Resource Pricing and Allocation in MEC Enabled Blockchain Systems: An A3C Deep Reinforcement Learning Approach

Jianbo Du, Wenjie Cheng, Guangyue Lu*, Haotong Cao, Xiaoli Chu, Zhicai Zhang, and Junxuan Wang*

***Abstract*—When using blockchain in mobile systems, computation intensive mining tasks pose great challenges to the processing capabilities of mobile miner equipment. Mobile edge computing (MEC) is an effective solution to alleviating the problem via task offloading. In the mining process, miners compete for rewards through puzzle solving, where only the miner that first completes the process will be rewarded. Thus, miners may wish to pay higher price and use more communication resources in task offloading and more computation resources in task processing for latency reduction. However, there are risks for the miners not profiting from consuming more resources or paying a higher price, so miners are rational in blockchain systems. In order to maximize the rational total profit of all miners, we use an asynchronous advantage actor-critic (A3C) deep reinforcement learning algorithm to obtain the resource pricing and allocation, considering the stochastic properties of wireless channels, and the prospect theory is employed to strike a good balance between risks and rewards. Numerical results show that our proposed A3C based joint optimization algorithm converges fast and outperforms the baseline algorithms in terms of the total reward.**

***Index Terms*—Asynchronous advantage actor-critic (A3C), blockchain, deep reinforcement learning, mobile edge computing, pricing, resource allocation.**

I. Introduction

With the fast development of digital transactions, electronic trading has turned its market from human intermediation to computer processing, where cryptocurrencies are traded among users over a peer-to-peer network. In order to support this financial service migration, a credible and flexible trading platform is urgently required for efficient transaction management [1]. Blockchain, a public distributed ledger, has been introduced recent years in order to meet this demand. Different from traditional centralized digital ledger, in blockchain systems, various kinds of transaction data blocks are recorded and confirmed distributedly, without dependent on any trustful third parties such as banks, financial regulatory authorities, etc [2], [3]. As a result, blockchain could reduce the costs in transaction processing greatly and improve the efficiency in record keeping effectively.

The operation, reliability and security issues of blockchain networks relay on a distributed consensus mechanism. In blockchain system, a group of participants, also called miners, manage to solve a computation intensive problem, e.g., the proof-of-work (PoW) puzzle, and the process is referred to as mining [4], [5]. In this process, each miner first selects and packages certain number of unconfirmed transaction records into a new block, and then it solves the PoW puzzle based on the value of the new block. Immediately the puzzle is settled, the miner will broadcast the newly generated block that integrates the transactions and relevant information to the network. Finally, the rest miners will verify the block for consensus, and the block will be appended to the blockchain if it passes the validation of most other miners. The miner that first completes the process will receive corresponding rewards as the incentive of mining.

As a public ledger, blockchain has heralded various sorts of commercial services such as bitcoin, filecoin, etc. However, its apply in mobile environments is still limited since the mining process, i.e., settling the PoW puzzle, requires powerful computation capabilities and consumes large quality of energy beyond the affordability of mobile miners, making it rather challenging to deploy blockchain in mobile systems. Recent years, mobile edge computing (MEC) [6], [7], which appears as a promising supplement to cloud computing [8], could provide computation capabilities in proximity to mobile subscribers, is becoming a powerful enabler for successful deployment of mobile blockchain systems [9]. In MEC enabled blockchain systems, computation intensive mining tasks can be offloaded from mobile miners to MEC servers, and be executed with stronger processing capabilities, therefore, mobile blockchain can be enabled, and more energy consumption can be saved for mobile miner devices [10], [11].

In the mining competition, only the miner that first completes the mining process could add its transaction records into blockchain and obtain the mining rewards. In order to win the competition, a miner could require MEC server for more wireless resource for task offloading [12], and more

*This work was supported in part by the Natural Science Foundation of Shaanxi Province under Grant 2020JQ-844, in part by the Natural Science Foundation of China under Grant 61901367, 61871321, and 62001357, in part by the Science and Technology Innovation Team of Shaanxi Province for Broadband Wireless and Application under Grant 2017KCT-30-02, in part by the National Science and Technology Major Project under Grant 2016ZX03001016. *(Corresponding author: Guangyue Lu and Junxuan Wang.)*

J. Du, W. Cheng, G. Lu and J. Wang are with Shaanxi Key Laboratory of Information Communication Network and Security, School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China. (Email: dujianboo@163.com; Chengwenjie1998@163.com; tonylugy@163.com; wangjx@xupt.edu.cn)

H. Cao is with College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. (caohaotong@163.com)

X. Chu is with Department of Electronic and Electrical Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK. (Email: x.chu@sheffield.ac.uk).

Z. Zhang is with the School of Physics and Electronic Engineering, Shanxi University, Taiyuan 030006, China (e-mail: zzcai@sxu.edu.cn).

computation resources for task processing. Also, it could pay higher price to MEC server so as to get higher chances in winning the competition of obtaining more resources. Thus, the joint optimization of pricing and allocation about the limited computation and communication resource arises as an important issue. Meanwhile, different miners have different preferences towards the trade-off between risks and rewards, which is also an nonnegligible factor. Moreover, due to the dynamics of mobile networks, resource pricing and allocation should be updated constantly to adopt to the rapidly changed environment. The above factors make the problem intractable when employing traditional optimization methods to solve. Fortunately, artificial intelligence [13], [14], federal learning [15] [16], especially deep reinforcement learning (DRL) algorithms [17], [18], have exhibited their efficiency in solving intractable policy-making problems with high dimensional state and action spaces, especially demonstrated their effectiveness for the issues with continuous state and action spaces.

In this paper, we investigate the joint optimization of pricing and allocation about both communication and computation resources in an MEC enabled blockchain system, with user preferences between risks and rewards considered. Based on DRL, we proposed effective algorithm to solve it. The main contributions of this paper are summarized as follows.

- We devise a system model for an MEC enabled mobile blockchain system where the mining task is offloaded to the MEC server for speedly cooperative local-MEC processing. We propose to maximize the long-term averaged rational total reward of all the miners by jointly optimizing the allocation of computational resource blocks and wireless subchannels, and the pricing strategy of each miner for both the computation and communication resources, which is described as a Markov decision process (MDP). Moreover, the prospect theory is employed to strike a balance between risks and the uncertain rewards according to different preferences of different miners.
- Considering the mixed integer properties and the high dynamics of wireless channels, we solve the joint resource pricing and allocation problem by employing the asynchronous advantage actor-critic (A3C) algorithm, where each agent is composed of two deep neural networks: one is used as the function approximator to estimate the value functions in the critic part, and the other is used as a parameterized stochastic policy in the actor part. The multiple agents are trained asynchronously using policy gradient algorithm.

The remainder of this paper is organized as follows. Section II presents the related works. Section III elaborates our system model and problem formulation. Section IV reformulates the problem as a model-free DRL problem, and solves it using our proposed A3C based joint resource pricing and allocation algorithm. Our simulation results are presented in Section V, and the paper is summarized in Section VI.

II. Related Works

Recently, the study of MEC empowered blockchain has become a hot research topic. In the related works of this topic, offloading the computation intensive mining tasks to cloud or MEC servers to relieve the pressure of mobile miners is the main idea [19], and different approaches have been adopted to solve the problem of offloading decision optimization and resource allocation for different goals, such as throughput maximization, profit maximization, etc [20].

