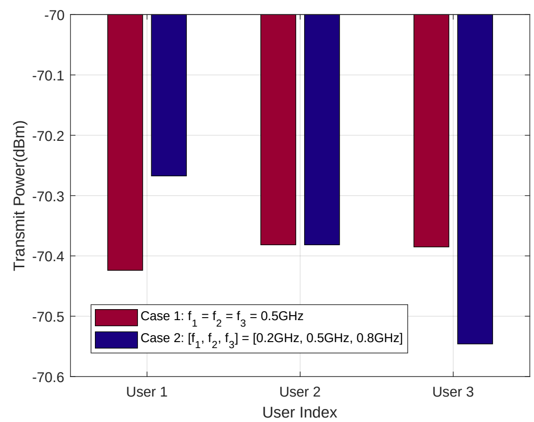
(a) Nonce Length



(b) Offloading Ratio



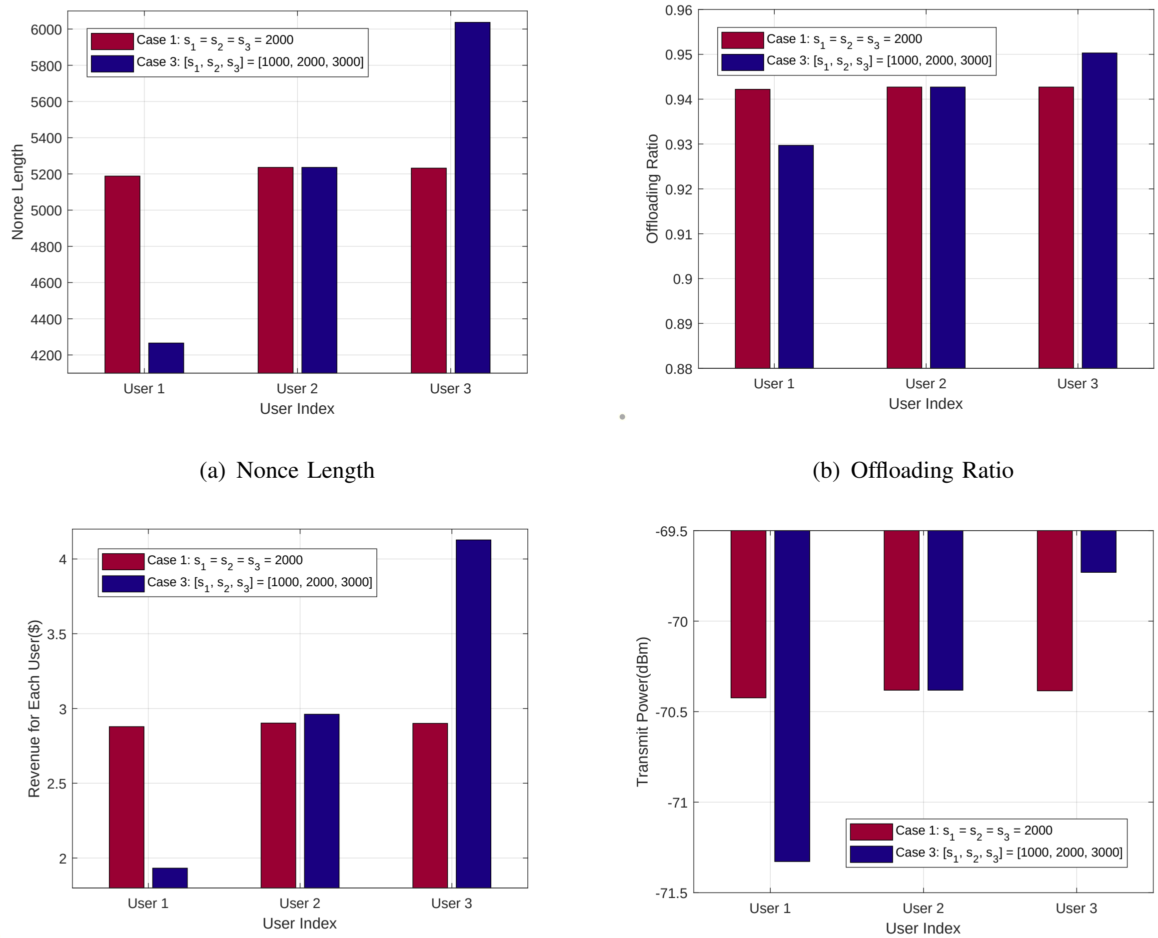
(c) Individual Revenue



(d) Transmission Power

Fig. 5. The effect of CPU frequency of the individual user. Case 1: CPU frequencies of all three users and MEC server are $f_1 = f_2 = f_3 = 0.5$ GHz and $f_E = 100$ GHz, respectively. The block size of all three users is $s_1 = s_2 = s_3 = 2000$; Case 2: CPU frequencies of the MEC server and the block size are $f_E = 100$ GHz and $s_1 = s_2 = s_3 = 2000$. CPU frequencies of three users are $f_1 = 0.2$ GHz, $f_2 = 0.5$ GHz, and $f_3 = 0.8$ GHz, respectively. (a) The individual nonce length in Case 1 and Case 2. (b) The individual offloading ratio to the MEC server in Case 1 and Case 2. (c) The individual revenue in Case 1 and Case 2. (d) The individual transmission power in Case 1 and Case 2.

