Cloud/Edge Computing Resource Allocation and Pricing for Mobile Blockchain: An Iterative Greedy and Search Approach

Yuqi Fan, Lunfei Wang, Weili Wu, *Senior Member, IEEE*, and Dingzhu Du

***Abstract*— Blockchain can provide a dependable environment for the Internet of Things (IoT), while the high computing power and energy required by blockchain hinder its applications in IoT. Offloading the computation at the resource-limited IoT devices to a cloud/edge computing service provider (CESP) is a feasible solution to the execution of computation-intensive blockchain tasks. The CESP provides computing resources to IoT users with a cloud and multiple edge servers that work collaboratively such that the users are able to perform mobile blockchain services. Resource allocation and pricing of computing resources at the cloud/edges have a significant impact on the revenues of CESP and users. Most of the existing works on the cooperative edge–cloud for computation offloading assumes that a user is mapped to a prespecified edge server or the cloud. However, the CESP may choose a server from either the edge servers or the cloud to run the offloaded tasks by jointly considering the cost and income of the service provisioning. In this article, we formulate a Stackelberg game with CESP as the leader and users as the followers for cloud/edge computing resource management. We prove the existence of Stackelberg equilibrium and analyze the equilibrium. We then model the resource allocation and pricing at the CESP as a mixed-integer programming problem (MIP) with the objective to optimize the CESP's revenue and propose an efficient iterative greedy-and-search-based resource allocation and pricing algorithm (IGS). The algorithm solves two subproblems comprising the CESP's revenue optimization problem: resource allocation under a given resource price and resource pricing based on a specified resource allocation scheme. The first subproblem evaluates where to execute the computing tasks via a greedy-and-search-based approach, whereas the second subproblem estimates the resource price through golden section search. We conduct experiments through simulations. Simulation results show that the proposed algorithm can effectively improve the revenue of both the CESP and the IoT terminals.**

***Index Terms*— Cloud computing, edge computing, mobile blockchain, resource allocation, resource pricing.**

I. Introduction

IN 2008, Nakamoto [1] proposed an electronic transaction system that does not rely on trust between system participants and for the first time explained the principles of bitcoin and blockchain technology. Blockchain has the characteristics of immutability, decentralization, distributed ledgers, consensus, and so on, which enables blockchain to overcome the problem of distrust among users in the decentralized network and ensure the reliability and security of transactions [2], [3]. Blockchain has a wide range of applications from cryptocurrencies, financial services, and the Internet of Things (IoT) to public social services [4]. As of December 2019, the cryptocurrency market capitalization statistics website coinmarketcap.com shows that there were 4914 cryptocurrencies in the world with a total market value of more than \$200 billion, of which bitcoin market value accounts for about 66.9% [5]. The annual revenue of blockchain enterprises is expected to increase to approximately \$20 billion in 2025 [6].

Consensus is the core of blockchain, which guarantees the consistency and correctness of each transaction on all nodes and enables the blockchain for efficient collaborative work on a large scale without relying on a centralized organization. Some consensus algorithms, such as proof of work (PoW), require a large amount of computation. Users (miners) win rewards through mining, where the users need to solve a computationally challenging problem. The first miner who successfully solves the computation problem and reaches an agreement with other miners is considered as the winner of the competition, and the winner will receive a reward for successful mining. Since the energy consumption and computing power required by the computation-intensive consensus are prohibitively high, blockchain cannot be directly applied to the resource-limited IoT devices, which hinders the practical use of blockchain in mobile environments.

Offloading the computing tasks of the IoT devices to a cloud/edge computing service provider (CESP) is a feasible solution to computation resource-demanding blockchain in mobile environments [7]. IoT devices can participate in the mining by applying solo mining or pooled mining protocols through mining task offloading [8], [9]. By providing more computing resources with the CESP, computation offloading can speed up the calculation process, improve the performance of applications, and enable blockchain deployment on devices with limited power supporting hashing and encryption algorithms [10].

The CESP provides computing resources to the users with cloud and multiple edge servers that work collaboratively such that the users are able to perform mobile blockchain services,

Manuscript received March 25, 2020; revised September 9, 2020 and November 1, 2020; accepted December 20, 2020. Date of publication January 21, 2021; date of current version April 1, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61701162 and in part by the National Science Foundation of USA under Grant 1747818 and Grant 1907472. *(Corresponding author: Yuqi Fan.)*

Yuqi Fan and Lunfei Wang are with the School of Computer and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: yuqi.fan@hfut.edu.cn; cxwlf@mail.hfut.edu.cn).

Weili Wu and Dingzhu Du are with the Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75082 USA (e-mail: weiliwu@utdallas.edu; dzdu@utdallas.edu).

Digital Object Identifier 10.1109/TCSS.2021.3049152

2329-924X © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

such as mining and encrypting. The resource allocation and pricing scheme of computing resources at the cloud/edges has a significant impact on the revenues of CESP and users.

Most of the existing works on the cooperative edge-cloud for computation offloading assume that a user is mapped to a prespecified edge server or the cloud. However, the CESP may choose a server from either the edge servers or the cloud to run the offloaded tasks by jointly considering the cost and income of the service provisioning. In this article, we study the cloud/edge computing resource allocation and pricing problem for mobile blockchain under the computing offloading framework in which a CESP provides both cloud and multiple edge servers to run users' offloaded tasks. The contributions of this article are as follows.

1) We construct a computation offloading framework that includes the cloud, multiple edge servers, and multiple users (miners), and the cloud/edge servers are provided by a CESP. We formulate a Stackelberg game with CESP as the leader and users as the followers for cloud/edge computing resource management. We prove the existence of the Stackelberg equilibrium and analyze the equilibrium.
2) We model the resource allocation and pricing at the CESP as a mixed-integer programming problem (MIP) with the objective to optimize the CESP's revenue and propose an efficient iterative greedy-and-search-based resource allocation and pricing algorithm (IGS). Algorithm IGS solves two subproblems comprising the CESP's revenue optimization problem: resource allocation under a given resource price and resource pricing with a specified resource allocation scheme.
3) We conduct experiments through simulations. Simulation results show that the proposed algorithm can effectively improve the revenues of both the CESP and the IoT users.

The rest of this article is organized as follows. Section II introduces the related work. The system model is formulated and analyzed in Section III. Section IV presents the proposed resource allocation and pricing algorithm. The simulations are given in Section V, and Section VI concludes this article.

II. Related Work

Blockchain has the characteristics of security, reliability, immutability, decentralization, and so on, and blockchain can provide a reliable environment for IoT devices. For example, Casado-Vara *et al.* [11] proposed an architecture based on blockchain and edge computing to improve the quality of IoT data and false data detection. Chamarajnagar and Ashok [12] designed a decentralized architecture using the blockchain technology, in order to promote distributed collaboration among mobile IoT devices to share their services and redundant computing resources. In vehicle edge networks, Kang *et al.* [13] proposed a reputation-based data sharing scheme, which introduces consortium blockchain and smart contract technology to implement data storage securely and prevent data sharing without authorization. Kim and Moon [14] proposed an edge computing architecture based on the blockchain technology to ensure the availability, scalability, and integrity of edge computing; blockchain structure and protocols were modified to support the execution of complex programs.

IoT devices can offload computation-intensive tasks to edge servers. For example, Xiong *et al.* [7] proposed a mobile blockchain framework that migrates computation-intensive tasks, e.g., PoW that requires a lot of computing resources, from IoT devices to the edge nodes with sufficient computing resources.

Resource allocation and pricing are of vital importance for the benefits of service providers and users. Some research applied game theory to the allocation and pricing of the cloud/edge computing resources. Xiong *et al.* [15] modeled the interaction between rational blockchain miners and cloud/fog providers as a two-stage Stackelberg game and studied the uniform pricing scheme and discriminatory pricing scheme of cloud/fog providers. Zhang *et al.* [16] proposed a joint optimization framework of fog nodes (FNs), data service operators (DSOs), and data service subscribers (DSSs), which implements the resource allocation scheme in a distributed manner. In this framework, the Stackelberg game is used to analyze the pricing problem of DSO and the resource allocation problem of DSS; many-to-many matching is used to study the matching problem between DSOs and FNs. Dhamal *et al.* [17] studied a stochastic game in which players (miners or computing power providers) can join and leave during the mining of a block. Chiu and Koeppl, [18] formalized the PoW protocol into a Cournot game, in which users compete to update the blockchain for rewards and are restricted from "double spending."

