Research Article

Cost-Effective Optimization for Blockchain-Enabled NOMA-Based MEC Networks

Jianbo Du,¹ Yan Sun,¹ Aijing Sun ,¹ Guangyue Lu ,¹ Zhixian Chang,¹ Haotong Cao,² and Jie Feng³

¹Shaanxi Key Laboratory of Information Communication Network and Security,

School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China

²Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing, China

³State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, Xi'an, China

Correspondence should be addressed to Aijing Sun; sunaijing@xupt.edu.cn and Guangyue Lu; tonylugy@163.com

Received 7 April 2021; Revised 29 May 2021; Accepted 21 June 2021; Published 12 July 2021

Academic Editor: Mohamed Amine Ferrag

Copyright © 2021 Jianbo Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Blockchain technology has been widely used in many fields. However, the proof of work (PoW) problem in the mining process of mobile devices requires a large amount of computing resources and energy consumption, which brings huge challenges to mobile devices. Mobile edge computing (MEC) can effectively solve the above problems, allowing mobile devices to offload tasks to edge servers to relieve the pressure of limited computing resources on mobile devices. Nonorthogonal multiple access (NOMA) is good at improving spectrum efficiency, so that the system can accommodate more users. In this paper, we propose a new NOMA-based MEC-enabled blockchain framework. Under the conditions of a given task execution deadline, the decision of offloading, local computing resource allocation, user clustering and admission control, and transmit power control is jointly optimized to minimize the total cost of the system. Since the problem is hard to solve, we decouple it into subproblems for low-complexity solutions. First, we propose two heuristic algorithms to obtain the binary offloading decision and user association, and then closed-form solutions of local resource allocation and transmit power control are obtained under the required delay constraints. Simulation results show that our proposed algorithms perform good in cost reduction compared with other baseline algorithms.

1. Introduction

In recent years, smart user devices, Internet of Things (IoT) devices, and their running smart applications have been widely popularized, and blockchain technology has also been widely used in various industrial applications [1], such as IoT and healthcare. Blockchain technology uses blockchain data structures to verify and store data, uses distributed node consensus algorithms to generate and update data, and uses cryptography to ensure the security of data transmission and access. Its essence is a decentralized database, and the core problem is to solve mining puzzles. Blockchain users or miners need to run a mining puzzle, that is, to solve the proof of work (PoW) problem, which is a computing process that requires intensive computing

resources. However, mobile devices have limited resources and are usually unable to participate in the mining process. Mobile edge computing (MEC) [2] provides an effective solution to address the challenge. It offloads the computing tasks of mobile devices to the edge of the wireless network [3], thereby improving computing efficiency and reducing delay. Meanwhile, the explosively increasing IoT user equipments make the scarce wireless resources further strained, which degrades the task offloading performances of MEC. In recent years, nonorthogonal multiple access (NOMA) [4, 5] has attracted lot of attention in the MEC domain for enhancing the performance of computation offloading. With NOMA, a group of users can offload their own tasks simultaneously using the same channels, and thus each user can enjoy more wireless channels [6] and/or

time slots [7], and the system could be able to accommodate more users [8].

On the one hand, many literature studies have proposed various methods to solve the offloading problem in the mobile blockchain networks, such as the method based on convex optimization. In [9], the authors offloaded some computationally intensive tasks to an untrusted MEC server and proposed a hash computing ordering (HCO) mechanism to enable the untrusted MEC server to provide a fairer distribution of computing resources for all mobile users. In [10], the authors proposed a blockchain-enabled IoT-edge-cloud computing architecture that benefits both from mobile cloud computing (MCC) and MEC, where MEC servers offer lower latency computing services, while MCC servers provide stronger computation power. In [11], the authors studied the calculation of moving edges for blockchain mining tasks. Based on Stackelberg theory, the pricing and consumption of computing resources are attributed to leader-follower coupling game, and an efficient offloading algorithm for mining tasks is proposed.

On the other hand, several studies have investigated the NOMA-based MEC system, where energy consumption minimization for user equipment is always the first concern. In [12], the authors formulated a joint user clustering and transmit power control problem and proposed a matching-coalition approach for problem solving. In [13], the authors applied NOMA to enable massive connectivity to support energy-efficient MEC in IoT networks by joint communication and computation resource allocation optimization. In [14], the authors studied task processing delay minimization issues in a multiuser NOMA-based MEC network by optimizing tasks partition ratios and transmit power control. In [15], the authors studied the energy consumption reduction in a NOMA-based MEC system by jointly optimizing the transmit time and power.