less than that of user 2. This shows that our proposed algorithm cannot provide a completely fair computation resource allocation. The reason for this phenomenon may be related to the order



(a) Nonce Length

(b) Offloading Ratio

(c) Individual Revenue

(d) Transmit Power

Fig. 6. The effect of the block size of the individual user. Case 1: CPU frequencies of all three users and MEC server are $f_1 = f_2 = f_3 = 0.5$ GHz and $f_E = 100$ GHz, respectively. The block size of all three users is $s_1 = s_2 = s_3 = 2000$; Case 3: CPU frequencies of all three users and MEC server are $f_1 = f_2 = f_3 = 0.5$ GHz and $f_E = 100$ GHz, respectively. The block sizes of three users are $s_1 = 1000$, $s_2 = 2000$, and $s_3 = 3000$, respectively. (a) The individual nonce length in Case 1 and Case 3. (b) The individual offloading ratio to the MEC server in Case 1 and Case 3. (c) The individual revenue in Case 1 and Case 3. (d) The individual transmission power in Case 1 and Case 3.

of users running Algorithm 1. Because users in our proposed MEC-assisted mobile blockchain networks are in the non-cooperative competitive relationship with each other. The first or last user enters this system to offload computing requests, other users may know their private information, such as orders and offloading requests. Therefore, users in the middle order have relatively less privacy during the computation offloading process, which leads to a greater nonce length selection strategy for these users. The offloading ratio to the MEC server declines when the user's CPU frequency enhances in Fig. 5(b). Within the maximum mining delay range, in order to save the cost charged by the MEC server for nonce hash computing service of PoW mining tasks, users choose more mining tasks to calculate locally when the user's local computing power becomes stronger. From Fig. 5(c), we can see that the individual revenue raises as the individual user's CPU frequency increases. For example, the individual revenue of user 1 drops from 2.88 $ to 2.70 $ when the CPU frequency of user 1 decreases from 0.5 GHz to 0.2 GHz. On the contrary, the individual revenue of user 3 rises from 2.90 $ to 3.08 $ when the CPU frequency of user 3 increases from 0.5 GHz to 0.8 GHz. Combining Fig. 5(a) and 5(b), the greater CPU frequency of the individual user, the smaller offloading ratio to MEC server, the lower fee paid to the MEC server, and the corresponding individual revenue is going to become larger. From Fig. 5(d), the individual transmission power declines when the CPU frequency of the individual user increases. Combined with Fig. 5(b), fewer PoW mining tasks need to be offloaded to the MEC server and the corresponding transmitted power becomes smaller.

3) Impact of Block Size: In the part, Fig. 6 plots the effect of block size on the computation resource allocation, revenue and transmit power for each user in Fig. 6(a), Fig. 6(b), Fig. 6(c), and Fig. 6(d), respectively. We assume that the number of user is $N = 3$, CPU frequency of MEC server and all users are $f_E = 100$ GHz and $f_1 = f_2 = f_3 = 0.5$ GHz. As shown in Fig. 6, we can see that the nonce length, offloading ratio, individual revenue, and transmit power all grow with the enlargement of the block size. This is because as the block size becomes larger, the transaction fee charged for successfully mining a new block increases, which offers incentives to the user mining a new block and causes the mobile user to choose more nonces in order to win the mining victory in Fig. 6(a). As shown in Fig. 6(a), for example, the nonce length of user 1 drops from 5188 to 4266 when its block size decreases from 2000 to 1000. Oppositely, the nonce length of user 3 increases from 5232 to 6037 with the increasing block size of user 3 from 2000 to 3000. Within a certain mining delay range, when the PoW mining task has become larger in Fig. 6(a), the offloading proportion to the MEC server grows in Fig. 6(b), and the corresponding transmit power will also raise in Fig. 6(d). The individual revenue increases in Fig. 6(c) with the augmentation of the block size, the reason is that the block size becomes larger, the increasing transaction fee is greater than the expense paid to the MEC server for hash computing services.