Some research adopted auction for resource allocation and pricing of cloud/edge resources. Jiao *et al.* [19] constructed an auction-based market model to achieve the allocation of computing resources; two bidding schemes, a fixed demand scheme (each miner bids for a fixed amount of resources) and a multiple demand scheme (miners can submit their preferred demand and bid), were considered; aiming at the fixed demand scheme, an optimal social welfare auction mechanism was proposed. For the multiple demand scheme, an approximate algorithm was designed with authenticity, individual rationality, and computational efficiency. Jin *et al.* [20] designed an incentive-compatible auction mechanism (ICAM) to stimulate cloudlets to provide services to nearby mobile devices, which reduces mobile device access delays and balances the workload of the cloud by using the resources of cloudlets. Luong *et al.* [21] proposed an optimal auction based on deep learning for the edge resource allocation, which uses miners' valuations as the training data to adjust the parameters of the neural networks to minimize the negated revenue of the edge computing service provider.

Some researchers managed the cloud/edge computing resources for blockchain via optimization, credit-based approach, and so on. Liu *et al.* [22] proposed a blockchain-based mobile edge computing (MEC) framework with the adaptive block size for video streaming; the tasks are offloaded to nearby MEC nodes or device-to-device (D2D) users. The problem of resource allocation, offload scheduling, and

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

adaptive block size was formulated as an optimization problem, and an alternating direction multiplier algorithm was used to solve the problem. Liu *et al.* [8] proposed a framework combining MEC and blockchain; a joint optimization problem of mining task offloading and block cryptographic hash cache was modeled, and an alternating direction multiplier method was adopted to solve the problem. Wu *et al.* [23] tackled the problem of mobile terminals acquiring computing power from the edge server and proposed an optimization problem to maximize the total net return of the mobile terminals while maintaining the fairness between the mobile terminals. Two algorithms were designed, with one for the single edge server case and the other for the multiple edge servers case, to determine the amount of computing power the mobile terminals could obtain from different edge servers. Abdellatif and Abdelmouttalib [6] studied the relationship between the resources provided by edge service providers and the needs of miners in the blockchain network and proposed a resource allocation model based on bipartite graph matching. Pan *et al.* [24] designed an edge IoT framework based on blockchain and smart contracts named "EdgeChain"; the framework connects edge cloud resource pool with the IoT device accounts and resources usage behavior through an internal currency system and then uses a credit-based resource management system to control the amount of resources a device can obtain from an edge server according to predefined priorities, application types, and previous behaviors.

Most of the existing works on the cooperative edge-cloud for computation offloading assume that a user is mapped to a prespecified edge server or the cloud. However, the CESP may choose a server from either the edge servers or the cloud to run the offloaded tasks by jointly considering the cost and income of the service provisioning. In this article, we study the cloud/edge computing resource allocation and pricing problem for mobile blockchain under the computing offloading framework consisting of a cloud, multiple edge servers, and multiple users, where the cloud/edge servers are provided by a CESP.

III. SYSTEM MODEL

Fig. 1 shows the system model. Each mobile terminal user performs mobile blockchain services, e.g., mining, for rewards. Due to the limited computing and energy resources, performing the blockchain services on the mobile terminals is challenging. A CESP provides computing resources to the users through the cloud and the edge servers such that the users can offload computation-intensive mining tasks to the cloud/edge servers. In general, the cloud is far away from the users, whereas the edge servers are close to the users. Assume that the cloud has enough computing capacities to process all the users' tasks and each edge server has a limited computing capacity.

The CESP gains revenue by providing paid resources to the users. Each unit computing resource provided by the CESP is denoted as a computing resource block (CRB). The CESP sets the price p of each CRB, and the users determine the amount of resources to purchase from the CESP based on the

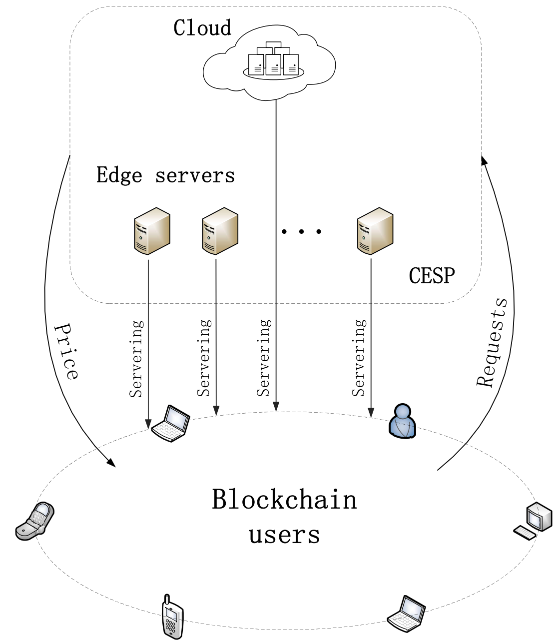


Fig. 1. System model.

CRB's price p. With more computing resources, the user is more likely to perform successful mining and hence potentially obtains more revenue [8], [25]. When the CRB's price is low, the users may be willing to purchase a large amount of computing resources due to the potential high revenue of successful mining. The users may be reluctant to buy the computing resources, if the CRB's price is high because of the high purchase cost of computing resources. Therefore, the CRB's price has a significant impact on the revenues of both CESP and users. Resource provisioning to the users also imposes a cost on the CESP, where the cost includes the energy consumption of the servers and the data transmission between the user and the allocated server.

After evaluating the number of CRBs to purchase, the user sends a resource purchase request to the CESP, who decides which edge server or the cloud to serve the request under the constraint of server's computing capacity. The computing tasks of a user will be offloaded to a server at either the cloud or the edge side. After purchasing the required resources, the user offloads the tasks to the allocated server.

During the interaction between the CESP and the users, the CESP determines the price based on the users requests and the users respond to the price by deciding the amount of computing resources to be purchased. The two events are sequential. Therefore, the interaction between the CESP and the users can be formalized as a Stackelberg game with a single leader and multiple followers, where the leader is the CESP and the followers are the users.

The symbols and notations used in this article are shown in Table I.

A. Computing Offloading Game Between CESP and Users

1) Stackelberg Game Between CESP and Users: The Stackelberg game between CESP and users consists of two subgames: 1) user subgame in which each user decides the amount

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

TABLE I
TABLE OF SYMBOLS AND NOTATIONS

Notation	Definition
b_i	User i
B	The set of users
M	The number of users
e_j	j-th edge server
E	The set of edge servers
K	The number of edge servers
r_i	The amount of resources required by user i
τ_i	The server (edge server or cloud) currently serving user i
c_j	Computing capacity of edge server e_j
C	The set of edge servers' computing capacities
p	The CRB's price set by the CESP
μ	The computing power of each CRB
α_i	The revenue parameter of user b_i
a_j	The unit energy consumption of computing resources at edge server e_j
a_{cloud}	The unit energy consumption of computing resources at the cloud
λ_1	The weighting coefficient of the data transmission between users and edge servers
λ_2	The weighting coefficient of the data transmission between users and cloud
$d_{i,j}$	The distance between user i and edge server e_j
$d_{i,cloud}$	The distance between user i and the cloud

of CRBs to purchase from the CESP to maximize the user's revenue and 2) CESP subgame in which the CESP decides the price of computing resources and chooses a server from either the edge servers or the cloud to run the users' offloaded tasks.

a) User subgame: If the user purchases a large amount of computing resources from the CESP, the blockchain task execution time is reduced, and hence, the user will potentially obtain a high reward by finishing the blockchain services in a short time. However, the revenue increase will slow down with the increase of computing power when the user has obtained a certain amount of computing resources. At the same time, the computing resource purchase causes a cost to the user. The more computing resources purchased, the higher the cost to the user. We estimate the user revenue by the following equation:

$$u_i^{\text{user}} = \alpha_i \ln(1 + \mu r_i) - p r_i. \tag{1}$$

The resource requirements of blockchain applications performed by various users are different. Therefore, we assign different revenue parameters to different users [24].

In user subgame, each user b_i needs to maximize his revenue by determining the amount of CRBs to purchase from the CESP, that is, user b_i is to

$$\max_{r_i} u_i^{\text{user}} = \alpha_i \ln(1 + \mu r_i) - p r_i$$
$$\text{s.t. } r_i \geq 0. \tag{2}$$

b) CESP subgame: The CESP makes profit by selling computing resources to the users, and the profit is affected by the CRB's price and the number of CRBs sold. Note that the resource allocation to the user will impose a cost on the CESP since the CESP needs to run the servers and transmit data between the user and the allocated server. The energy consumption used to run a task at a server is affected by the unit energy consumption of computing resources at the server and the amount of resources required by the task. The communication cost is decided by the user-to-server distance [26]. The computing tasks offloaded by user b_i may be deployed at the cloud or an edge server, and the cost of accommodating the computing task offloading request from user b_i, cost_i, is defined as

$$\text{cost}_i = x_{i,j}(a_j r_i + \lambda_1 d_{i,j}) + x_{i,\text{cloud}}(a_{\text{cloud}} r_i + \lambda_2 d_{i,\text{cloud}}) \tag{3}$$

where $x_{i,j}$ ($x_{i,j} \in \{0, 1\}$) and $x_{i,\text{cloud}}$ ($x_{i,\text{cloud}} \in \{0, 1\}$) represent whether the tasks of user b_i will be offloaded to edge server e_j and the cloud, respectively (=0, yes; =1, no). We say that user b_i is mapped/assigned to edge server e_j/cloud, if the tasks of user b_i are run by edge server e_j/cloud, and the computing tasks of user b_i are offloaded to one and only one server.