From the above works, we find that there is still a lack of research on the optimal strategy for implementing mining task offloading in mobile blockchain. Therefore, this paper considers the joint optimization of data offloading and resource allocation to improve the offloading efficiency in the blockchain network. In addition, user clustering and resource allocation [16, 17] optimization are important factors in NOMA-based MEC system, and users' energy consumption minimization is the most general goal of the optimization. The above works only considers user clustering, resource allocation, and power control issues, in which all users offload their tasks to the MEC server, without considering the optimization of task offloading decisions. In most IoT scenarios, this is not reasonable, as the local processing power is not used and the capacity of the MEC server is limited; thus, the task processing fails when the user is not received. Moreover, when the number of IoT devices is too huge, then local and MEC server could not afford, and hence, cloud is necessary. Motivated by the above considerations, we consider the joint optimization of offloading decision, user clustering and admission control, and computation resource allocation and transmit power control in a NOMA-based mixed edge and cloud-enabled blockchain network. We intend to minimize the energy consumption in

local and MEC processing, minimize the economical pay in cloud processing mode, and maximize mining profits. Most importantly, we ensure all users' tasks are processed successfully, i.e., no user will fail in task processing. The formulated problem includes binary, integer, and continuous variables, so it is hard to solve.

For this purpose, we devote to propose low-complexity algorithm for easy implementation. First of all, we use two heuristic algorithms to solve binary offload decision and user association, respectively, and then obtain the closed-form solutions of local resource allocation. In addition, the closed-form solutions of MEC transmission power control and cloud center power control are solved under the premise of satisfying the delay constraint. Simulation results show that the low-complexity algorithm proposed by us can effectively reduce the system cost compared with other simplified methods.

The remainder of this paper is organized as follows. In Section 2, the system model is presented and the problem is formulated. Section 3 provides the offloading decision optimization. Admission control and user clustering are presented in Section 4; the local computation resource allocation is developed in Section 5, and Section 6 presents the transmit power control. Section 7 develops the simulation results and discussion. Section 8 concludes the paper.

2. System Model

We consider a NOMA-based MEC and cloud-enabled blockchain network, consisting of M access points (APs), N randomly distributed users, and a remote cloud center. Each AP is equipped with a MEC server by wired link, as shown in Figure 1. In the following, the MEC server m and AP m are used interchangeably. Denote $\mathcal{N} = \{1, 2, \dots, N\}$ is the set of users and $\mathcal{M} = \{1, 2, \dots, M\}$ is the set of APs. We assume that the users who offload their tasks to a MEC server m via NOMA [18, 19] form a group because of each user can offload its task to a MEC server. Let $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$ denotes the set of groups, and the maximum number of users in each group that can be served by an AP is K . So, all APs could accommodate at most MK users. In particular, we assume that each group is assigned an orthogonal sub-channel with bandwidth B . The same group of users offloads their tasks to the AP over the same subchannel. Lastly, we assume that the MEC servers and the APs have complete channel status information (CSI).

Especially, there is a remote cloud center, and users can also offload tasks to the cloud center. The cloud has enough computing resources and wireless bandwidth for users, but when a user chooses to offload the task to it, it charges for the use of bandwidth and computing resources.

Each user acts as a miner, running a blockchain application to record transactions performed in the network [20]. Denote $\Lambda_n = \{D_n, \lambda_n, T_n^{\max}\}$ as the task of miner n . Each task is characterized by the input size D_n [21], i.e., the number of bits that must be transferred from the user to the AP, and the workload λ_n , i.e., the processing density (in CPU cycles per bit). $T_n^{\max}$ is the deadline to complete the task. Then, we have $C_n = D_n * \lambda_n$, where C_n is the size of processing workload

FIGURE 1: System model.

(in CPU cycles). We assume that all APs and miners in the blockchain network supported by NOMA-based MEC have the computing power to provide task execution services. Mining tasks that cannot be calculated locally by the miner can be offloaded to nearby access points. Therefore, each miner can execute tasks locally or offload to the MEC server m (AP m) or cloud center for execution. We use x_n , y_n , and z_n to denote the task of miner n is processed by miner n itself, or by MEC server m , or the cloud center, respectively [3, 22].

2.1. Local Processing. When the task is processed locally, denote the local processing capability of user n is $f_n^{\max}$, we have

$$\begin{aligned} T_n^{\text{loc}} &= \frac{D_n}{f_n^{\text{loc}}} \gamma_n, \\ E_n^{\text{loc}} &= \alpha D_n \gamma_n f_n^{\text{loc}^2}, \end{aligned} \quad (1)$$

where f_n^{loc} is the local computation resource allocation, which is constrained by $f_n^{\max}$, and γ_n depends on the nature of application, e.g., computational complexity of application. α is a coefficient.