Convex optimization has been employed in may works focusing on the short-term performance improvement. In [21], the authors proposed to achieve the optimal trade-off between the performance of the MEC system and that of the blockchain system, through jointly optimizing user association, resource allocation, and block producer scheduling, and developing effective iterative algorithms for problem solving. In [22], the authors proposed to offload the mining tasks to and cache the cryptographic hashes of blocks on MEC server, and propose an alternating direction method of multipliers based algorithm for the MEC server's profit maximization.

Game theory is also a common method focusing on short-term performance optimization in blockchain based MEC systems. In [23], the authors intended to obtain the optimal auction based resource allocation, and thus to optimize the expected revenue of the edge computing service provider in mobile blockchain networks. In [24], the authors investigated the pricing based computation resource allocation to support PoW mining tasks offloading in a mixed cloud/fog system, where the problem is formulated as a two stage Stackelberg game and the profit of cloud/fog srever and the utility of miners are maximized. In [25], the authors formulated the mining task offloading issue as a Stackelberg game based on prospect theory, in order to maximize the utilities of both miners and MEC service providers.

Some other works put their focus on the long-term performance optimization by employing DRL in MEC enabled blockchain systems. In [26], the authors studied the wireless spectrum allocation, block size and the number of consecutive blocks optimization based on deep Q network (DQN) to improve the throughput of the overlaid blockchain system along with the quality of services of the users in the underlaid MEC system. In [27], the authors combined genetic algorithms into DRL to speed up the exploration process, employing which the offloading decision of both mining tasks and data processing tasks are optimized. In [28], miners offload their mining tasks to cloud servers for performance enhancement, where access control, computing and networking resources allocation are jointly optimized in the process of mining task offloading using a dueling DQN based approach.

The above works have presented some insightful ideas for task offloading decision making and/or resource allocation in MEC enabled blockchain systems [21]–[28], and it can also be known from the above works that pricing strategies also plays important roles in blockchain systems where economical profit is usually the major concern. Considering the long-term performance improvement and the high dynamics, DRL plays their effect, where A3C is an excellent rising method that could adopt to both continuous and high-dimensional discrete action and/or state space, compared with other traditional DRL algorithms. Motivated by the above considerations, in this paper, we intend to study the communication and computation

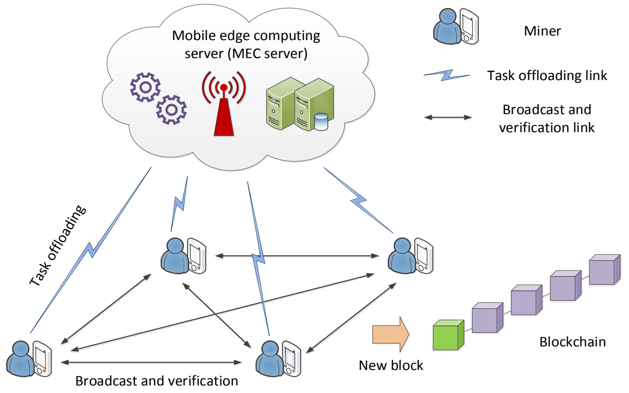

Mobile edge computing server (MEC server)
Miner
Task offloading link
Broadcast and verification link
Task offloading
Blockchain
New block
Broadcast and verification

Fig. 1: System model of MEC enabled mobile blockchain system.

resource allocation issues and the pricing strategies of the two kinds of resources in an MEC based blockchain system, in order to maximize the reward of miners, and we propose an efficient A3C based DRL algorithm for problem solving and long-term performance optimization.

III. System Model and Problem Formulation

In this section, we first introduce the system model, including the delay of of mining, the probability of each miner in wining the mining competition, and the utility gains of mining. Then, we present our problem formulation.

Fig. 1 illustrates our concerned MEC based mobile blockchain system. There is one blockchain mining task, which is described in a two-terms tuple as $\Lambda = \{C, D\}$. Here, C is the required computation amount (in CPU cycles) to complete the mining task, and D is the size of the input data (in bits) of this task. There are N mobile devices acting as miners to compete for completing the task, and also acting as block producers to record the generated transactions into a blockchain. The processing capacity of each miner device n is denoted by f_n^{loc}, $n \in \mathcal{N} = \{1, 2, ..., N\}$, where $\mathcal{N}$ is the set of miner devices. There is one Edge Computing Service Provider (ECSP) that owns an MEC server, and provides wireless communication and computation resources to miners for task offloading and processing, and charge corresponding fees from the miners. The wireless communication resource and the computation resource of the MEC server is limited, so the miners will have to compete for resource utilization. If a miner obtains some computation resource from the MEC server, the obtained computation resource will be combined with the feasible local computation resources to accelerate the task execution in the mining process, which is called local-edge collaborative task processing mode in this paper.

A. Delay in Different Steps of Mining

Generally, a successful mining process usually comprises three steps, i.e., the processing step, the propagation step, and the validation step [24], [25]. In the processing step, a block is generated by a miner through solving the PoW puzzle of the chosen task. Then the miner propagates its newly mined block to other miners for validation. When all the rest miners (or the majority of the rest miners) have validated the block as a consensus, the block will be appended to the blockchain, and the first miner whose task reaches consensus will obtain the corresponding reward. In this paper, we assume that all validators are honest, and a block should be validated by all the other miners to reach consensus. Correspondingly, the total mining delay consists of three parts, i.e., the task processing delay, the propagation delay, and the verification delay. However, there's something different in local-edge collaborative task processing mode.

In our MEC enabled collaborative task processing blockchain system, the computation resources that each miner utilizes for task processing come form two parts, i.e., the local feasible computation resource and the computation resource allocated by the MEC server. To enable collaborative task processing, a copy of the input data should be transmitted to MEC server, this is called task offloading process; then task is processed with the computation resources of local and MEC server collaboratively, this is called task processing process. Consequently, besides the three steps mentioned above, there first should be a task offloading process, and correspondingly, there should be an additional task offloading delay, too. Next, we will discuss the four delays in the following.

1) Delay in Task Offloading Process: Similar to many previous works, in this paper, time is slotted and we denote the index of a time slot as t, and correspondingly, we denote the set and number of time slots as $\mathcal{T}$ and T, respectively. Similarly to many previous works [29], [30], we use the quasi-static assumption that the environment remains unchanged during each time slot while changes between different time slots. We assume that the total wireless bandwidth is divided into B subchannels, and each subchannel is with a bandwidth B_0. In time slot t, denote the average signal-to-noise ratio (SNR) between miner n and the MEC server as $\gamma_n(t)$, and assume miner n is allocated with $b_n(t)$ orthogonal subchannels, then the wireless data rate of miner n in task offloading can be given by

$$r_n(t) = b_n(t) B_0 \log_2 \left(1 + \gamma_n(t)\right), \tag{1}$$

and the task offloading delay T_n^t is given by

$$T_n^t(t) = \frac{D}{r_n(t)}. \tag{2}$$

2) Delay in Task Processing Process: Denote the processing capability of MEC server as F (in CPU cycles per second), and assume that F can be divided into F/F_0 computational resource blocks, where each is with a processing capability F_0. In our local-edge collaborative task processing mode, we assume that each miner takes half its processing capabilities, i.e., $f_n^{loc}/2$, as the feasible local computation resource for mining task processing, and let $f_n(t)$ denote the number of computational resource blocks allocated to miner n by MEC server for task processing, the processing delay is given by