We define the CESP's revenue as

$$u^{\text{cesp}} = \sum_{i=1}^{M} (p r_i - \text{cost}_i). \tag{4}$$

The CESP maximizes the revenue by controlling the CRB's price, that is, the CESP is to

$$\max_p u^{\text{cesp}} = \sum_{i=1}^{M} (p r_i - \text{cost}_i)$$
$$\text{s.t. } \mathcal{C}1 : p > 0. \tag{5}$$

2) Analysis of Stackelberg Game Between CESP and Users: We first show that the Nash equilibrium exists in user subgame and the optimal amount of CRBs to be purchased by a user can be calculated at a given computing resource price. We then prove the existence of equilibrium of the Stackelberg game between the CESP and the users.

Lemma 1: Nash equilibrium exists in user subgame.

Proof: The user's revenue function defined in (1) is continuous, and the second derivative of the user's revenue function can be calculated as

$$\frac{\partial^2 u_i^{\text{user}}}{\partial (r_i)^2} = -\frac{\mu^2 \alpha_i}{(1 + \mu r_i)^2}. \tag{6}$$

Since $\mu > 0$, $\alpha_i > 0$, and $r_i \geq 0$, we can get

$$-\frac{\mu^2 \alpha_i}{(1 + \mu r_i)^2} < 0. \tag{7}$$

Therefore, the user's revenue function defined in (1) is quasi-concave about request r_i. According to the Nash existence theorem, the Nash equilibrium exists in the user subgame. □

Lemma 2: Given the CRB's price p, the optimal amount of CRBs to be purchased by user b_i is calculated by the following equation:

$$r_i^* = \max\left(\frac{\alpha_i}{p} - \frac{1}{\mu}, 0\right). \tag{8}$$

Proof: It can be seen from (6) and (7) that the second derivative of the user's revenue function with respect to r_i is less than 0, that is, the user's revenue function with respect to r_i is a concave function. The first derivative of the user's revenue function is calculated as

$$\frac{\partial u_i^{\text{user}}}{\partial r_i} = \frac{\mu \alpha_i}{1 + \mu r_i} - p. \tag{9}$$

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

The extreme point of the user's revenue function is calculated via the following equation by making (9) equal to 0, i.e., $\partial u_i^{\text{user}}/\partial r_i = 0$:

$$r_i' = \frac{\alpha_i}{p} - \frac{1}{\mu}. \tag{10}$$

According to the property of the concave function, the extreme point of the function is the maximum point of the function. In (10), r_i' may be less than 0. However, r_i^* should satisfy $r_i^* \geq 0$. When $r_i' \geq 0$, $r_i^* = r_i'$; when $r_i' < 0$, $r_i^* = 0$, since the user's revenue function decreases while $r_i > r_i'$ and $r_i \geq 0$. Therefore, $r_i^* = \max(r_i', 0)$. The lemma is proved. □

Theorem 1: The Stackelberg equilibrium exists in the Stackelberg game between the CESP and the users.

Proof: According to Lemma 1, the users can always reach a Nash equilibrium after the CRB's price p is set by the CESP. Next, we analyze the CESP's revenue function of (4), which can be rewritten as (11), shown at the bottom of the page, based on Lemma 2.

The second derivative of CESP's revenue function can be calculated as (12), shown at the bottom of the page.

Since $x_{i,j} \geq 0$, $x_{i,\text{cloud}} \geq 0$, $a_j > 0$, $a_{\text{cloud}} > 0$, $\mu > 0$, $d_{i,j} > 0$, $d_{i,\text{cloud}} > 0$, $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, $\alpha_i > 0$, and $p > 0$, we can get

$$\frac{\partial^2 u^{\text{cesp}}}{\partial p^2} < 0. \tag{13}$$

Therefore, there is an optimal price $p*$ that maximizes the CESP's revenue, and the Stackelberg equilibrium exists. We denote the Stackelberg equilibrium as $(p^*, r_1^*, r_2^*, \ldots, r_m^*)$. □

B. Resource Allocation and Pricing of CESP

The analysis of Stackelberg game between CESP and users shows that the users can always reach a Nash equilibrium after the CRB's price is set by the CESP and the optimal amount of CRBs to be purchased by a user can be calculated at a given computing resource price. Therefore, it is important to decide the resource allocation and pricing of the CESP.

The CESP maximizes the revenue by setting CRB's price p and deciding where to run the tasks offloaded by the users. The CRB's price affects the amount of computing resources to be purchased by the users, which in turn has an impact on the CESP's revenue. Given CRB's price p ($p > 0$) and the amount of computing resources purchased by the users, the CESP allocates each resource request to a server either at the edge server side or at the cloud side, so as to minimize the resource allocation cost and maximize the CESP's revenue, that is, the objective of CESP is to

$$\max_{p, x_{i,j}, x_{i,\text{cloud}}} \; u^{\text{cesp}} \tag{14}$$

where u^{cesp} is defined in the following equation:

$$\begin{aligned}
\text{s.t. } & \mathcal{C}1\\
& \mathcal{C}2: \sum_{i=1}^{M} x_{i,j} \max\left(\frac{\alpha_i}{p} - \frac{1}{\mu}, 0\right) \leq c_j \quad \forall e_j \in E\\
& \mathcal{C}3: \sum_{j=1}^{K} (x_{i,j} + x_{i,\text{cloud}}) = 1 \quad \forall b_i \in B\\
& \mathcal{C}4: x_{i,j} \in \{0, 1\}, x_{i,\text{cloud}} \in \{0, 1\}.
\end{aligned}$$

Constraint $\mathcal{C}2$ signifies that the edge servers cannot be overloaded when running the tasks offloaded from the users. Constraint $\mathcal{C}3$ determines that the computing tasks of user b_i are offloaded to one and only one server. Constraint $\mathcal{C}4$ dictates that the mapping relationship between the users and the cloud/edge servers is represented by binary variables.

IV. Resource Allocation and Pricing Algorithm

The Stackelberg game between CESP and users consists of two subgames: user subgame and CESP subgame, as shown in Fig 2. In the user sub-game, the users maximize revenue by calculating the optimal amount of CRBs to purchase according to Lemma 2. In the CESP sub-game, the CESP maximizes revenue by deciding the price of computing resources and choosing the servers to run the users offloaded tasks. The problem of maximizing the CESP's revenue is an MIP since $p > 0$, $x_{i,j} \in \{0, 1\}$, and $x_{i,\text{cloud}} \in \{0, 1\}$.

In this section, we propose an efficient iterative greedy-and-search-based resource allocation and pricing algorithm (IGS) to solve the CESP's revenue optimization problem that consists of two subproblems: resource allocation under a given resource

$$\begin{aligned}
u^{\text{cesp}} &= \sum_{i=1}^{M} \left\{ \sum_{j=1}^{K} x_{i,j}\big(pr_i - a_j r_i - \lambda_1 d_{i,j}\big) + x_{i,\text{cloud}}\big(pr_i - a_{\text{cloud}} r_i - \lambda_2 d_{i,\text{cloud}}\big) \right\}\\
&= \sum_{i=1}^{M} \left\{ \sum_{j=1}^{K} x_{i,j}\left(\max\left(\frac{\alpha_i}{p} - \frac{1}{\mu}, 0\right)\big(p - a_j\big) - \lambda_1 d_{i,j}\right) + x_{i,\text{cloud}}\left(\max\left(\frac{\alpha_i}{p} - \frac{1}{\mu}, 0\right)(p - a_{\text{cloud}}) - \lambda_2 d_{i,\text{cloud}}\right) \right\}
\end{aligned} \tag{11}$$

$$\frac{\partial^2 u^{\text{cesp}}}{\partial p^2} = \sum_{i=1}^{M} \begin{cases} 0, & \dfrac{\alpha_i}{p} - \dfrac{1}{\mu} \leqslant 0 \\ -\dfrac{\alpha_i}{p^3}\left(\displaystyle\sum_{j=1}^{K} \lambda_1 x_{i,j} a_j + \lambda_2 x_{i,\text{cloud}} a_{\text{cloud}}\right), & \dfrac{\alpha_i}{p} - \dfrac{1}{\mu} > 0 \end{cases} \tag{12}$$

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

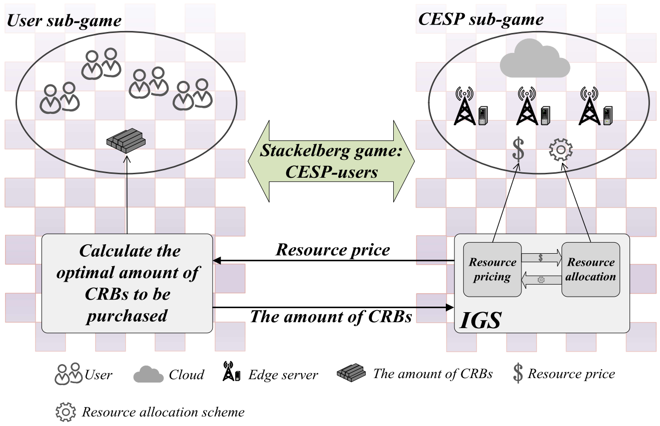


Fig. 2. Diagram of Stackelberg game between CESP and users.

price and resource pricing with a specified resource allocation scheme. Algorithm IGS initializes the resource price with a high value. Algorithm IGS then solves the first subproblem of resource allocation by evaluating where to execute the computing tasks based on a specific resource price via a greedy-and-search based approach while tackling the second subproblem of resource pricing by estimating the resource price based on a specified resource allocation scheme via golden section search.