2.2. MEC Processing. In the area, there are M MEC servers providing services for users. Thus, in the MEC processing model, each user could offload its task to any one of them. Suppose user n offloads its task to MEC server m , the task transmit rate is given by

$$r_{n,m} = v_{n,m} B \log_2 \left(1 + \frac{p_n^{\text{mec}} g_{n,m}}{\sum_{g_s < g_n} v_{s,m} p_s^{\text{mec}} g_{s,m} + N_0} \right), \quad (2)$$

where p_n^{mec} is the transmit power of user n when offloading to MEC server and the binary variable $v_{n,m}$ is the access control, where $v_{n,m} = 1$ indicates user n is served by MEC m and $v_{n,m} = 0$ otherwise. Based on $r_{n,m}$, the transmit delay and the energy consumption of user n are given by

$$\begin{aligned} T_n^{\text{tra}} &= \frac{D_n}{r_{n,m}}, \\ E_n^{\text{tra}} &= \frac{D_n}{r_{n,m}} p_n^{\text{mec}}. \end{aligned} \quad (3)$$

When the task is processed by the MEC server, the user will wait for the result. The task processing delay and the energy consumption of user n in the process are given by

$$\begin{aligned}
T_n^{\text{pro}} &= \frac{C_n}{F_m}, \\
E_n^{\text{wait}} &= \frac{C_n}{F_m} p_n^{\text{idle}},
\end{aligned} \tag{4}$$

where p_n^{idle} is the power consumption of user n when waiting and F_m is the computation resource allocated by MEC server m , which is a constant in this paper. The delay and energy consumption in MEC processing model are given by

$$\begin{aligned}
T_n^{\text{mec}} &= v_{n,m} (T_n^{\text{tra}} + T_n^{\text{pro}}), \\
E_n^{\text{mec}} &= v_{n,m} (E_n^{\text{tra}} + E_n^{\text{wait}}).
\end{aligned} \tag{5}$$

Users who have tasks offloaded to the MEC server will get extra profit. The profit for completing the mining task is given by

$$\phi_n = v_{n,m} (\zeta - \eta E_n^{\text{mec}}), \tag{6}$$

where ζ (\$) is mining rewards and η (in \$/J) is the unit price of energy consumption.

2.3. Cloud Processing. When user n offloads its task to the cloud center, it will be allocated with exclusive bandwidth B_1 , and processing resource F_n^{cloud} . The transmit rate of user n in cloud processing mode is given by

$$r_n^{\text{cloud}} = B_1 \log_2 \left(1 + \frac{P_n^{\text{cloud}} g_n^{\text{cloud}}}{N_0} \right). \tag{7}$$

The delay in cloud processing mode can be given by

$$\begin{aligned}
T_n^{\text{cloud}} &= T_n^{\text{cloud,tra}} + T_n^{\text{cloud,pro}} \\
&= \frac{D_n}{r_n^{\text{cloud}}} + \frac{C_n}{F_n^{\text{cloud}}}.
\end{aligned} \tag{8}$$

The cloud operator will charge users for using communication and computation resources, where the prices are λ (\$/bps) and ω (\$/(CPU cycles per second)), respectively.

Therefore, the cost paid by user n in communication and task processing are given by

$$\begin{aligned}
\text{Pay}_n^{\text{tra}} &= \lambda r_n^{\text{cloud}}, \\
\text{Pay}_n^{\text{pro}} &= \omega F_n^{\text{cloud}}.
\end{aligned} \tag{9}$$

2.4. Total Delay and Total Cost. Based on the above delays in different modes, the total delay of user n is given by

$$T_n = x_n T_n^{\text{loc}} + y_n T_n^{\text{mec}} + z_n T_n^{\text{cloud}}. \tag{10}$$

The total cost of user n is defined as

$$\text{Cost}_n = \mu_e (x_n E_n^{\text{loc}} + y_n E_n^{\text{mec}}) + \mu_{\text{pay}} z_n \text{Pay}_n - y_n \phi_n, \tag{11}$$

where μ_e (in \$/J) and μ_{pay} (a dimensionless number) are the weight parameters.