$$T_n^c(t) = \frac{C}{F_0 f_n(t) + f_n^{loc}/2}. \tag{3}$$

3) Delay in Block Propagation Process: In the block propagation process, we denote the wireless transmit rate from miner n to miner m as $r^p_{n,m}(t)$, where $n, m \in \mathcal{N}, m \neq n$. Then, the wireless propagation rate of miner n is given by $r^p_n(t) = \min\limits_{m\in\mathcal{N},m\neq n}\{r^p_{n,m}(t)\}$ [21]. Denote the size of a block as χ (in bits), the delay in block propagation can be given by

$$T^p_n(t) = \frac{\chi}{r^p_n(t)}. \tag{4}$$

4) Delay in Block Verification Process: In block verification process, we only focus on the computational delay of the cryptographic operations as in [21]. Assume the required number of CPU cycles in block verification is φ, then the delay of block validation is given by

$$T^v_n(t) = \max_{m\in\mathcal{N},m\neq n} \frac{\varphi}{f^v_{n,m}(t)}, \tag{5}$$

where $f^v_{n,m}(t)$ is the computation resource afforded by validator m for block verification, and $f^v_{n,m}(t) = f^{loc}_n/2N$ in this paper. Denote $f^v_n(t) = \min\limits_{m\in\mathcal{N},m\neq n}\{f^v_{n,m}(t)\}$ [21], we have

$$T^v_n(t) = \frac{\varphi}{f^v_n(t)}. \tag{6}$$

Therefore, the total delay of miner n in successfully complete a transaction is given by

$$\begin{aligned} T^{total}_n(t) &= T^t_n(t) + T^c_n(t) + T^p_n(t) + T^v_n(t) \\ &= \frac{D}{r_n(t)} + \frac{C}{F_0 f_n(t) + f^{loc}_n/2} + \frac{\chi}{r^p_n(t)} + \frac{\varphi}{f^v_n(t)}. \end{aligned} \tag{7}$$

The probability that miner n wins the mining competition is given by [25]

$$p_n(t) = \frac{\alpha^{-T^{total}_n(t)}}{\sum_{i=1}^N \alpha^{-T^{total}_n(t)}}, \tag{8}$$

where α is a coefficient, and $\alpha > 1$.

B. Utility Gain

Let e denote the reward that the winner obtains in the mining competition. Since all the miners process the same mining task as mentioned above, when SNR $\gamma_n(t)$ is larger, i.e., the miner n is in good channel condition, it has higher incentive to use more wireless subchannels for task offloading, since its transmit delay can be reduced greatly. Meanwhile, it also has the incentive to pay higher price in order for more chance to be allocated with more subchannels. Similarly, when local processing capability f^{loc}_n is not strong enough, the miner n has stronger motivation to use more edge computing resource blocks for task processing delay reduction. In order to facilitate this desire, it is much willing to pay higher price for higher probabilities to win the mining task processing.

Denote the price miner n pays for using an unit wireless resource, i.e., an unit wireless transmit rate, as $\eta_n(t)$ (in \$/bps), and denote the price miner n pays for using an unit computation resource as $v_n(t)$ (in \$/(CPU cycles per second)). In order to express the incentive of the price each miner pays on the probability of wining the competition, we modify the price-based total delay of miner n as

$$\begin{aligned} &T^{total}_{n,price}(t) = T^t_{n,price}(t) + T^c_{n,price}(t) + T^p_n(t) + T^v_n(t) \\ &= \frac{D}{r_n(t)\eta_n(t)} + \frac{C}{F_0 f_n(t) v_n(t) + f^{loc}_n/2} + \frac{\chi}{r^p_n(t)} + \frac{\varphi}{f^v_n(t)}, \end{aligned} \tag{9}$$

noting that in the above equation, in order to keep logic consistency, both $\eta_n(t)$ and $v_n(t)$ only act as weight values of $r_n(t)$ and $f_n(t)$, without considering their unit, and thus the unit of $T^t_{n,price}(t)$ and $T^c_{n,price}(t)$ is still second. Similarly, the price-based probability of miner n in wining the competition is modified as [25]

$$p_{n,price}(t) = \frac{\alpha^{-T^{total}_{n,price}(t)}}{\sum_{i=1}^N \alpha^{-T^{total}_{n,price}(t)}}. \tag{10}$$

Based on the above definitions, the expected utility gained by miner n in the mining competition is given by

$$g_n(t) = p_{n,price}(t)e - F_0 f_n(t) v_n(t) - r_n(t)\eta_n(t). \tag{11}$$

As mentioned above, only the miner who first completes the mining process will obtain a reward. Consequently, although there are some chances for miners to obtain high reward through mining, however, there are still high risks when nothing can be obtained after the miners have spent some money on using communication and computation resources for task offloading and processing. Actually, different miners have different preferences for the tradeoff between risks and rewards. Therefore, when a miner assesses the effectiveness of the price they should pay for communication and computation resources, and how much resources they should request, each miner's economical preference needs to be taken into account, i.e., each miner is rational in mining decision making. For this purpose, we resort to prospect theory [25], i.e., a behavioral economical theory which describes how people make their decisions in the consideration of risks and the uncertain goals. We adopt the prospect theory as in [25], where each miner considers its prospect through the following function, which transform the obtained gain of each miner to an utility value, and the output utility value is determined based on a reference point. We set the reference point as zero gain value, and the utility of each miner is given by

$$U_n(t) = \begin{cases} g_n(t)^{\mu_n}, & g_n(t) \geq 0 \\ -\lambda_n(-g_n(t))^{\xi_n}, & g_n(t) < 0 \end{cases}, \tag{12}$$

where μ_n, ξ_n and λ_n are devices n's prospect parameters, and we have $0 < \mu_n, \xi_n < 1$ and $\lambda_n > 1$, respectively.

IV. Problem Formulation

In this section, we present our problem formulation. As was analyzed, in order to reduce the mining delay, and thus to win the mining competition and obtain the incentive reward e, each miner focuses on using more communication and computation resources to reduce the task offloading and processing delay, and pay higher price to increase the opportunities of being allocated with more resources. However, this will lead to a reduction in their obtained utility gains as in (11). In order to optimize the long-term averaged rational system utility of all miners in the system, we formulate the joint optimization problem of computation resource allocation, wireless subchannel assignment, and the price each user would like to pay for using each unit computation and communication resources, respectively. Since our optimization involves both continuous and integer variables, it is generally NP-hard and thus intractable to solve [31], [32].

A. DRL

DRL is a learning scheme where an agent interacts with the environment over a series of discrete episodes. At each episode, the agent takes an action under the current state according to a certain policy, then the environment will transforms to a new state, and will return the agent with an immediate reward, based on which the network is updated, in order to maximize the long-term expected cumulative reward that the agent could get.

Combing deep learning [33] and reinforcement learning, DRL can enhance the performance of traditional reinforcement learning algorithms when the environment contains high dimension input or large action sets, and it is especially effective in solving intractable problems containing both continuous and integer variables.

For this purpose, we resort to DRL and formulate our joint optimization problem as an MDP [29], [34]. Our state space, action space, policy, problem formulation and reward function are defined as follows.

B. State Space

Let $\mathcal{S} = \{s(t), t \in \mathcal{T}\}$ denote our system space, and $s(t)$ is the state at the tth time slot, which includes the following parameters.

- Average SNR between each miner and the MEC server: $\boldsymbol{\gamma}(t) = \{\gamma_n(t)\}$;
- The transmit rate of each miner in block propagation: $\mathbf{r}^p(t) = \{r^p_{n,m}(t)\}$;
- Feasible channel number allocation set: $\mathbf{N}_b(t)$;
- Feasible computation resource block allocation set: $\mathbf{N}_f(t)$.