VI. Conclusion

In this paper, we have considered the novel MEC-assisted mobile blockchain networks, where plenty of PoW mining tasks can be offloaded to the MEC server. In this network, we formulated the individual computation resource distribution strategies as a non-cooperative game. The existence of NE was also proved in the non-cooperative game. To obtain optimal delay-limited computation resource allocation strategies for all mobile users, we designed a CRGR-based alternating iterative algorithm. We also derived the closed-form expression of the optimal transmission power for the individual user within the maximum mining delay range. We have efficiently achieved optimal delay-limited computation offloading and transmit power strategies for all users by using the proposed CRGR-based alternating iterative algorithm. The individual transmission power extends accordingly with the optimal computation resource allocation strategies for all users. Numerical results indicate that our proposed CRGR-based alternating iterative algorithm has a fast convergence rate and different parameters, such as the number of users, block size, and CPU frequency of MEC server, have great influences on the system performance of our proposed delay-limited mobile blockchain networks. Additionally, we have conducted the numerical experiments to evaluate the system performance to achieve optimal delay-limited nonce length, offloading ratio to the MEC server, and transmit power strategies for all mobile users.

References

[1] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," Oct. 2008.

[2] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, 2019.

[3] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5G and beyond networks: A state of the art survey," *arXiv:1912.05062*, Dec. 2019.

[4] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, Jan. 2019.

[5] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *arXiv:1906.00245*, Jun. 2019.

[6] B. Cao, *et al.*, "When internet of things meets blockchain: challenges in distributed consensus," *arXiv:1905.06022*, May 2019.

[7] P. Danzi, A. E. Kalor, C. Stefanovic, and P. Popovski, "Delay and communication tradeoffs for blockchain systems with lightweight IoT clients," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2354–2365, Apr. 2019.

[8] K. Kotobi and S. G. Bilen, "Blockchain-enabled spectrum access in cognitive radio networks," in *Proc. Int. Wireless Commun. Symp. (WTS)*, Apr. 2017, pp. 1–6.

[9] S. Bayhan, A. Zubow, and A. Wolisz, "Spass: Spectrum sensing as a service via smart contracts," in *IEEE Int. Symp. on Dyna. Spec, Access Net. (DySPAN)*, Oct. 2018, pp. 1–10.

[10] S. Ding, K. X. Pan, S. K. Bose, Q. Zhang, and G. Shen, "Blockchain-assisted spectrum trading between elastic virtual optical networks," *arXiv:1911.08802*, Nov. 2019.

[11] A. El Gamal and H. El Gamal, "A single coin monetary mechanism for distributed cooperative interference management," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 757–760, Jun. 2019.

[12] Z. Xiong, *et al.*, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, pp. 1–1, Sept. 2018.

[13] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distrib. Syst.*, pp. 1–1, Mar. 2019.

[14] M. Liu, F. R. Yu, Y. Teng, *et al.*, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11 008–11 021, Aug. 2018.

[15] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, Mar. 2017.

[16] W. Yu, *et al.*, "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, pp. 6900–6919, Nov. 2018.

[17] W. Wen, Y. Cui, F.-C. Zheng, S. Jin, and Y. Jiang, "Enhancing performance of random caching in large-scale heterogeneous wireless networks with random discontinuous transmission," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6287–6303, Aug. 2018.

[18] G. Qiao, S. Leng, H. Chai, A. Asadi, and Y. Zhang, "Blockchain empowered resource trading in mobile edge computing and networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[19] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K.-K. R. Choo, "Neural-blockchain-based ultrareliable caching for edge-enabled UAV networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5723–5736, Jun. 2019.

[20] D. Lin, S. Hu, Y. Gao, and Y. Tang, "Optimizing MEC networks for healthcare applications in 5G communications with the authenticity of users priorities," *IEEE Access*, vol. 7, pp. 88 592–88 600, Jun. 2019.

[21] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Secure computation offloading in blockchain based IoT networks with deep reinforcement learning," *arXiv:1908.07466*, Aug. 2019.

[22] ——, "Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning," *arXiv:1908.07467*, Aug. 2019.

[23] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: A secure, decentralized and privacy-preserving system," *arXiv:1906.10893*, Jun. 2019.

[24] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, pp. 1–1, Apr. 2020.

[25] Y. Zuo, S. Jin, and S. Zhang, "Computation offloading in the untrusted MEC-aided mobile blockchain IoT system," *arXiv:1911.08255*, Nov. 2019.

[26] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge university press, 2012.

[27] W. Wen, Y. Fu, T. Q. S. Quek, F. Zheng, and S. Jin, "Joint uplink/downlink sub-channel, bit and time allocation for multi-access edge computing," *IEEE Wireless Commun. Lett.*, vol. 23, no. 10, pp. 1811–1815, Jul. 2019.

[28] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.