A. Resource Allocation

Resource allocation chooses a server from either the edge servers or the cloud to run the computing tasks based on a resource price. Given CRB's price p ($p > 0$), the CESP's revenue maximization problem turns into an integer programming problem. We propose a resource allocation algorithm based on a greedy-and-search approach, as shown in Algorithm 1. The algorithm is divided into two stages. The first stage (lines 1–12) greedily finds the server for each task offloading request, and the second stage (lines 13–32) conducts resource reallocation to search for better resource allocation solutions.

1) Greedy Resource Allocation: We create a computing resource request list $l_{req} = \{r_1^*, r_2^*, \ldots, r_i^*, \ldots, r_M^*\}$, where each r_i^* ($1 \le i \le M$), calculated via (8), is the optimal amount of CRBs purchased by user b_i under a given CRB's price. We then construct a value matrix G_v with M rows (M users) and $N = K+1$ columns (K edge servers and a cloud) as shown in (15). Each element $v_{ij} \in G_v$, calculated by (16), represents the revenue of serving user b_i at the cloud if $j = N$; otherwise, v_{ij} indicates the revenue of edge server e_j running the tasks of user b_i. Algorithm 1 is to choose an element from each row of matrix G_v to maximize the sum of the elements under constraint $\mathcal{C}3$

$$G_v = \begin{pmatrix} v_{11} & \cdots & v_{1K} & v_{1N} \\ \vdots & \ddots & \vdots & \vdots \\ v_{M1} & \cdots & v_{MK} & v_{MN} \end{pmatrix} \quad (15)$$

$$v_{ij} = pr_i^* - \text{cost}_i. \quad (16)$$

We construct another auxiliary matrix G_w based on matrix G_v, as shown in (17), where element $w_{ij} = v_{ij}/r_i^*$, indicating the cost-effectiveness of serving user b_i at edge server e_j/cloud

$$G_w = \begin{pmatrix} w_{11} & \cdots & v_{1K} & w_{1N} \\ \vdots & \ddots & \vdots & \vdots \\ w_{M1} & \cdots & v_{MK} & w_{MN} \end{pmatrix}. \quad (17)$$

We define the unserved user set $B' = \{b_1', b_2', \ldots, b_i', \cdots\}$ ($|B'| \le K$), in which each element is the user to be allocated the computing resources. We also define server's computing capacity set $C' = \{c_1', c_2', \ldots, c_j', \ldots, c_K', c_{\text{cloud}}'\}$ ($|C'| = N$), with each element c_j' indicating the available computing capacity of edge server e_j ($1 \le j \le K$) or cloud ($j = N$). Since the computing resources of the cloud are sufficient, we set c_{cloud}' as a large value.

Algorithm 1 Greedy-and-Search Based Resource Allocation

Input: User set B, Edge server set E,
Set $C = \{c_1, c_2, \cdots, c_j, \cdots, c_K\}$, CRB's price p.
Output: Resource allocation scheme $\boldsymbol{x}$,
CESP's revenue of u^{cesp}.

1: Create computing resource request list l_{req}, where each request in the list is calculated via Eq. (8);
2: Construct two auxiliary matrices of G_v and G_w as Eqs. (15) and (17);
3: $B' \leftarrow B$, $C' \leftarrow C + \{c_{cloud}'\}$, $u^{cesp} \leftarrow 0$;
4: **while** $B' \ne \emptyset$ **do**
5: Select user b_i' and server e_j with w_{ij} being the largest element w_{ij} in matrix G_w;
6: **if** $c_j \ge r_i^*$ **then**
7: $x_{ij} \leftarrow 1$, $c_j \leftarrow c_j - r_i^*$, $B' \leftarrow B' - \{b_i'\}$;
8: $u^{cesp} \leftarrow u^{cesp} + v_{ij}$, and $w_{ij'} = \mathcal{T}$ ($\forall 1 \le j' \le N$);
9: **else**
10: set $w_{ij} \leftarrow \mathcal{T}$;
11: **end if**
12: **end while**
13: $B \leftarrow B$;
14: **while** $B' \ne \emptyset$ **do**
15: Select the first user b_i' from B', $flag \leftarrow 0$;
16: **for** each $b_{i'}' \in B$ **do**
17: **if** $\pi_1(i, \tau_i, i', \tau_{i'}) > 0$ and $c_{\tau_i}' + r_i^* \ge r_{i'}^*$ and $c_{\tau_{i'}}' + r_{i'}^* \ge r_i^*$ **then**
18: $x_{i\tau_i} \leftarrow x_{i\tau_{i'}} \leftarrow 1$, $x_{i'\tau_{i'}} \leftarrow x_{i'\tau_i} \leftarrow 1$,
19: $c_{\tau_i}' \leftarrow c_{\tau_i}' + r_i^* - r_{i'}^*$, $c_{\tau_{i'}}' \leftarrow c_{\tau_{i'}}' + r_{i'}^* - r_i^*$;
20: $u^{cesp} \leftarrow u^{cesp} + \pi_1(i, \tau_i, j, \tau_{i'})$, $flag \leftarrow 1$;
21: **end if**
22: **end for**
23: **for** each $e_q \in E$ **do**
24: **if** $\pi_2(i, \tau_i, q) > 0$ and $c_q' \ge r_i^*$ **then**
25: $x_{i\tau_i} \leftarrow x_{iq} \leftarrow 1$, $c_{\tau_i}' \leftarrow c_{\tau_i}' + r_i^*$;
26: $c_q' \leftarrow c_q' - r_i^*$, $u^{cesp} \leftarrow u^{cesp} + \pi_2(i, \tau_i, q)$, $flag \leftarrow 1$;
27: **end if**
28: **end for**
29: **if** $flag = 0$ **then**
30: $B' \leftarrow B' - \{b_i'\}$;
31: **end if**
32: **end while**
33: **return** $\boldsymbol{x}, u^{cesp}$.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 proceeds iteratively. In each iteration, the algorithm selects the largest element w_{ij} from matrix G_w with greedy strategy, that is, the algorithm selects the most cost-effective user-server mapping. If the available computing capacity of server e_j can satisfy the resource request from user b_i, the tasks of b_i are executed by server e_j and all elements in the ith row in matrix G_w are set as a small value $\mathcal{T}$, where $\mathcal{T} < w_{min}$ and w_{min} is the smallest element in matrix G_w; otherwise, element w_{ij} is set as $\mathcal{T}$. The process continues until all the users are mapped to the servers.

2) Resource Reallocation: Once each user is mapped to a server, the CESP's revenue may be increased by exchanging the mapping relationship of two user-server assignments. Assume that users b_i and $b_{i'}$ $(i \neq i')$ are currently mapped to servers e_{τ_i} and $e_{\tau_{i'}}$ $(\tau_i \neq \tau_{i'})$, respectively. The exchange reassigns b_i and $b_{i'}$ to servers $e_{\tau_{i'}}$ and e_{τ_i}, respectively. We define $\pi_1(i, \tau_i, i', \tau_{i'})$, the benefit of exchange, as (18), where $\pi_1(i, \tau_i, i', \tau_{i'}) > 0$ indicates that the CESP's revenue can be increased through the exchange

$$\pi_1(i, \tau_i, i', \tau_{i'}) = (v_{i\tau_{i'}} + v_{i'\tau_i}) - (v_{i\tau_i} + v_{i'\tau_{i'}}). \tag{18}$$

Assigning the user to another server may also increase the CESP's revenue. We evaluate $\pi_2(i, \tau_i, q)$, the benefit of reassignment, via (19). We can improve the CESP's revenue by performing the reassignment, if $\pi_2(i, \tau_i, q) > 0$

$$\pi_2(i, \tau_i, q) = v_{iq} - v_{i\tau_i}. \tag{19}$$

Algorithm 1 conducts resource reallocation by exchange and reassignment. The algorithm proceeds iteratively. Specifically, for each user, the algorithm finds another user such that the CESP's revenue can be increased by exchanging the mapping relationship of the two users (lines 16–22). For each user, the algorithm also searches for another server to evaluate whether reassigning the user to the server can increase the CESP's revenue (lines 23–28). The process continues until exchange and reassignment are executed for all the users.