Thus, the system state at time slot t is denoted as $s(t)$, which is given by

$$s(t) \triangleq \{\boldsymbol{\gamma}(t), \mathbf{r}^p(t), \mathbf{N}_b(t), \mathbf{N}_f(t)\}, \tag{13}$$

and is known at the beginning of each time slot t.

C. Action Space

Let $\mathcal{A} = \{a(t), t \in \mathcal{T}\}$ denote the action space. At each time slot t, the action $a(t) \in \mathcal{A}$ comprises the following items.

- The computation resource allocation: $\mathbf{f}(t) = \{f_n(t)\}$, where each term $f_n(t)$ denotes the number of computation resource blocks allocated to miner n;
- Wireless subchannel allocation: $\mathbf{b}(t) = \{b_n(t)\}$, where $b_n(t)$ is the number of subchannels assigned to miner n;
- The price of computation resources: $\mathbf{v}(t) = \{v_n(t)\}$, where $v_n(t)$ represents the price that miner n would like to pay for each unit computation resource, and its unit is \$/(CPU cycles per second);
- The price of communication resource: $\boldsymbol{\eta}(t) = \{\eta_n(t)\}$, where $\eta_n(t)$ is the price miner n would like to pay for each unit communication resource, with its unit as \$/bps.

Thus, the action at time slot t is given by

$$a(t) \triangleq [\mathbf{f}(t), \mathbf{b}(t), \mathbf{v}(t), \boldsymbol{\eta}(t)]. \tag{14}$$

D. Policy

The policy is a mapping operator from state space to action space, which can be denoted as $\pi(a(t)|s(t)) : \mathcal{S} \rightarrow \mathcal{A}$.

E. Problem Formulation

In this paper, we intend to maximize the long-term averaged rational system utility of all miners by performing decision searching within the action space, under the information of state space. Our problem is formulated as

$$\begin{aligned} (\mathcal{P}_1): \ & \max_{a(t)} \frac{1}{T} \sum_{t\in\mathcal{T}} \sum_{n\in\mathcal{N}} U_n(t) \\ \text{s.t. (C1)}: \ & \sum_{n\in\mathcal{N}} f_n(t) F_0 \le F, t \in \mathcal{T}, \\ \text{(C2)}: \ & \sum_{n\in\mathcal{N}} b_n(t) B_0 \le B, k \in \mathcal{K}, t \in \mathcal{T}, \\ \text{(C3)}: \ & f_n(t) \in \mathcal{F}, \\ \text{(C4)}: \ & b_n(t) \in \mathcal{B}, \\ \text{(C5)}: \ & v_n(t) \in [v_{min}, v_{max}], t \in \mathcal{T}, \\ \text{(C6)}: \ & \eta_n(t) \in [\eta_{min}, \eta_{max}], t \in \mathcal{T}, \end{aligned} \tag{15}$$

In problem $(\mathcal{P}_1)$, (C1) constrains that the allocated computation resource could not exceed the processing capacity of the MEC server; (C2) indicates that the allocated radio resource could not be greater than the total system bandwidth; (C3) and (C4) indicates the number of computation resource blocks and subchannels each user can rent can only be allocated from set $\mathcal{F}$ and $\mathcal{B}$, respectively; and (C5) and (C6) give the constraint on the price each miner pays for using each unit computation and communication resource, where v_{min} and η_{min} are the minimum price of MEC server could accept, and v_{max} and η_{max} are the maximum price miners could afford.

F. Reward Function

In order to utilize DRL algorithms to solve our formulated problem $(\mathcal{P}_1)$, we need to transform it into standard module of DRL framework. In our problem formulation, the optimization variables and the known quantities have been corresponded to the action space and the state space, respectively. In $(\mathcal{P}_1)$, we intend to maximize the long-term averaged rational system utility of all miners, which motivates us to consider the immediate total utility $\sum_{n\in\mathcal{N}} U_n(t)$, i.e., the total rational utility of all the miners in time slot t, as the immediate reward $r(t)$. Therefore, the formulated problem is transformed into a standard DRL fremework. When an action $\mathbf{a}(t)$ is taken, the

DRL agent will receive an immediate reward as $r(t)$, which is defined as follows as was analyzed

$$r(t) = \sum_{n \in \mathcal{N}} U_n(t). \tag{16}$$

V. A3C Based Joint Optimization Algorithm

In this section, we first present the newly emerging DRL algorithm A3C, and then we introduce our A3C based joint resource allocation and pricing algorithm to solve our formulated problem $(\mathcal{P}_1)$.

Remark 1: *In the following, time indexes are represented using subscripts rather than using parentheses as above sections for notational simplicity.*

A. A3C Based Joint Resource Pricing and Allocation Framework

Before we introduce the principle of A3C algorithm, we will first briefly introduce its basis algorithm, named Actor-Critic (AC) [35]. Similar to other DRL algorithms, there's an agent which interacts with the environment by states, actions, and rewords, in order to maximize the discount return. In AC framework, the agent comprises an actor and a critic, where in each episode, the actor performs an action under the current state using the current the policy, and the environment will transforms to a new action and will return the critic with a reward. The critic is updated using TD algorithm, in order for better judge and grade capabilities, and the actor is updated using policy gradient method in order for higher return.

A3C [36] is proposed on the basis of AC algorithm [37], where the difference is that A3C employs multiple actors to work concurrently, and trains the neural networks of the multiple actors asynchronously, thus to be able to accelerate the convergence significantly. In A3C algorithm, there's a central server which stores the network parameters. Each agent obtains its gradients and sends them to the central server when the maximum action index or the terminal state is reached. The central controller updates the global parameters and distributes them to the agents to guarantee they can share the same policy. In this way, the parameters are less correlated than single agent does, making there's no need to keep a replay memory as traditional DQN [38] does. Moreover, the training duration can also be reduced greatly.

Remark 2: *A3C outperforms other many existing DRL algorithms due to its policy-based and step-based updating, and asynchronous training. Due to policy based character, A3C could search in continuous spaces, and can deal with problems with huge state or action spaces. Due to step-based updating, its working efficiency is greatly enhanced. Also, multiple agents work in parallel could accelerate the training efficiently, and explore the solution space effectively.*

Next, we will give the working process of A3C.

At each time slot t, the environment is in state s_t, and the estimated state value is $V(s_t;\theta_v)$. The agent executes an action a_t according to policy $\pi(a_t|s_t;\theta)$ under the current state s_t, and the environment will transfer to a following state s_{t+1} under certain probabilities, and the agent will receive a reward r_t. The state value function of A3C is given by

$$\begin{aligned} V(s_t;\theta_v) &= E\left[G_t|s=s_t,\pi\right] \\ &= E\left[\sum_{k=0}^{\infty}\gamma^k r(t+k)\Big|s=s_t,\pi\right], \end{aligned} \tag{17}$$

where $G_t = \sum_{k=0}^{\infty}\gamma^k r_{t+k}$ is the discounted return, i.e., discounted accumulated reward, of state s_t, and $\gamma \in [0,1]$ is the discount factor, indicating how the future rewards influence the current state value.

A3C utilizes k-step reward for parameter updating, which is given by

$$R_t = \sum_{i=0}^{k-1}\gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta_v), \tag{18}$$

where k is upper-bounded by t_{max}, and both the policy and value function are updated after t_{max} actions are taken, or when a final state is reached.