2.5. Problem Formulation. In this paper, we intend to minimize the total system cost, i.e., the energy consumption for local and MEC processing, economic payments in cloud processing mode, and additional profits from mining tasks for miners. In particular, minimizing economic payments has a higher priority. We achieve the purpose by jointly optimizing the offloading decision $\Pi = \{x_n, y_n, z_n\}$, the access control and user clustering $\mathbf{V} = \{v_{n,m}\}$, the local computation resource allocation $\mathbf{f}^{\text{loc}} = \{f_n^{\text{loc}}\}$, and the transmit power control in MEC and cloud processing modes, i.e., $\mathbf{p}^{\text{mec}} = \{p_n^{\text{mec}}\}$ and $\mathbf{p}^{\text{cloud}} = \{p_n^{\text{cloud}}\}$, respectively. The problem is formulated as

$$\begin{aligned}
(\mathcal{P}_1): \quad & \min_{\Pi, \mathbf{v}, \mathbf{f}^{\text{loc}}, \mathbf{p}^{\text{mec}}, \mathbf{p}^{\text{cloud}}} \sum_{n \in \mathcal{N}} \text{Cost}_n \\
\text{s.t. (C1): } & T_n \leq T_n^{\text{max}}, \quad \forall n \in \mathcal{N}, \\
\text{(C2): } & 0 \leq f_n^{\text{loc}} \leq f_n^{\text{max}}, \quad \forall n \in \mathcal{N}, \\
\text{(C3): } & 0 \leq p_n^{\text{mec}} \leq p_n^{\text{max}}, \quad \forall n \in \mathcal{N}, \\
\text{(C4): } & 0 \leq p_n^{\text{cloud}} \leq p_n^{\text{max}}, \quad \forall n \in \mathcal{N}, \\
\text{(C5): } & x_n, y_n, z_n \in \{0, 1\}, \quad \forall n \in \mathcal{N}, \\
\text{(C6): } & x_n + y_n + z_n \leq 1, \quad \forall n \in \mathcal{N}, \\
\text{(C7): } & v_{n,m} \in \{0, 1\}, \quad \forall n \in \mathcal{N}, \forall m \in \mathcal{M}, \\
\text{(C8): } & \sum_{n \in \mathcal{N}} v_{n,m} \leq K, \quad \forall m \in \mathcal{M},
\end{aligned} \tag{12}$$

where (C1) is the task completion deadline; (C2) is the constraint on local computation resource allocation; (C3)

and (C4) are the constraints on transmit power control in MEC and cloud processing, respectively; (C5) and (C6)

require that each user task can only be processed in one place, i.e., local, or a certain MEC server, or the cloud center; (C7) is the binary constraint on access control, and (C8) requires that the maximum number of users per AP service should not exceed K . Problem $(\mathcal{P}_1)$ contains binary, integer, and continuous variables, which is generally nonconvex and hard to solve. In the following sections, we will propose low-complexity algorithms to solve it.

3. Offloading Decision Optimization

In this section, we propose a low-complexity algorithm to obtain the offloading decision-making $\Pi = \{x_n, y_n, z_n\}, n \in \mathcal{N}$, as shown in Algorithm 1.

Since the tasks processed on cloud will be charged for fees, so we encourage users as much as possible to process their tasks either locally by themselves or offload their tasks to the MEC servers. Moreover, for the noncloud processing tasks, we encourage as many users to offload their tasks to the MEC servers for energy saving. Our main idea can be briefed as the following steps:

- (i) We divide all the users in $\mathcal{N}$ into two subsets, local feasible set $\mathcal{N}_0$ and local infeasible set $\mathcal{N}_1$.
- (ii) Next, we judge whether $N_1 \leq MK$. If yes, this means the number of local infeasible users is smaller than the capacity that the MEC servers can afford. Therefore, let all the users in $\mathcal{N}_1$ offload their tasks to MEC processing and there's no need to offload tasks to the cloud server. Moreover, in order to save energy for users and thus to decrease the system energy consumption while guaranteeing latency requirement, it is necessary to saturate the number of users of the MEC servers as much as possible. The MEC server with the best channel gain, i.e., g_n , will reduce the data transmission delay, the user with a larger workload, i.e., C_n , means that the task is more complex and less suitable for local processing, users with smaller data transmission, i.e., D_n means lower transmission delay. Combining these three factors, we define a new parameter $G_n = (g_n C_n / D_n)$. The larger G_n , the less suitable the task of user n is to be processed locally. Therefore, according to the size of G_n , move $MK - N_1$ users from local feasible set $\mathcal{N}_0$ out to MEC processing.
- (iii) If $N_1 > MK$, this means some tasks have to be offloaded for cloud processing, and users have to pay some fees for task processing. We should pick out MK users to MEC processing, and the rest $N_1 - MK$ tasks will be offloaded to the cloud.

Performing the above steps, the offloading decision can be obtained. The proposed offloading decision-making algorithm is summarized in Algorithm 1.

4. Admission Control and User Clustering

After offloading decision-making, MK users are picked out, and their tasks will be processed by the M MEC

servers. Next, we will allocate the MK users to the M MEC servers.