B. Resource Pricing

Resource pricing decides the resource price based on a specified resource allocation scheme. After obtaining the computing resource allocation scheme $\boldsymbol{x}$ by Algorithm 1 under a given CRB's price, we search for another CRB's price p to optimize the CESP's revenue with resource allocation scheme $\boldsymbol{x}$. Since $p > 0$, we make $p \geq \varepsilon$ during the resource pricing, where ε is a small value close to 0.

The CESP's revenue is related to the CRB's price and the amount of computing resources sold. A high price will make the users reluctant to purchase the computing resources, and no user will purchase any computing resources if CRB's price p is higher than a value $p_{\max}$, that is, the total amount of computing resources purchased by all users will be 0 if $p \geq p_{\max}$.

Theorem 2: When CRB's price $p \geq \max\{\mu\alpha_i | \forall b_i \in B\}$, the total amount of computing resources purchased by all the users is 0, that is, $p_{\max} = \max\{\mu\alpha_i | \forall b_i \in B\}$.

Proof: According to Lemma 2, when $\alpha_i/p - 1/\mu \leqslant 0$, i.e., $p \geq \mu\alpha_i$, the amount of computing resources purchased by user b_i is 0. If $\max(\alpha_i/p - 1/\mu) \leqslant 0$ for each user $b_i \in B$, i.e., $p \geq \max\{\mu\alpha_i | \forall b_i \in B\}$), no user will purchase the computing resources. Therefore, $p_{\max} = \max\{\mu\alpha_i | \forall b_i \in B\}$. □

Since $x_{i,j}$ $(\forall b_i \in B, 1 \leq j \leq K)$ and $x_{i,\text{cloud}}$ $(\forall b_i \in B)$ are known in a given resource allocation scheme $\boldsymbol{x}$, the optimization problem in Section III-B can be simplified with (20) as the objective function under constraint $\mathcal{C}2'$

$$\begin{aligned} &\max_p u^{\text{cesp}} = \sum_{i=1}^{M} \left(\frac{\alpha_i}{p} - \frac{1}{\mu} \right) (p - a_{\tau_i} - \lambda' \mu d_{i,\tau_i}) \\ &\text{s.t. } \mathcal{C}2' : \varepsilon \leq p \leq p_{\max} \end{aligned} \tag{20}$$

where τ_i denotes the server providing computing services to user b_i. $\lambda' = \lambda_1$ if τ_i is an edge server; $\lambda' = \lambda_2$ if τ_i is the cloud. The revenue function in (20) is a continuous concave function about price p under a given resource allocation scheme $\boldsymbol{x}$. We apply golden section search [27] to find the optimal price p. The resource pricing process is illustrated in Algorithm 2. The algorithm searches for the optimal price by iteratively narrowing the search range of values. Initially, the search range [left, right] is set as $[\varepsilon, p_{\max}]$. In each iteration, the algorithm updates the search range with the golden ratio. Specifically, the algorithm calculates two points in the range [left, right] as left $+ 0.382 *$ (right $-$ left) and left $+ 0.618 *$ (right $-$ left). The algorithm decides whether the optimal price falls into [left, left $+ 0.618 *$ (right $-$ left)] or [left $+ 0.382 *$ (right $-$ left), right] by comparing the revenues at the two prices of left and right via Algorithm 1. The search range narrowing process continues until the interval of the search range is smaller than a predefined threshold δ. The algorithm outputs the middle value of the final search range as the price.

Algorithm 2 Golden Section Search-Based Resource Pricing

Input: Resource allocation scheme $\boldsymbol{x}$.
Output: CRB's price p.

$left \leftarrow \varepsilon, right \leftarrow p_{max}, left' \leftarrow \varepsilon, right' \leftarrow p_{max}$;
while $right - left \geqslant \delta$ **do**
  $p \leftarrow left'$, $(val_1, \boldsymbol{x}) \leftarrow Algorithm\ 1(p)$;
  $p \leftarrow right'$, $(val_2, \boldsymbol{x}) \leftarrow Algorithm\ 1(p)$;
  if $val_1 > val_2$ **then**
    $right \leftarrow right'$;
  else
    $left \leftarrow left'$;
  end if
  $left' \leftarrow left + 0.382 * (right - left)$;
  $right' \leftarrow left + 0.618 * (right - left)$;
end while
$p \leftarrow (right + left)/2$;
return p.

C. Iterative Greedy-and-Search-Based Resource Allocation and Pricing

The proposed iterative greedy-and-search-based resource allocation and pricing algorithm is shown in Algorithm 3 by iteratively solving the subproblems of resource allocation and resource pricing. The algorithm initializes the resource price as

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3 IGS

Input: The user set B, Edge server set E,
Maximum number of iterations Γ,
Edge servers' computing capacity set C.
Output: New feasible solution $\boldsymbol{x}$,
Price p,
The revenue of CESP u^{cesp}.

$u^{cesp} \leftarrow 0$, $p \leftarrow p_{max}$, $val_{max} \leftarrow 0, iteration \leftarrow 1$;
while $iteration \leqslant \Gamma$ **do**
  $(val, \boldsymbol{x}') \leftarrow Algorithm\ 1(p)$;
  if $val > u^{cesp}$ **then**
    $u^{cesp} \leftarrow val, \boldsymbol{x} \leftarrow \boldsymbol{x}'$;
  end if
  $p' \leftarrow Algorithm\ 2(\boldsymbol{x}')$;
  if $|p - p'| \leqslant \Delta$ **then**
    break;
  end if
  $p \leftarrow p'$;
end while
return $\boldsymbol{x}$, p, $value$.

$p_{\max}$ and the CESP's revenue as 0. After the initialization, the algorithm starts an iterative resource allocation and resource pricing process. During each iteration, the algorithm decides the resource allocation scheme $\boldsymbol{x}'$ and the corresponding CESP's revenue val through Algorithm 1. If the CESP's revenue is increased with resource allocation scheme $\boldsymbol{x}'$, the algorithm updates $\boldsymbol{x}'$ and val as the current best resource allocation scheme and the current highest CESP's revenue, respectively. The algorithm then evaluates the optimal resource price with the obtained allocation scheme $\boldsymbol{x}'$ via Algorithm 2. The iterative process continues until the change of resource price is within a predetermined threshold Δ or the predefined maximum number of iterations is reached. After the Stackelberg game between users and CESP reaches the equilibrium, the users obtain the optimal amount of CRBs to purchase via (8) according to Lemma 2.

V. Performance Evaluation

In this section, we investigate the impact of important parameters on the proposed algorithm IGS. We also evaluate the performance of algorithm IGS against the benchmarks.

A. Simulation Setup

The mobile terminal users and the edges servers are randomly distributed in the region with 1.2 km radius. The revenue parameter α_i of all the mobile terminal users is the same. The minimum resource price ε is 0.001. Parameters of δ and Δ are set as 0.001 and 0.01, respectively. Other simulation parameters are shown in Table. II.

B. Impact of Different Parameters on the Proposed Algorithm

We investigate the impact of important parameters on the proposed algorithm.

TABLE II
Table of Simulation Parameters

Simulation Parameters	Value
Revenue parameter α_i of the user b_i	[5-60]
Computing power parameter μ	[0-6]
Computing capacity each edge server	[20-100]
The unit energy consumption of computing resources at each edge server	[1-10]
The unit energy consumption of computing resources at the cloud	1
λ_1, weighting coefficient of the data transmission between users and edge servers	0.1
λ_2, weighting coefficient of the data transmission between users and cloud	0.05

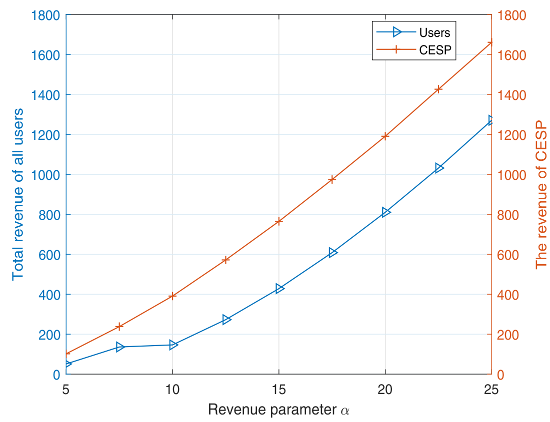


Fig. 3. Revenues of CESP and users versus different parameters α.

1) Impact of Different Parameters on the Revenue of CESP and Users: Fig. 3 shows that the revenues of both the CESP and the users increase with the increase of revenue parameter α when the computing power of each CRB is 1 ($\mu = 1$) and the number of users is 160 ($M = 160$). The unit computing resource leads to more revenue with a higher revenue parameter α than that with a lower one. Therefore, with the increase in revenue parameter α, the users are willing to purchase more computing resources to perform mobile blockchain services to make more revenues. At the same time, the CESP increases the revenue by providing more computer resources and appropriately increasing computing resource price p.