Similar to AC algorithm, A3C also defines the advantage function A_t in order to reduce the variance of the estimation, which is given by

$$A(s_t,a_t;\theta,\theta_v) = R_t - V(s_t;\theta_v), \tag{19}$$

where θ and θ_v are the parameters of actor and critic network, respectively, R_t is the real reward as defined in (18), and $V(s_t;\theta_v)$ is the estimated state value. Therefore, advantage A_t can be used to enhance the agent's learning capacity so as not to overestimate or underestimate the action, and thus to improve the decision-making abilities.

On the basis of advantage function A_t, the loss function of the actor is given by

$$f_\pi(\theta) = \log\pi(a_t|s_t;\theta)(R_t - V(s_t;\theta_v)) + \beta H(\pi(s_t;\theta)), \tag{20}$$

where $H(\pi(s_t;\theta))$ is an entropy item used for encouraging exploration in training procedure and thus to avoid possible premature convergence, and β is a parameter used to control the strength of the entropy regularization and thus to facilitate the tradeoff between exploration and exploitation.

The loss function for the estimated critic network is defined as

$$f_v(\theta) = (R_t - V(s_t;\theta_v))^2, \tag{21}$$

which is used to update the value function $V(s_t;\theta_v)$. The critic update is performed on the basis of the following accumulated gradient

$$d\theta_v \leftarrow d\theta_v + \frac{\partial(R_t - V(s_t;\theta_v))^2}{\partial\theta_v'}. \tag{22}$$

The actor is updated by

$$\begin{aligned} d\theta \leftarrow d\theta &+ \nabla_{\theta'}\log\pi(a_t|s_t;\theta')(R_t - V(s_t;\theta_v)) \\ &+ \delta\nabla_{\theta'}H(\pi(s_t;\theta')). \end{aligned} \tag{23}$$

Training is conducted using standard non-centered RMSProp algorithm [36], [39], where by minimizing the two loss functions, parameters of the actor and the critic are updated

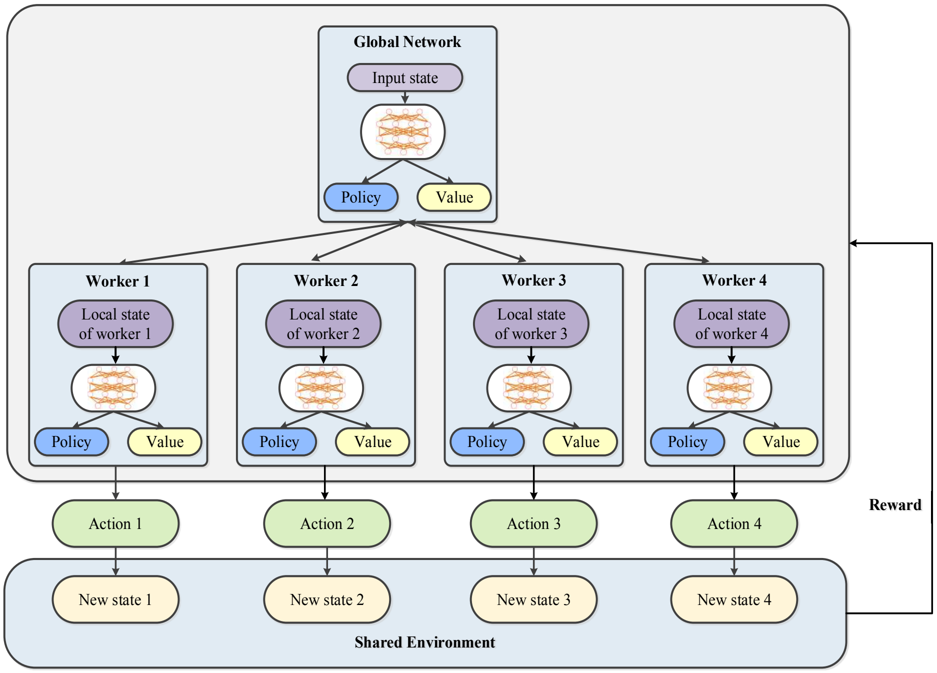


Fig. 2: Framework of A3C algorithm.

based on their accumulated gradients as in (23) and (22). The estimated gradient under RMSProp can be given by

$$g = \alpha g + (1-\alpha)\triangle\theta^2, \tag{24}$$

where α is the momentum, and $\triangle\theta$ is the accumulated gradients of the policy or value loss function.

Based on the obtained g, update is performed according to

$$\theta \leftarrow \theta - \eta\frac{\triangle\theta}{\sqrt{g+\epsilon}}, \tag{25}$$

where η is the learning rate, and ϵ is a tiny positive number used to avoid errors when denominator equals to 0 [36], [39].

The global fremework of the A3C algorithm in this paper is illustrated in Fig. 2, and our proposed A3C based joint radio and computational resource pricing and allocation algorithm is detailed in Algorithm 1 [2], [30], [36], [40].

Remark 3: *In Algorithm 1, parameters θ' and θ'_v are the parameters of each worker, and θ and θ_v are the parameters of the global actor and critic network, respectively.*

B. Implementation Details of Algorithm 1

We consider the MEC server as the central agent of A3C, a total of six threads act as six workers which interact with the MEC enabled blockchain environment concurrently. We use two deep neural networks with weights θ and θ_v to approximate the actor and the critic of each worker. In the training process, each worker calculates its own successive gradients during each episode, and the parameters θ and θ_v are optimized using gradients defined in eqs. (23) and (22), respectively. At the end of each episode, each worker updates the global network and then collects the new state of the global weights. Training is repeated at each episode until the final episode, when the algorithm should have converged, and the averaged reward is maximized.

Algorithm 1 A3C Based Joint Resource Pricing and Allocation Algorithm

```
Initialization:
Initialize the global actor network and global critic network with parameters $\theta$ and $\theta_v$.
Initialize global shared counter as $T = 0$ and thread-specific counter as $t = 1$.
Initialize the thread-specific actor and thread-specific critic network parameters $\theta'$ and $\theta'_v$.
Initialize $T_{max}$, $t_{max}$, and all the parameters as in Table I, respectively.
Iteration:
while $T < T_{max}$ do
  for each woker do
    Initialize the gradients of global agent: $d\theta = 0$, $d\theta_v = 0$.
    Synchronous parameters of each worker with global parameters $\theta' = \theta$ and $\theta'_v = \theta_v$.
    Obtain the system state $s_t$.
    for $t \leq t_{max}$ do
      Perform $a_t$ under policy $\pi(a_t|s_t;\theta')$.
      Obtain reward $r_t$ and new state $s_{t+1}$.
      $t = t + 1$.
    end for
    $R = \begin{cases} 0, & \text{for terminal state,} \\ V(s_t, \theta'_v), & \text{for non}-\text{terminal state.} \end{cases}$
    for $t = t_{max}, t \geq 1$ do
      $R = r_t + \gamma R$.
      Obtain accumulate gradient wrt $\theta'$ based on (23);
      Obtain accumulate gradient wrt $\theta'_v$ based on (22);
    end for
    Update $\theta$ and $\theta_v$ according to (25).
    $T = T + 1$.
  end for
end while
```

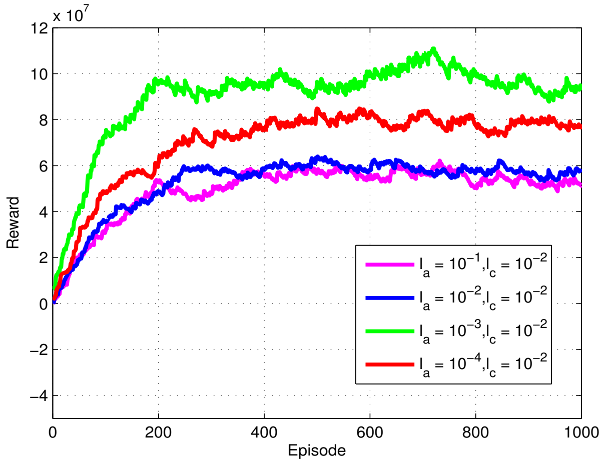


Fig. 3: Total obtained reward of all the miners under different learning rate of the actor network.