Since we intend to minimize the energy consumption of MEC processing, a new parameter $\Gamma_{n,m} = g_{n,m} F_m$ is defined. The MEC server M with the best channel gain, i.e., $g_{n,m}$, will reduce the data transmission delay, and the server with the most computation resource, i.e., F_m , will reduce the task processing delay. Thus, the larger the $\Gamma_{n,m}$, the user n is more would like to access MEC server m , the smaller the delay in processing its tasks, and the smaller the power consumption. Based on the above idea, our admission control and user clustering algorithm are detailed in Algorithm 2.

After Algorithms 1 and 2 are performed, the offloading decision, admission control and user clustering are obtained, i.e., the user set is divided into several subsets, including (i) the local processing set $\mathcal{N}_{loc}$, where the offloading decision of tasks n is $x_n = 1$; (ii) M MEC processing subsets, and in each subset $\mathcal{G}_m$, the offloading decision $y_n = 1$, and the admission control $v_{n,m} = 1$, and all users in each subset $\mathcal{G}_m$ will offload their tasks to MEC server m sharing the same bandwidth via NOMA; (iii) the cloud processing set $\mathcal{N}_{cloud}$, where each user in this set will offload their tasks for cloud processing.

For the users in $\mathcal{N}_{loc}$, we need to perform local computation resource allocation optimization $\mathbf{f}^{loc} = \{f_n^{loc}\}$; for the users in each $\mathcal{G}_m$ and in $\mathcal{N}_{cloud}$, we need to perform transmit power control optimization, i.e., we need to determine $\mathbf{p}^{mec} = \{p_n^{mec}\}$ and $\mathbf{p}^{cloud} = \{p_n^{cloud}\}$, respectively.

5. Local Computation Resource Allocation

For the users in $\mathcal{N}_{loc}$, we need to perform local computation resource allocation for local task processing energy consumption minimization, and we have

$$\begin{aligned}
 (\mathcal{P}_2): \min_{\mathbf{f}^{loc}} \sum_{n \in \mathcal{N}} E_n^{loc} \\
 \text{s.t. (C1): } T_n^{loc} \leq T_n^{\max}, \quad \forall n \in \mathcal{N}, \\
 \text{(C2): } 0 \leq f_n^{loc} \leq f_n^{\max}, \quad \forall n \in \mathcal{N}.
 \end{aligned} \tag{13}$$

For user n , given the data input size D_n , a coefficient α and computational complexity of application γ_n , the energy consumption is only affected by the local computing resource allocation. In order to minimize the energy consumption while ensuring the delay requirements, let $T_n^{loc} = T_n^{\max}$, so that we only need the minimum computing resources that meet the latency requirements, and get f_n^{loc} .

6. Transmit Power Control

After offloading decision-making, access control and user clustering, and local computation resource allocation are obtained, problem $(\mathcal{P}_1)$ degrades into the transmit power control optimization in MEC processing and cloud processing, respectively. For the users who choose MEC processing, we need to perform transmit power control among the users in a same cluster as follows:


```

(1) Initialization: set  $\mathcal{N}_0 = \emptyset, \mathcal{N}_1 = \emptyset, \mathcal{N}_{\text{temp}} = \mathcal{N}, N_0 = 0, N_1 = 0, \mathcal{N}_{\text{loc}} = \emptyset, \mathcal{N}_{\text{MEC}} = \emptyset, \mathcal{N}_{\text{cloud}} = \emptyset$ 
(2) for  $n \in \mathcal{N}$  do
(3)   if  $(C_n/f_n^{\max}) \leq T_n^{\max}$  then
(4)      $\mathcal{N}_0 = \mathcal{N}_0 \cup n, N_0 = N_0 + 1$ , remove  $n$  from  $\mathcal{N}_{\text{temp}}$ 
(5)   else
(6)      $\mathcal{N}_1 = \mathcal{N}_1 \cup n, N_1 = N_1 + 1$ , remove  $n$  from  $\mathcal{N}_{\text{temp}}$ 
(7)   end if
(8) end for
(9) if  $N_1 \leq \text{MK}$  then
(10)   $\mathcal{N}_{\text{MEC}} = \mathcal{N}_1, N_{\text{MEC}} = N_1, N_{\text{gap}} = \text{MK} - N_1$ 
(11)  for  $n \in \mathcal{N}_0$  do
(12)     $g_n = \sum_{m \in \mathcal{M}} g_{n,m}/M$ 
(13)     $G_n = g_n C_n / D_n$ 
(14)  end for
(15)  Sort the  $N_0$  users in ascending order according to  $G_n$ 
(16)  Remove the end  $N_{\text{gap}}$  users from  $\mathcal{N}_0$  to  $\mathcal{N}_{\text{MEC}}$ , and we set  $y_n = 1, n \in \mathcal{N}_{\text{mec}}$ 
(17)  Remove the rest users in  $\mathcal{N}_0$  to  $\mathcal{N}_{\text{loc}}$ , and we set  $x_n = 1, n \in \mathcal{N}_{\text{loc}}$ 
(18) else
(19)  Among the users in  $\mathcal{N}_1$ , perform Algorithm 2 for access control, where MK users are selected for MEC processing, and put the
      MK users into  $\mathcal{N}_{\text{MEC}}$ , and set  $y_n = 1, n \in \mathcal{N}_{\text{mec}}$ 
(20)  Remove the  $N_1 - \text{MK}$  users to  $\mathcal{N}_{\text{cloud}}$ , and set  $z_n = 1, n \in \mathcal{N}_{\text{cloud}}$ 
(21) end if
(22) Output:  $\Pi = \{x_n, y_n, z_n\}, n \in \mathcal{N}$ .