Fig. 4 shows that the revenues of both the CESP and the users increase as parameter μ increases when $\alpha = 15$ and $M = 160$. μ represents the computing power of each CRB. As parameter μ increases, the users can obtain more computing power with the same amount of CRBs purchased such that the users get more returns. The CESP increases the revenue performance by adapting CRB's price p to the amount of CRBs required by the users.

2) Impact of Different Parameters on the Total Number of CRBs Purchased by the Users: Fig. 5 shows the impact of revenue parameter α on the total number of CRBs purchased by all the users with a different number of users, assuming $\mu = 1$. The total number of CRBs purchased by the users increases with the increase of α. The higher parameter α, the more revenues the users can obtain by executing each unit computing resource for mobile blockchain services. Therefore, the users are inclined to purchase more CRBs with a higher

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

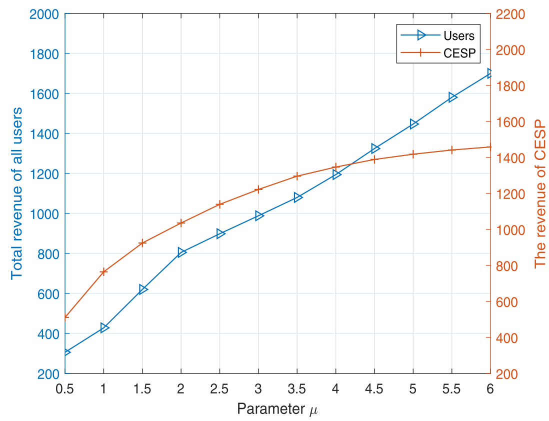


Fig. 4. Revenues of CESP and users with different parameters μ.

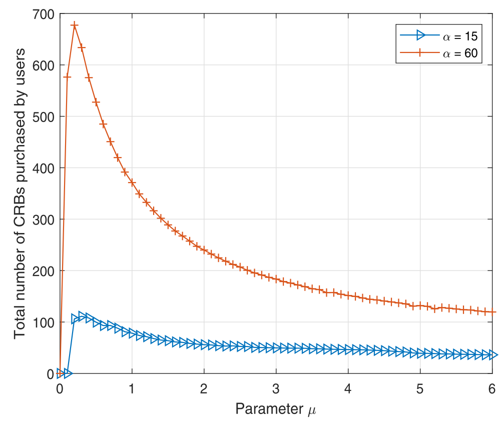


Fig. 6. Total number of CRBs sold with different parameters μ.

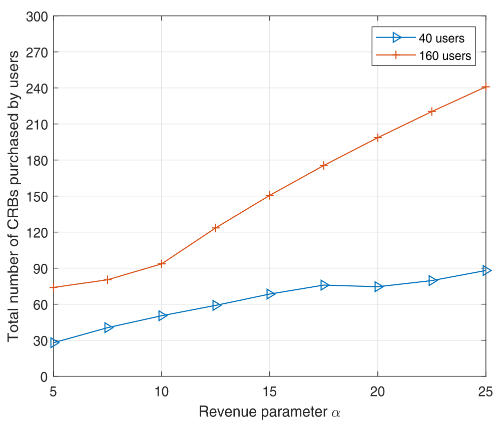


Fig. 5. Total number of CRBs sold versus different parameters α.

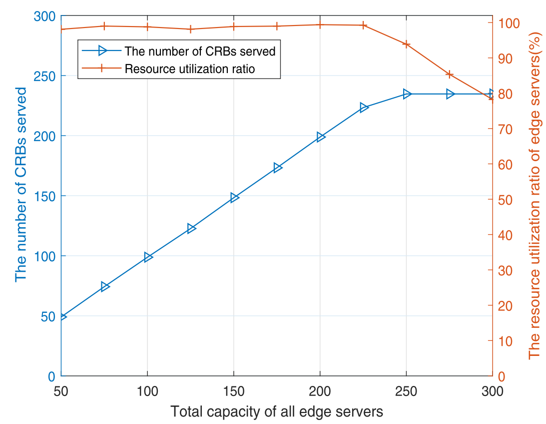


Fig. 7. Resource utilization of edge servers versus different total numbers of CRBs available at the edge servers.

α than with a lower α. The amount of computing resources increases with more users since more users will buy more computing resources to run the blockchain tasks.

Fig. 6 shows the impact of parameter μ on the total number of CRBs purchased by all the users with different α values, assuming that the number of users is 160. The total number of CRBs purchased by the users increases with the increasing μ when μ is small and then decreases with the increasing μ when μ is big. A higher μ indicates that each CRB is more powerful and can reward the users more than a lower μ. Therefore, users are willing to buy a large number of CRBs when μ is low. After obtaining sufficient computing power to serve the users' offloading requests, the users will not buy more CRBs. Therefore, with higher μ, the users only need to buy a smaller number of CRBs.

3) Impact of Different Parameters on the Resource Utilization of Edge Servers: Fig. 7 shows the edge server resource utilization performance versus different total number of CRBs available at the edge servers, assuming $M = 160$, $\alpha = 15$, and $\mu = 1$. The edge servers' resource utilization ratio is always above 95% when the total capacity of all the edge servers is within 225 CRBs. In this case, the transmission cost between the users and the edge servers is low since the edge servers are close to the users. The CESP will potentially prefer to provide the computing resources of the edge servers. All the computing resource demands can be satisfied by the edge servers when the total capacity of all the edge servers is larger than 225 CRBs, that is, the total number of CRBs served by the edge servers keeps stable. Therefore, the edge servers' resource utilization ratio decreases as the total number of CRBs available at the edge servers increases when the total capacity of all the edge servers is larger than 225.

Fig. 8 shows the performance of the total number of CRBs served by the edge servers by varying the unit energy consumption of computing resources at the edge servers, assuming that $M = 160$, $\alpha = 15$, $\mu = 1$, and the total number of CRBs available at the edge servers is 75. When the unit energy consumption of computing resources at the edge servers is lower than 4.5, the edge servers have high resource utilization. A high unit energy consumption of computing resources at the edge servers makes the cost of serving users by the edge servers exceed the cost of using the cloud to serve the users. Fig. 8 shows the number of CRBs served by the edge servers decreases rapidly to 0 when the unit energy consumption of computing resources at the edge servers is higher than 4.5.

4) Impact of Different Parameters on the Income and Cost of CESP: Fig. 9 shows the income and cost of CESP versus different parameters α when $\mu = 1$. With the increase of parameter α, the CESP's cost increases. With a high α, the users will buy a large amount of computing resources such that the CESP needs to spend more cost on the resource provisioning. At the same time, the CESP obtains more income with the increase in the number of CRBs sold.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

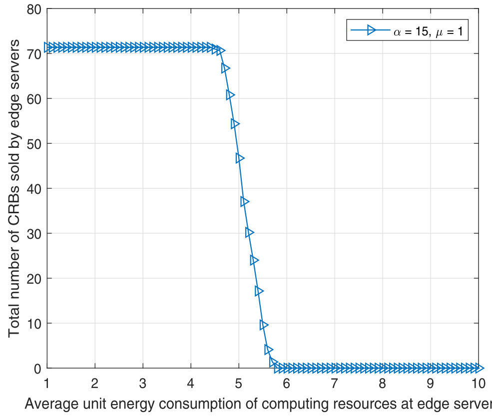


Fig. 8. Total number of CRBs served by the edge servers with various unit energy consumption of computing resources at the edge servers.

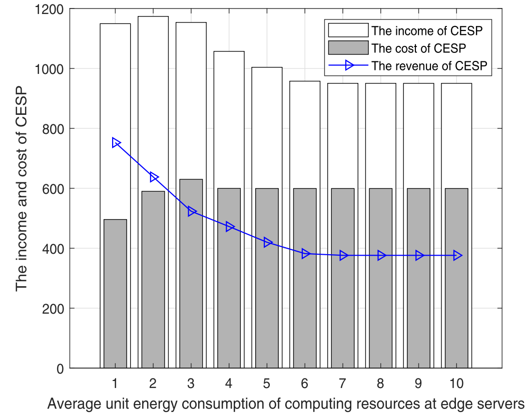


Fig. 10. Income and cost of CESP with different unit energy consumption of computing resources at the edge servers.

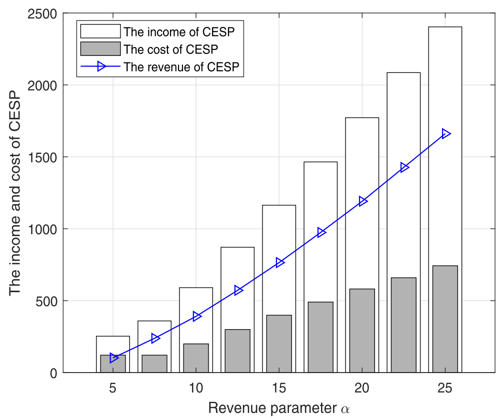


Fig. 9. Income and cost of CESP versus different parameters α.