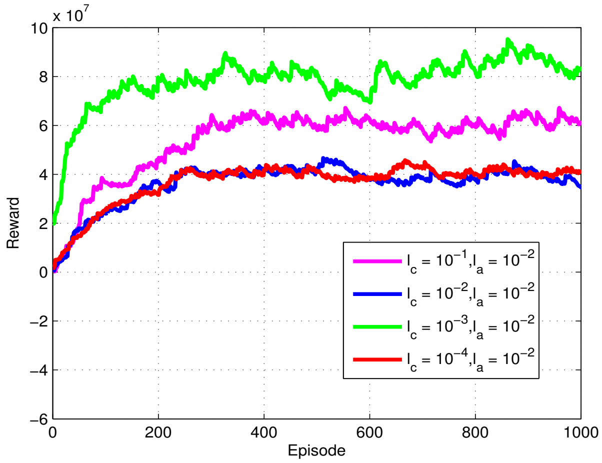


Fig. 4: Total obtained reward of all the miners under different learning rate of the critic network.

TABLE I: Simulation Parameter Settings

Parameter	Value
Number of miners, N	10
Miners' local processing capability, f_n^{loc}	(1,4) G CPU cycles/s
Input data size of mining task, D	(30,50) Mbit
Computation amount of mining task, C	(1,10) G CPU cycle
Processing capability of MEC server, F	100 G CPU cycles/s
Processing capability of each computation resource block, F_0	1 G CPU cycles/s
Total bandwidth for task offloading, B	50 MHz
Bandwidth of each subchannel, B_0	1 MHz
Block size, χ	8Kbit
Computation amount for block verify , φ	100 Kbit
Block propagation rate, $r_{n,m}^p$	(0.1, 2) Mbps
Prospect theory parameters, μ, ξ, and λ	(0.01, 0.99), (0.01, 0.99), (1,10)
Minimum, maximum price of computation resource, v_{min}/v_{max}	0.001×10^{-6}, 2×10^{-6} \$/(CPU cycles/s)
Minimum, maximum price of wireless resource, η_{min}/η_{max}	0.1×10^{-3}, 1×10^{-3} \$/bps
Coefficient α	4
Reward of mining, e	4×10^7

VI. Simulation Results and Discussions

In this section, we provide simulations to verify the performance of our proposed joint A3C based joint optimization algorithm.

There are $N = 10$ mobile devices with various prospect preferences act as block miners, whose local processing capacities f_n^{loc} are randomly taken from (1, 4) G CPU cycles/s. In each time slot, the average wireless SNR between miners and the MEC server take their values form the discrete set of $\gamma = \{1, 3, 7, 15, 31\}$ [41], where $\gamma = 1$ means the wireless channel for task offloading is rather bad, while $\gamma = 31$ means very good. The SNRs between miners and the MEC server may change based on certain transition probabilities at the beginning of each time slot. The total wireless bandwidth of the MEC server is $B = 30$ MHz, which is partitioned into 30 subchannels and each is with a bandwidth $B_0 = 1$ MHz, and therefore, the miners' wireless task offloading rate $r_n(t)$ will take their values from the set of $\{1, 2, 3, 4, 5\}$ Mbps when it is allocated with one wireless subchannel [41].

We assume there are 50 subchannels, each user can rent an integer number of subchannels from the set $\mathcal{B} = \{1, 2, 3, 4, 5\}$, where each element in the set will be allocated to one miner, and each miner can only select one element from the set. The computation capability of the MEC server is $F = 100$ G CPU cycles/s, which is partitioned into 100 computation resource blocks, and each is with computation resource of $F_0 = 1$ G CPU cycles/s. Similarly, each miner can only select one element form the set $\mathcal{F} = \{1G, 2G, ..., 10G\}$ [41]. The maximum episode is 1000 and the maximum steps in each episode is 100. The default learning rate of the actor and the critic are set as $\alpha_a = 0.01$ and $\alpha_c = 0.01$, respectively. The default value of other parameters are summarized in Table I.

A. Convergence of Algorithm 1

We first illustrate the convergence of our proposed algorithm under different learning rates. Fig. 3 shows the convergence under different actor's learning rate, with the critic's learning rate set as the default value $l_c = 10^{-2}$, and Fig. 4 shows the convergence under different critic's learning rate, while the actor's learning rate takes the default value $l_a = 10^{-2}$. As can be seen from the two figures, the system reward first increase sharply, and converges at nearly the 300th episode under different learning rate combinations, demonstrating our proposed algorithm converges fast.

B. Performance Evaluation of Algorithm 1

Next, we evaluate the performance of our proposed algorithm, which is termed as "Proposed A3C based algorithm" in the following, by comparing it with the following algorithms: (i) "Fixed pricing algorithm": Where the price each miner pays for both radio and computational resources are fixed as 5×10^{-4} \$/bps and 5×10^{-6} \$/(CPU cycles/s), respectively. (ii) "Uniform resource allocation algorithm": In this method, both radio and computation resources are allocated to each miner uniformly, i.e., each miner is allocated with 3 wireless subchannels and 5 computation resource blocks fixedly under the default parameter settings. (iii) "AC based algorithm": The only difference between this method and our proposed

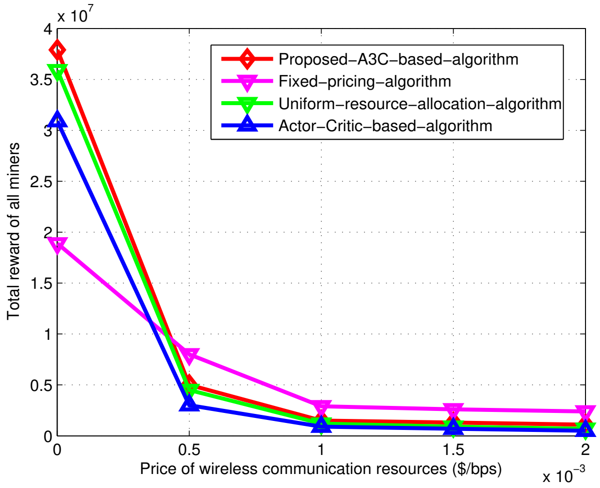


Fig. 5: Total reward of all the miners vs. the price of wireless transmit resources.

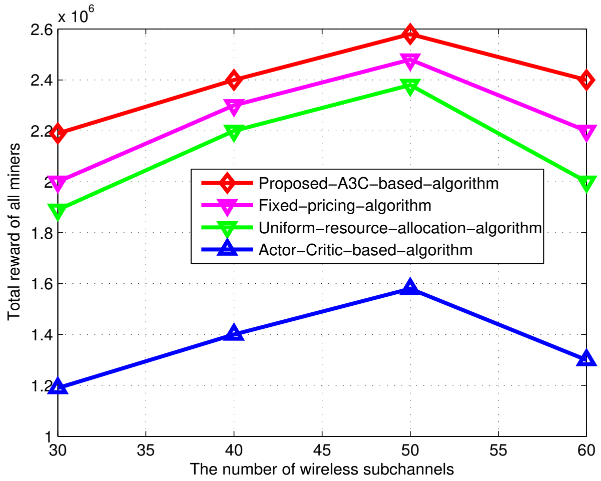


Fig. 6: Total reward of all the miners vs. the number of wireless subchannels.