```

ALGORITHM 1: Heuristic offloading decision-making algorithm.

```

(1) Initialization:
(2) Set  $\mathcal{G}_1 = \emptyset, \mathcal{G}_2 = \emptyset, \dots, \mathcal{G}_M = \emptyset, G_1 = 0, G_2 = 0, \dots, G_M = 0, N_{\text{total}} = 0$ 
(3)  $G_{\text{temp}} = \{1, 2, \dots, M\}$ 
(4)  $\mathbf{V} = \{0\}_{N_1, M}$ 
(5) for  $n \in \mathcal{N}_1$  do
(6)   for  $m \in \mathcal{M}$  do
(7)      $\Gamma_{n,m} = g_{n,m} F_m$ 
(8)      $\Gamma = \{\Gamma_{n,m}\}$ 
(9)   end for
(10) end for
(11) while  $N_{\text{total}} \leq \text{MK}$  do
(12)    $(n^*, m^*) = \text{argmax} \Gamma$ 
(13)   if  $G_{m^*} < K$  then
(14)      $v_{n^*, m^*} = 1, \mathcal{G}_{m^*} = \mathcal{G}_{m^*} \cup n^*, G_{m^*} = G_{m^*} + 1$ 
(15)     Remove  $n^*$  from  $\mathcal{N}_1$ 
(16)     Set  $\Gamma_{n^*} = \{\Gamma_{n^*, m}\} = 0, m \in \mathcal{M}$ 
(17)      $N_{\text{total}} = N_{\text{total}} + 1$ 
(18)   else
(19)     Remove  $m^*$  from  $G_{\text{temp}}$ 
(20)     Set  $\Gamma_{m^*} = \{\Gamma_{n, m^*}\} = 0, n \in \mathcal{N}_1$ 
(21)   end if
(22) end while
(23) Output: user admission control and clustering  $\mathbf{V} = \{v_{n,m}\}, n \in \mathcal{N}_1, m \in \mathcal{M}$ ; the rest users in  $\mathcal{N}_1$  which will offload their tasks to the cloud center.

```

ALGORITHM 2: Admission control and user clustering.

TABLE 1: Simulation parameters.

Parameter	Value
Number of UEs (N)	50
Number of APs (M)	4
Max users number of each AP (K)	8
Bandwidth of each AP (B)	400 Hz
Bandwidth of cloud (B_1)	10 kHz
Channel gain	5 ~ 14 randomly
Max transmit power of user n ($p_n^{\max}$)	0.1 W
Input data size (D_n)	1 ~ 20 kbit
Local processing coefficient (α)	0.05
Processing density (λ_u)	100 ~ 500
Local processing capability of user n ($f_n^{\max}$)	500 CPU cycles/s
Waiting power consumption of user n (p_n^{idle})	10^{-6} W
Power of noise (σ^2)	$10^{-7.4}$
Maximum tolerable latency ($T_n^{\max}$)	0.3 ms
Price of radio resources (λ)	10^{-4} \$/bps
Price of computing resources (ω)	10^{-4} \$/CPU cycles/s
Parameters μ_e and μ_{pay}	10^3 \$/J, 10^{-4}
Unit price of MEC energy consumption (η)	0.01 \$/J

$$\begin{aligned}
(\mathcal{P}_3): \min_{\mathbf{p}^{\text{mec}}} \sum_{n \in \mathcal{N}} E_n^{\text{mec}} \\
\text{s.t. (C1): } T_n \leq T_n^{\max}, \quad \forall n \in \mathcal{N}, \\
\text{(C3): } 0 \leq p_n^{\text{mec}} \leq p_n^{\max}, \quad \forall n \in \mathcal{N}.
\end{aligned} \quad (14)$$

For cloud processing users, we have

$$\begin{aligned}
(\mathcal{P}_4): \min_{\mathbf{p}^{\text{cloud}}} \sum_{n \in \mathcal{N}} \text{Pay}_n \\
\text{s.t. (C1): } T_n \leq T_n^{\max}, \quad \forall n \in \mathcal{N}, \\
\text{(C4): } 0 \leq p_n^{\text{cloud}} \leq p_n^{\max}, \quad \forall n \in \mathcal{N}.
\end{aligned} \quad (15)$$

In order to enable tractable solutions, we use the following simple method to obtain a closed-form solution.