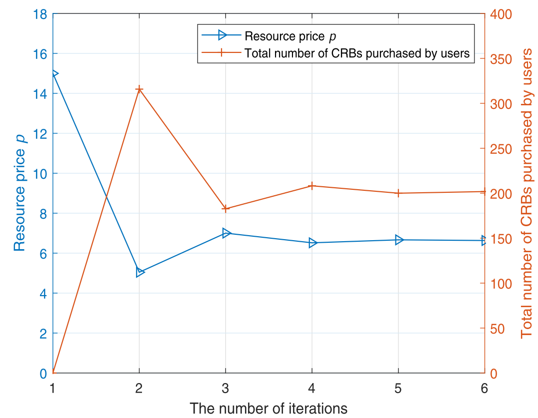


Fig. 11. Iterative procedure of algorithm IGS.

Fig. 10 shows that the income and cost of CESP with various unit energy consumption of computing resources at the edge servers. As the unit energy consumption of computing resources at the edge servers increases, the cost of providing services grows. When the unit energy consumption of computing resources at the edge servers is large, some of the users will be served by the cloud. In this case, the CESP can lower the resource price p to increase the amount of CRBs purchased by the other users who will be served by the edge servers to increase the utilization of the edge servers. Therefore, the CESP's income increases although the service provisioning cost also increases. The income and the cost will tend to stabilize as the unit energy consumption of computing resources at the edge servers increases to a certain value since it is so large that all the users will be served by the cloud.

C. Iterative Procedure of Algorithm IGS

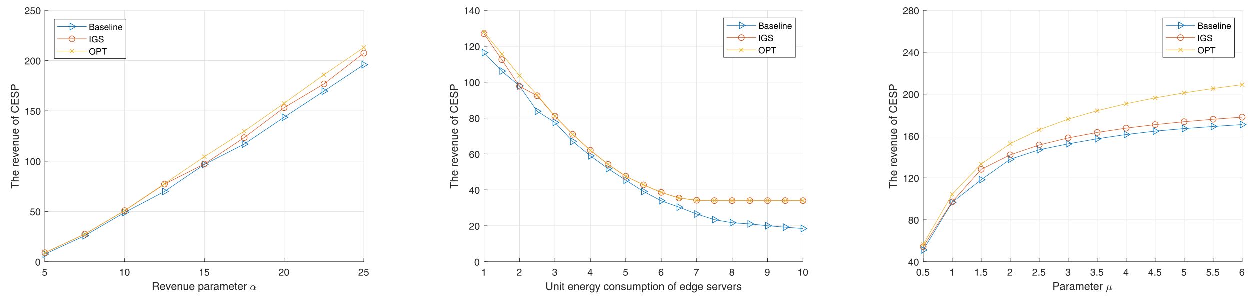
We discuss the iterative procedure of algorithm IGS and how the algorithm reaches the Nash equilibrium.

Fig. 11 shows the iterative process of algorithm IGS, assuming that $M = 160$, $\alpha = 15$, and $\mu = 1$. The algorithm first starts from a high resource price p, which makes all users unwilling to purchase the computing resources. The algorithm then dynamically adjusts the computing resource price p. Users dynamically adjust the amount of computing resources to purchase according to the resource price p. When the number of iterations is above 6, the resource price p keeps stable at about 6.6, and the total number of CRBs purchased by users stabilizes at about 200, that is, the Stackelberg game between users and CESP reaches the equilibrium, and the game between users reaches the Nash equilibrium.

D. Performance of the Proposed Algorithm Against Benchmarks

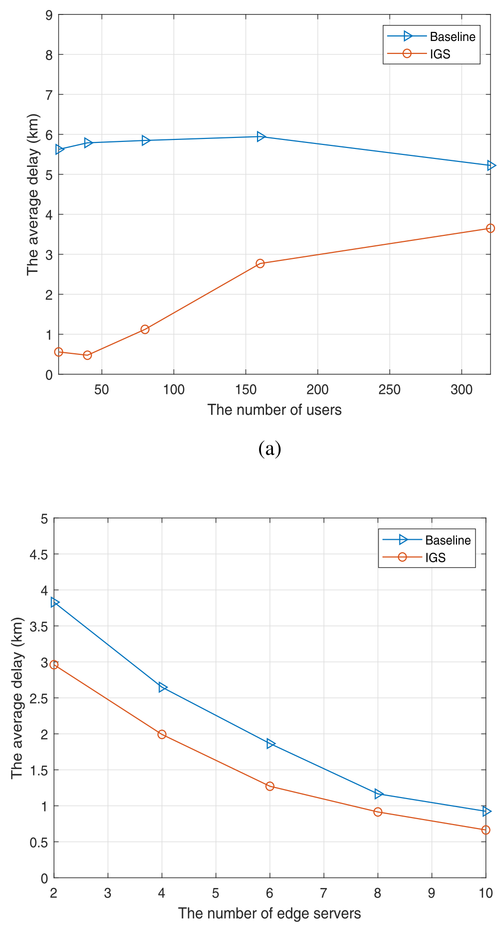
We evaluate the performance of the proposed algorithm IGS against benchmarks in two scenarios: a small-scale problem and a big-scale problem. In the first scenario, the benchmarks are the optimal solution OPT and the resource allocation algorithm proposed in [16] which is the most similar state of the art and denoted as Baseline in this article. The optimal solution OPT to the resource allocation and pricing problem defined in Section III-B is obtained via CPLEX 12.8.0 [28]. Baseline applies many-to-many matching to solve the resource allocation problem. Baseline obtains the resource price via the equilibrium analysis of a modeled game that is not applicable in this article. Therefore, we use the proposed resource pricing method in this article to decide the resource price for Baseline. In the second scenario, we only compare algorithm IGS with Baseline since we cannot obtain the optimal solution when the problem has a big scale. We use the physical distance between the user and the allocated server as an approximation of the latency [26], [29].

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

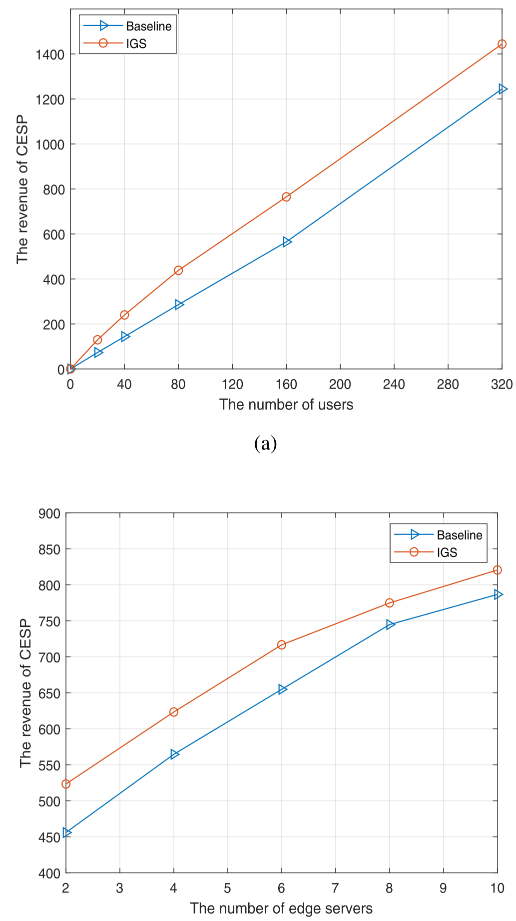
Fig. 12. Revenue of CESP with a small-scale problem. (a) Different parameters α. (b) Different average unit energy consumption of computing resources at the edge servers. (c) Different parameters μ.



(a)

(b)

Fig. 13. Average delay between users and servers with a large-scale problem. (a) Different number of users. (b) Different number of edge servers.



(a)

(b)

Fig. 14. Revenue of CESP with a large-scale problem. (a) Different number of users. (b) Different number of edge servers.

1) Small-Scale Problem: Fig. 12 shows the CESP's revenue performance of algorithms IGS, Baseline, and OPT versus different parameters with a small-scale problem, assuming that $M = 20$ and $\mu = 1$. In general, IGS performs better than Baseline and close to OPT. Fig. 12(a) shows that the CESP's revenues of all the three algorithms increase with the increase of revenue parameter α. Algorithm IGS outperforms algorithm Baseline by up to 15.3%. Fig. 12(b) shows that the CESP's revenues of the three algorithms decrease with the increase of average unit energy consumption of computing resources at the edge servers. Algorithm IGS achieves very close performance to OPT and increasingly better performance than Baseline. In particular, algorithm IGS is better than algorithm Baseline up to 83.9%. Fig. 12(c) shows that the CESP's revenues of the three algorithms increase with the increase of parameter μ. Algorithm IGS outperforms Baseline from 0.4% to 8.4%.