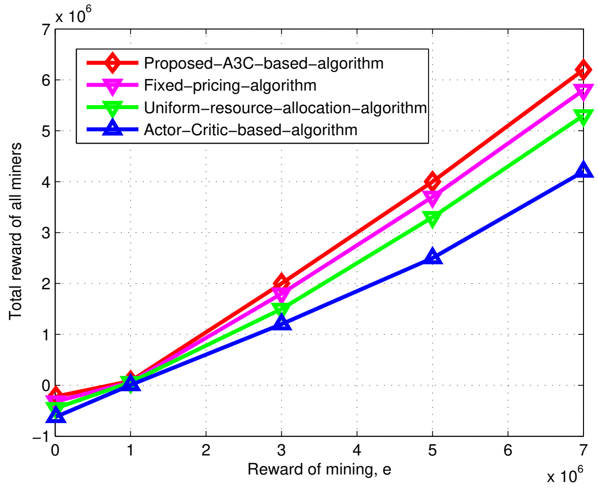


Fig. 7: Total reward of all the miners vs. the reward of mining.

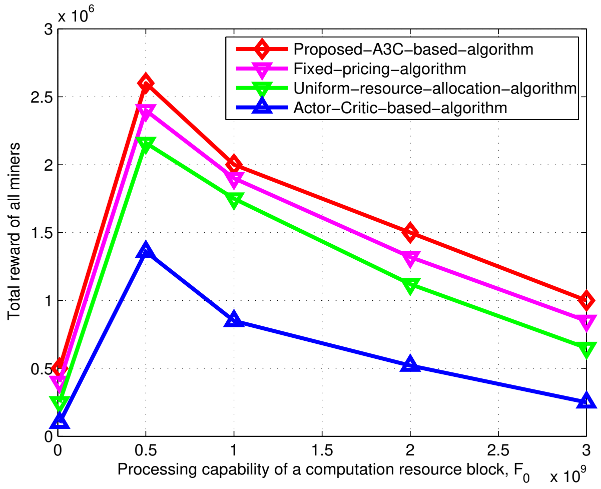


Fig. 8: Total reward of all the miners vs. the processing capability of each computation resource block of MEC server.

algorithm is that this method is based on AC algorithm, while our proposed algorithm is based on A3C framework.

In Fig. 5, we plot how the total reward changes under different price of wireless communication resources. As is shown, when the price is very low, i.e., the communication resource is very cheap, the proposed, the Uniform resource allocation, and the AC-based algorithms could obtain high rewards. Since in Fixed-pricing algorithm, the price of communication resource is set as 5×10^{-4} \$/bps fixedly, which is much higher, so the obtained reward of this method is much lower than that of other three algorithms. Since the reward of mining keeps unchanged, with the price increase, the total rewards obtained by miners first drop sharply, and then slow down gradually, for all the algorithms. It also can be seen that the performance of the proposed algorithm always performs better than Uniform resource allocation algorithm, and followed by AC based algorithm. We can also find that when the price of communication resource is greater than 5×10^{-4} \$/bps, Fixed pricing algorithm performs better than other algorithms, even than our proposed joint optimization algorithm, this is because the default price of communication resource is taken from 1-10×10^{-4} \$/bps, and that of Fixed pricing algorithm is 5×10^{-4} \$/bps.

Fig. 6 depicts the relationship between the total reward and the number of wireless subchannels, i.e., the system bandwidth for task offloading. As can be seen, with the number of subchannels increase, the total obtained reward of all the miners first increase, and then decrease. This can be explained like this, first when the number of subchannels is small, each miner could not be allocated with enough subcarriers for task offloading. When subchannels gets more, the subchannels allocated to each miners gets gradually plenty enough for task offloading, so the obtained reward increases. When subchannels becomes extra, since in the four algorithms, all the subchannels should be allocated to miners, so miners have to pay extra fees for the extra subcarriers, leading to decrease in total obtained reward. It can also be observed that our proposed algorithm always performs the optimal, followed by Fixed pricing, Uniform resource allocation, and AC based algorithms by sequence.

Fig. 7 shows how mining reward e affects the total reward of all the miners. First when the mining reward is very small, the obtained total reward of all the miners is minus for all the algorithms, i.e., miners will loss money in mining. When the mining reward increases, the total reward of all the miners grows nearly linearly, and it can be observed that our proposed A3C based joint algorithm always performs the best.

Fig. 8 presents how the total reward of all the miners changes with different processing capability F_0 of each computation resource block of the MEC server. When F_0 is small, using the computation resource of MEC server for mining task processing is of no much use, so the obtained reward is small. When F_0 grows, the obtained reward increases rapidly and reaches the maximum when F_0 grows to 0.25 G CPU cycles/s. Afterwards, the obtained total reward of miners decreases with F_0 increases, and this can be interpreted like this. When $F_0 = 0.25$ G, miners could obtain the maximal reward, this means $F_0 = 0.25$ G is strong enough to process the mining task together with the local processing resources, and much stronger F_0 is not necessary. Since all the computation resources will be allocated to miners, miners have to pay for the extra processing resources when $F_0 > 0.25$ G, leading to a reduction in the total obtained rewards. Also, it can be known that our proposed A3C based algorithm always performs the optimum among all the schemes.

VII. Conclusions

In this paper, we have investigated the maximization of miners' long-term averaged utility in a mobile edge computing enabled blockchain system, by jointly optimizing the communication and computation resource pricing and allocation optimization, in conjunction with a prospect perspective to strike a balance between risks and uncertain rewards. Based on A3C deep reinforcement learning algorithm, we developed a low-complexity algorithm to solve the joint optimization problem. Simulation results have verified the convergence of our algorithm, and have demonstrated that our algorithm performs better than the existing algorithms in terms of the miners' total reward.

References

[1] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, and e. a. D. Niyato, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, pp. 22 328–22 370, 2019.

[2] J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile edge computing: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, 2019.

[3] J. Feng, X. Zhao, K. Chen, F. Zhao, and G. Zhang, "Towards random-honest miners selection and multi-blocks creation: Proof-of-negotiation consensus mechanism in blockchain networks," *Future Generation Computer Systems*, vol. 105, no. 4, pp. 248–258, 2020.

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

[5] H. Sheng, Y. Zheng, W. Ke, D. Yu, X. Cheng, W. Lv, and Z. Xiong, "Mining hard samples globally and efficiently for person re-identification," *IEEE Internet of Things Journal*, accepted, DOI:10.1109/JIOT.2020.2980549, 2020.

[6] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1079–1092, Feb. 2019.

[7] X. Li, X. Wang, P.-J. Wang, Z. Han, and V. C. M. Leung, "Hierarchical edge caching in device-to-device aided mobile networks: Modeling, optimization, and design," *IEEE JSAC*, vol. 36, no. 8, pp. 1768–1785, Aug. 2018.

[8] H. Cao, S. Wu, G. S. Aujla, Q. Wang, L. Yang, and H. Zhu, "Dynamic embedding and quality of service-driven adjustment for cloud networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1406–1416, Feb. 2020.

[9] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Transactions on Wireless Communications*, vol. 33, no. 5, pp. 156–165, 2019.