For cloud processing users, it can be known that each user is allocated with the same bandwidth, so the fees is only affected by r_n^{cloud} . However, under the given channel gain, the uplink transmit rate r_n^{cloud} is only affected by the transmit power p_n^{cloud} . The larger the p_n^{cloud} , the higher the r_n^{cloud} . Therefore, in order to minimize the total fees while guaranteeing the delay requirement, we take the transmit power that could satisfy $T_n^{\text{cloud}} = T_n^{\max}$.

For MEC processing users in each cluster, we take the following strategy: (i) For the end user, i.e., user K , we take the transmit power p_K^{mec} that could satisfy $T_K^{\text{mec}} = T_K^{\max}$. (ii) After p_K^{mec} is determined, the transmit rate of user $K-1$, i.e., $r_{K-1,m}$ in (3) is only affected by its transmit power p_{K-1}^{mec} , so we can also take the transmit power that could satisfy $T_{K-1}^{\text{mec}} = T_{K-1}^{\max}$. (iii) When p_K^{mec} and p_{K-1}^{mec} are determined, p_{K-2}^{mec} can be obtained using the same method, and then $p_{K-3}^{\text{mec}} \dots p_1^{\text{mec}}$ can be determined one after another following the same way.

7. Simulation Results

In this section, we evaluate the system performance of NOMA-based MEC-enabled blockchain network. The simulation parameters are listed in Table 1.

We compare our proposed algorithm with the following algorithm. (i) Random computing offloading algorithm: users randomly offloaded their tasks to the local or MEC serves or cloud center. In addition, it will use our proposed algorithm for local computing resource allocation, cloud center power control, and MEC power control. (ii) Random power algorithm: the power is generated randomly, and other resource allocation schemes use our proposed algorithm.

Figure 2 shows how the number of users' access affects the total cost of the system. As the number of access users in each APs increase, the energy consumption of AP processing tasks also increase, and so does the total cost of the system. The trends of the three curves in Figure 2 are consistent with our analysis. But our algorithm has the lowest cost. Since our algorithm has power optimization processing in both MEC and cloud center, the energy consumption of our algorithm is lower than that of the random power algorithm. Similarly, our algorithm also optimizes random offloading and users access, so with the increase of access point users, the cost of random offloading algorithm is higher than our algorithm.

In Figure 3, we plot the effect of the weight parameter μ_e on the system cost. As the weight parameter μ_e regulates the local energy consumption and MEC energy consumption, the local and MEC costs are on the same order of magnitude as the cloud center costs. When the weight parameter μ_e is 0, only the cost of users in the cloud center is included in the total cost. In the three curves of Figure 3, since the offloading decision and user access in the random offloading algorithm are random, the cost of users in the cloud center is the highest, that is, the total cost is also the highest. With the increase of weight parameter μ_e , the proposed algorithm has the lowest cost compared with other baseline algorithms through joint optimization of various factors.

Figure 4 plots how local processing capacity $f_n^{\max}$ affects the total cost of the system. With the increase of local processing capacity $f_n^{\max}$, the number of users processing

FIGURE 2: Total cost vs. number of users.

FIGURE 4: Total cost vs. local processing capability $f_n^{\max}$. FIGURE 3: Total cost vs. weight parameter μ_e . FIGURE 5: Total cost vs. unit price of MEC energy consumption η .

tasks locally increases, the local processing energy consumption increases, and the total system cost also increases. As can be seen from the figure, all three curves increase as local processing capacity increases. Since the local users of the random offloading algorithm are randomly generated, while our algorithm is the local users generated by the optimization algorithm, the total cost of the random offloading is higher than our proposed algorithm due to the two different offloading methods. In addition, another factor that determines cost is transmit power. Since the transmit power allocation of our algorithm is optimized, while the power allocation of the random power algorithm is random, the cost of our algorithm is lower than the random power algorithm.

In Figure 5, we show how the unit price of MEC energy consumption η affects the system cost. It can be seen that, as the unit price of MEC energy consumption η increases, the cost of all algorithms increases. This is in line with our

intuition. As the unit price increases, the additional net income from mining will decrease. According to formula (11), the total cost of the system will increase. In addition, since our proposed algorithm is optimized in all aspects, our total cost is the least compared to other algorithms.