2) Big-Scale Problem: Fig. 13(a) shows the average communication delay between users and servers obtained by algorithms IGS and Baseline by varying the number of users,

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

assuming that $\alpha = 15$ and $\mu = 1$. The average communication delay obtained by algorithm IGS is significantly lower than that obtained by Baseline. Fig. 13(b) shows the average communication delay between users and servers obtained by algorithms IGS and Baseline with a different number of edge servers, assuming that $M = 160$, $\alpha = 15$, and $\mu = 1$. The average communication delay obtained by the two algorithms decreases with the increasing number of edge servers. With more edge servers, the users' tasks are more likely be deployed on edge servers. The average communication latency obtained by algorithm IGS is better than that obtained by Baseline by 29.3%–46.7%.

Fig. 14(a) shows the CESP's revenue performance of algorithms IGS and Baseline by varying the number of users with a large-scale problem, assuming that $\alpha = 15$ and $\mu = 1$. The CESP's revenues of the two algorithms increase with the increase of the number of users since more users will generate more computing resource requests and the CESP can have more income. The improvement of algorithm IGS over Baseline is up to 77.1%. Fig. 14(b) shows the CESP's revenue of algorithms IGS and Baseline with a different number of edge servers, assuming that $M = 160$, $\alpha = 15$, and $\mu = 1$. The CESP's revenues of the two algorithms increase with the increasing number of edge servers. With more edge servers, both of the two algorithms have more options to deploy the users' tasks. The results of algorithm IGS are always better than those of Baseline. In particular, algorithm IGS outperforms Baseline by 4.0%–14.7%.

VI. Conclusion and Future Work

The introduction of cloud/edge computing paradigm makes mobile blockchain possible, where the cloud/edge servers execute the services offloaded by the IoT users. The CESP provides computing resources to IoT users with a cloud and multiple edge servers that work collaboratively. It is critical for the CESP to decide the resource price and which server to run the offloaded tasks to maximize the revenue. In this article, we formulated a Stackelberg game with CESP as the leader and users as followers for cloud/edge computing resource management. We proved the existence of Stackelberg equilibrium and analyzed the equilibrium. We then modeled the resource allocation and pricing at the CESP as an MIP with the objective to optimize the CESP's revenue and proposed an efficient iterative greedy-and-search-based resource allocation and pricing algorithm (IGS). The algorithm solves two subproblems comprising the CESP's revenue optimization problem: resource allocation under a given resource price to decide where to run the offloaded tasks and resource pricing based on a specified resource allocation scheme. Algorithm IGS uses a greedy-and-search-based approach to solve the first subproblem and adopts golden section search to tackle the second subproblem. Simulation results showed that the proposed algorithm could effectively increase the revenues of both the CESP and the users, the revenues of both the CESP and the users increase with the increase of revenue parameter α_i of users and the computing power of the servers, and the proposed algorithm could also effectively reduce the delay.

In this article, we studied the problem of resource management for mobile blockchain when the computation is offloaded to a server from either the edge servers or the cloud. Note that an edge server can also offload the computation to the other edge servers or the cloud or both when the edge server is overloaded. We will study the resource management problem under the new computation offloading model in the future.

References

[1] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Jan. 9, 2021. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[2] M. Swan, *Blockchain: Blueprint For a New Economy*. Newton, MA, USA: O'Reilly Media, 2015.

[3] W. Wang *et al.*, "A survey on consensus mechanisms and mining management in blockchain networks," 2018, pp. 1–33, *arXiv:1805.02707*. [Online]. Available: https://arxiv.org/pdf/1805.02707

[4] R. Arenas and P. Fernandez, "CredenceLedger: A permissioned blockchain for verifiable academic credentials," in *Proc. IEEE Int. Conf. Eng., Technol. Innov. (ICE/ITMC)*, Jun. 2018, pp. 1–6.

[5] *Coinmarketcap*. Accessed: Dec. 30, 2019. [Online]. Available: https://coinmarketcap.com

[6] K. Abdellatif and C. Abdelmouttalib, "Graph-based computing resource allocation for mobile blockchain," in *Proc. 6th Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2018, pp. 1–4.

[7] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, Aug. 2018.

[8] M. Liu, F. R. Yu, and Y. E. A. Teng, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11008–11021, Nov. 2018.

[9] R. Recabarren and B. Carbunar, "Hardening stratum, the bitcoin pool mining protocol," *Proc. Privacy Enhancing Technol.*, vol. 2017, no. 3, pp. 57–74, Jul. 2017.

[10] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.

[11] R. Casado-Vara, F. de la Prieta, J. Prieto, and J. M. Corchado, "Blockchain framework for IoT data quality via edge computing," in *Proc. 1st Workshop Blockchain-Enabled Netw. Sensor Syst.*, Nov. 2018, pp. 19–24.

[12] R. Chamarajnagar and A. Ashok, "Opportunistic mobile IoT with blockchain based collaboration," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[13] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[14] J.-Y. Kim and S.-M. Moon, "Blockchain-based edge computing for deep neural network applications," in *Proc. Workshop Intell. Embedded Syst. Architectures Appl.*, Oct. 2018, pp. 53–55.

[15] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/Fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4585–4600, Jun. 2019.

[16] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[17] S. Dhamal, W. Ben-Ameur, T. Chahed, E. Altman, A. Sunny, and S. Poojary, "A stochastic game framework for analyzing computational investment strategies in distributed computing," 2018, *arXiv:1809.03143*. [Online]. Available: http://arxiv.org/abs/1809.03143

[18] J. Chiu and T. Koeppl, "Incentive compatibility on the blockchain," in *Social Design*. Cham, Switzerland: Springer, 2019, pp. 323–335.

[19] 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.*, vol. 30, no. 9, pp. 1975–1989, Sep. 2019.

[20] A.-L. Jin, W. Song, and W. Zhuang, "Auction-based resource allocation for sharing cloudlets in mobile cloud computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 45–57, Jan. 2018.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.

[21] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato, "Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[22] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Jan. 2019.

[23] Y. Wu *et al.*, "Optimal computational power allocation in multi-access mobile edge computing for blockchain," *Sensors*, vol. 18, no. 10, p. 3472, Oct. 2018.

[24] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, and Y. Zhao, "EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4719–4732, Jun. 2019.

[25] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3602–3609, Jun. 2019.

[26] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han, "Fog computing in multi-tier data center networks: A hierarchical game approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

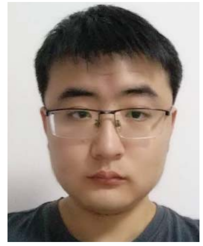
[27] L. Kiefer, "Sequential minimax search for a maximum," *Proc. Amer. Math. Soc.*, vol. 4, no. 3, pp. 502–506, Jun. 1953.

[28] *CPLEX Optimizer*. Accessed: Feb. 28, 2020. [Online]. Available: http://https://www.ibm.com/analytics/cplex-optimizer

[29] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, Helsinki, Finland, 2012, pp. 7–12.

Yuqi Fan received the B.S. and M.S. degrees in computer science and engineering from the Hefei University of Technology, Hefei, China, in 1999 and 2003, respectively, and the Ph.D. degree in computer science and engineering from Wright State University, Dayton, OH, USA, in 2009.

He is currently an Associate Professor with the School of Computer and Information Engineering, Hefei University of Technology, Hefei, China. His research interests include blockchain, cloud computing, social networks, computer networks, software-defined networks (SDNs), and the Internet of Things (IoT).

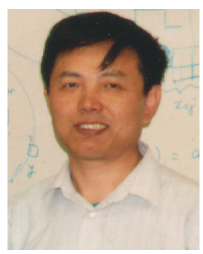


Lunfei Wang received the B.S. degree in information security from Qingdao University, Qingdao, China, in 2018. He is currently pursuing the master's degree in computer science with the School of Computer and Information Engineering, Hefei University of Technology, Hefei, China.

His research interests include blockchain, cloud computing, and software-defined networks.

Weili Wu (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science and engineering from the University of Minnesota, Minneapolis, MN, USA, in 1992 and 2002, respectively.

She is currently a Professor with the Department of Computer Science, The University of Texas at Dallas, Richardson, TX, USA. Her research interests include blockchain, social networks, cloud computing, data science, and machine learning.



Dingzhu Du received the M.S. degree in operations research from the Chinese Academy of Sciences, Beijing, China, in 1982, and the Ph.D. degree in mathematics from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 1985.

He is currently a Professor with the Department of Computer Science, The University of Texas at Dallas, Richardson, TX, USA. His research interests include blockchain, social networks, cloud computing, data science, and machine learning.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on July 25,2021 at 23:41:53 UTC from IEEE Xplore. Restrictions apply.