[10] A. Jindal, G. S. Aujla, and N. Kumar, "Survivor: A blockchain based edge-as-a-service framework for secure energy trading in sdn-enabled vehicle-to-grid environment," *Computer Networks*, vol. 153, pp. 36–48, 2019.

[11] G. S. Aujla and A. Jindal, "A decoupled blockchain approach for edge-envisioned iot-based healthcare monitoring," *IEEE Journal on Selected Areas in Communications*, 2020.

[12] S. Shen, Y. Han, X. Wang, and Y. Wang, "Computation offloading with multi-agent in edge computing-supported iot," *ACM Transactions on Sensor Networks*, vol. 7, no. 1, 2019.

[13] M. Chen, U. Challita, W. Saad, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.

[14] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *arXiv:1909.07972*, 2019.

[15] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 177–191, Jan. 2020.

[16] X. Wang, C. W. X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching," *IEEE IoT Journal*, accepted, DOI:10.1109/JIOT.2020.2986803, April 2020.

[17] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[18] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L. C. Wang, "Deep reinforcement learning for mobile 5g and beyond: Fundamentals, applications, and challenges," *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 44–52, Apr. 2019.

[19] X. Yuan, H. Tian, H. Wang, H. Su, J. Liu, and A. Taherkordi, "Edge-enabled wbans for efficient qos provisioning healthcare monitoring: A two-stage potential game-based computation offloading strategy," *IEEE Access*, Accepted, doi: 10.1109/ACCESS.2020.2992639, 2020.

[20] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Networks and Applications*, 2020, doi:10.1007/s11036-020-01624-1.

[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*, accepted, 2020.

[22] M. Liu, F. R. Yu, Y. Teng, V. C. 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. 11 008–11 021, 2018.

[23] 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 *IEEE International Conference on Communications (ICC)*. Kansas City, USA, May. 2018, pp. 1–6.

[24] 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 of Things Journal*, vol. 6, no. 3, pp. 4585–4600, 2019.

[25] K. Zhang, J. Cao, S. Leng, C. Shao, and Y. Zhang, "Mining task offloading in mobile edge computing empowered blockchain," in *IEEE International Conference on Smart Internet of Things (SmartIoT)*. Tianjin, China, 2019, pp. 234–239.

[26] F. Guo, F. R. Yu, H. Zhang, H. Ji, M. Liu, and V. C. M. Leung, "Adaptive resource allocation in future wireless networks with blockchain and mo-

bile edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1689–1703, 2020.

[27] X. Qiu, L. Liu, W. C. Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8050–8062, 2019.

[28] C. Qiu, H. Yao, C. Jiang, S. Guo, and F. Xu, "Cloud computing assisted blockchain-enabled internet of things," *IEEE Transactions on Cloud Computing*, accepted, 2019.

[29] Y. He, F. R. Yu, N. Zhao, V. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[30] 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. accepted, Dec. 2020.

[31] Y. Wang, Y. Zhang, M. Sheng, and K. Guo, "On the interaction of video caching and retrieving in multi-server mobile-edge computing systems," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1444–1447, 2019.

[32] J. Du, G. Lu, J. Jiang, D. Zhai, F. R. Yu, and Z. Ding, "When mobile edge computing (mec) meets non-orthogonal multiple access (noma) for the internet of things (iot): System design and optimization," *IEEE Internet of Things Journal*, 2021, online.

[33] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, accepted, DOI: 10.1109/COMST.2020.2970550, 2020.

[34] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[35] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actorccritic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[36] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the 33nd International Conference on Machine Learning 2016*. New York, USA, June 19-24 2016, pp. 1928–1937.

[37] L. Liu, J. Feng, Q. Pei, C. Chen, Y. Ming, B. Shang, and M. Dong, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actorccritic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2021.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, and etc., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[39] T. Tieleman and G. Hinton, *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*. COURSERA: Neural Networks for Machine Learning, 2012, ch. 4, p. 2.

[40] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

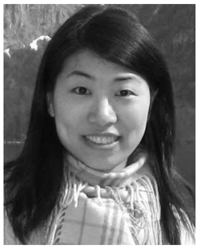
[41] Y. Wei, F. R. Yu, M. Sun, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled iot using natural actor-critic deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019.

Jianbo Du received the Ph.D. in communication and information systems from Xidian University, Xi'an, Shaanxi, China, in 2018. She was a visiting scholar of Carleton university, Canada, in 2019. She is now a lecturer with the department of Communication and Information Engineering, Xi'an University of Posts and Telecommunications. Her research interests include mobile edge computing, NOMA, artificial intelligence, resource allocation, convex optimization, and blockchain, and their applications in wireless communications.

Wenjie Cheng received the B.S. degree in communication engineering from Xi'an University of Posts and Telecommunications in 2020. He is now working towards the M.S. degree in communication and information systems at Xi'an University of Posts and Telecommunications. His research interests include mobile edge computing, resource managemen-t, NOMA, and blockchain, and their applications in wireless communications.

Guangyue Lu received the Ph.D. degree from Xidian University, Xian, China, in 1999. From September 2004 to August 2006, he was a Guest Researcher with the Signal and Systems Group, Uppsala University, Uppsala, Sweden. Since 2005, he has been a Professor with the Department of Telecommunications Engineering, Xian Institute of Posts and Telecommunications, Xian. His current research area is in signal processing in communication systems, cognitive radio, spectrum sensing. Due to his excellent contributions in education and research, he received the Award from the Program for New Century Excellent Talents in University, Ministry of Education, China, in 2009.

Haotong Cao (S'17-M'20) received the B.S. Degree in Communication Engineering from Nanjing University of Posts and Telecommunications (NJUPT) in 2015. He received the Ph.D. Degree form N-JUPT, China, in 2020. He was a visiting scholar of Loughborough University, U.K. in 2017. He is currently the PostDoc of The Hong Kong Polytechnic University, SAR, China. He has served as the TPC member of multiple IEEE conferences, such as IEEE INFOCOM, IEEE ICC, IEEE Globecom. He has published multiple academic research papers since 2016. His research interests include wireless communication theory, 5G and B5G, resource allocation in wired and wireless networks.



Xiaoli Chu (M'06-SM'15) received the B.Eng. degree in electronic and information engineering from Xi'an Jiao Tong University, Xian, China, in 2001, and the Ph.D. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2005. She is a Senior Lecturer with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K. From September 2005 to April 2012, she was with the Centre for Telecommunications Research, Kings College London. She has published more than 100 peer-reviewed journal and conference papers. She is the Lead Editor/author of the book Heterogeneous Cellular Networks: Theory, Simulation and Deployment (Cambridge University Press, 2013) and the book 4G Femtocells: Resource Allocation and Interference Management (Springer 2013).

Zhicai Zhang (Member, IEEE) received the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He is currently an Assistant Professor with the School of Physics and Electronic Engineering, Shanxi University, Taiyuan, China. From September 2017 to September 2018, he was with Carleton University, Ottawa, ON, Canada, as a Visiting Scholar. He is a member of the Shanxi Institute of Communications. His research interests include vehicular networks, mobileedge computing, machine learning, and blockchain. Dr. Zhang serves/served as a reviewer for several journals, including IEEE Communications Magazine, Wireless Networks, and the International Journal of Machine Learning and Cybernetics.

Junxuan Wang received the B.S. degree from Northwestern Polytechnical University, China, in 1994, the M.E. degree from the Xi'an University of Science and Technology, China, in 2002, and Ph.D. degree from the Beijing university of Post and Telecommunications, China, in 2005, respectively. He is currently a Full Professor with the School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi, China. His research interests include the areas of 5G networks and wireless communications.