8. Conclusions

In this paper, we have considered the joint optimization of offloading decision, local computation resource allocation, user clustering and admission control, and transmit power control, subjecting to the task execution deadline in a NOMA-based MEC-enabled blockchain network for minimizing the system cost. To solve the problem effectively, we propose heuristic algorithms with low complexity. In our problem solution process, the offloading decision is solved

based on heuristic Algorithm 1, and the user clustering and admission control are solved through Algorithm 2, and then closed-form solutions of local resource allocation and transmit power control are obtained under the required delay constraints. Simulation results show that the proposed algorithms have a good effect on cost reduction.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Shaanxi Province under grant no. 2020JQ-844, Natural Science Foundation of China under grant nos. 61901367, 62001357, 61871321, and U1965102, Science and Technology Innovation Team of Shaanxi Province for Broadband Wireless and Application under grant no. 2017KCT-30-02, the National Science and Technology Major Project under grant no. 2016ZX03001016, and Project of Key Laboratory of Science and Technology on Communication Network under grant no. 6142104200412.

References

- [1] T. M. Fernandez-Carames and P. Fraga-Lamas, "A review on the use of blockchain for the internet of things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.
- [2] J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Co-operative computation offloading and resource allocation for blockchain-enabled mobile edge computing: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, pp. 6214–6228, 2019.
- [3] J. Du, F. R. Yu, X. Chu et al., "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 1079–1092, 2018.
- [4] D. Zhai, R. Zhang, Y. Wang, H. Sun, L. Cai, and Z. Ding, "Joint user pairing, mode selection, and power control for d2d-capable cellular networks enhanced by nonorthogonal multiple access," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8919–8932, 2019.
- [5] D. Zhai, R. Zhang, L. Cai, and F. R. Yu, "Delay minimization for massive internet of things with non-orthogonal multiple access," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 553–566, 2019.
- [6] F. Fang, J. Cheng, and Z. Ding, "Joint energy efficient sub-channel and power optimization for a downlink NOMA heterogeneous network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1351–1364, 2019.
- [7] D. Zhai, R. Zhang, L. Cai, B. Li, and Y. Jiang, "Energy-efficient user scheduling and power allocation for NOMA-based wireless networks with massive IoT devices," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1857–1868, 2018.
- [8] H. Cao, Y. Zhu, G. Zheng, and L. Yang, "A novel optimal mapping algorithm with less computational complexity for virtual network embedding," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 356–371, 2018.
- [9] Y. Zuo, S. Zhang, Y. Han, and S. Jin, "Computation resource allocation in mobile blockchain-enabled edge computing networks," in *Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 617–622, Chongqing, China, August 2020.
- [10] H. Wu, K. Wolter, P. Jiao et al., "EEDTO: an energy-efficient dynamic task offloading algorithm for blockchain-enabled IoT-edge-cloud orchestrated computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2163–2176, 2021.
- [11] K. Zhang, J. Cao, S. Leng, C. Shao, and Y. Zhang, "Mining task offloading in mobile edge computing empowered blockchain," in *Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 234–239, Tianjin, China, August 2019.
- [12] G. Zheng, C. Xu, and L. Tang, "Joint user association and resource allocation for NOMA-based MEC: a matching-coalition approach," in *Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC)*, Seoul, Korea, May 2020.
- [13] B. Liu, C. Liu, and M. Peng, "Resource allocation for energy-efficient MEC in NOMA-enabled massive IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 1015–1027, 2020.
- [14] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannis, "Optimal resource allocation for delay minimization in NOMA-MEC networks," *IEEE Transactions on Communications*, vol. 68, pp. 7867–7881, 2020.
- [15] Z. Ding, J. Xu, O. A. Dobre, and H. V. Poor, "Joint power and time allocation for NOMA-MEC offloading," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 6207–6211, 2019.
- [16] H. Cao, S. Wu, Y. Hu et al., "An efficient energy cost and mapping revenue strategy for interdomain NFV-enabled networks," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5723–5736, 2020.
- [17] H. Cao, L. Yang, and H. Zhu, "Novel node-ranking approach and multiple topology attributes-based embedding algorithm for single-domain virtual network embedding," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 108–120, 2018.
- [18] X. Li, J. Li, Y. Liu, Z. Ding, and A. Nallanathan, "Residual transceiver hardware impairments on cooperative NOMA networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 680–695, 2020.
- [19] X. Li, M. Zhao, Y. Liu, L. Li, Z. Ding, and A. Nallanathan, "Secrecy analysis of ambient backscatter noma systems under I/Q imbalance," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12286–12290, 2020.
- [20] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11008–11021, 2018.
- [21] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, pp. 4321–4334, 2020.
- [22] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "Mec-assisted immersive VR video streaming over terahertz wireless networks: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9517–9529, 2020.