Cloud Mining Pool Aided Blockchain-Enabled Internet of Things: An Evolutionary Game Approach

Tianle Mai, *Student Member, IEEE*, Haipeng Yao, *Senior Member, IEEE*, Ni Zhang, Lexi Xu, *Member, IEEE*, Mohsen Guizani, *Fellow, IEEE*, and Song Guo, *Fellow, IEEE*

Abstract—The past few years have witnessed an exponential growth of diverse Internet of Things (IoT) devices as well as compelling applications ranging from industrial production to medical care. Dramatic advances in IoT technology not only brought enormous economic opportunities but also challenges (e.g., privacy and security vulnerabilities). Recently, with the appearance of blockchain technology, the integration of IoT and blockchain (BCoT) is considered a promising solution to address these issues. Blockchain provides a secure and scalable data management framework for IoT devices. However, the huge computation and energy cost of the consensus process in blockchain prevents it from being directly applied as a generic platform. To overcome this challenge, in this article, we propose a cloud mining pool-aided BCoT architecture, where the IoT devices can rent the computing resources from the cloud mining pools to offload the mining process. Based on this architecture, we study the mining pool selection problem and analyze the colony behaviors of IoT devices with different pooling strategies. We propose a centralized evolutionary game-based pool selection algorithm for the sake of maximizing the system utility. Considering the non-cooperative relationship among multiple miners, we also propose a lightweight distributed reinforcement learning algorithm, named the 'WoLF-PHC' algorithm.

Index Terms—Evolutionary game, mining pool, cloud mining pool, WoLF-PHC

1 Introduction

In the past decade, the Internet of things (IoT) has attracted a large amount of attention from both academia and industry [1]. The IoT refers to the billions of physical devices that are now connected to and transfer data through the Internet without requiring human-to-human or human-to-computer interaction. These connected IoT devices are slowly entering every aspect of our lives ranging from healthcare to industrial manufacture. According to Gartner's prediction, it is expected more than 25 billion IoT connections in the future year 2025. However, with the large-scale IoT deployments, IoT applications are facing challenges in the aspect of scalability, privacy, and security [2]. The current IoT system adopts a centralized management platform to authenticate, authorize and connect a massive of heterogeneous IoT devices, which will turn into a bottleneck. Besides, unsecured IoT devices provide an easy target for distributed-denial-of-service (DDoS) attacks, malicious attackers, and data breaches.

In recent years, another breakthrough technology, blockchain, offers significant opportunities to address these challenges [3]. The blockchain is a distributed digital ledger of transactions that is maintained by a community of participants without the intervention of a trusted third party [4]. Within a blockchain community, any new transactions or events must be validated upon the agreement among the majority of the participants through a consensus process (e.g., proof of work (PoW), proof of stake (PoS)) before they are attached to the chain [5]. Such a process creates tamper-resistant records of shared transactions and events among the involved parties. Therefore, no single organization has control over the data generated by IoT devices in blockchain, thereby protecting the privacy of data and enhancing scalability. Moreover, blockchain adds a layer of security in terms of encryption, the removal of a single point of failure, and the ability to quickly identify the weak point in the network [6]. Recently, a large number of applications combining blockchain and the IoT can be seen [2]. For example, Deloitte uses blockchain and IoT technology in supply chain traceability.

While blockchain provides a secure and scalable data management framework, there still exist challenges to be addressed before it can serve as a generic platform for IoT. As discussed above, the consensus process (e.g., proof of work (PoW)) in the blockchain is particularly computationally intensive and energy-consuming. The participants, termed as miners, have to constantly try to solve a

- *Tianle Mai and Haipeng Yao are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: machealmai@gmail.com, yaohaipeng@bupt.edu.cn.*
- *Ni Zhang is with the Sixth Research Institute, China Electronic Corporation, Beijing 100141, China. E-mail: Zhangni@ncse.com.cn.*
- *Lexi Xu is with the Research Institute, China United Network Communications Corporation, Beijing 100140, China. E-mail: xulx29@chinaunicom.cn.*
- *Mohsen Guizani is with the College of Engineering, Qatar University, Doha, Qatar. E-mail: mguizani@gmail.com.*
- *Song Guo is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. E-mail: cssongguo@comp.polyu.edu.hk.*

Manuscript received 20 Jan. 2021; revised 26 June 2021; accepted 26 Aug. 2021. Date of publication 8 Sept. 2021; date of current version 8 Mar. 2023.
(Corresponding Author: Haipeng Yao.)
Recommended for acceptance by D. Mohaisen.
Digital Object Identifier no. 10.1109/TCC.2021.3110965

cryptographic puzzle in the form of the hash computation. Considering that the majority of IoT devices are too limited in terms of computing, storage, and energy resources, this computationally intensive process hinders the integration of IoT and blockchain. While some energy-efficient consensus algorithms (e.g., proof of stake (PoS), practical byzantine fault tolerance (PBFT)) are developed, the computation and energy overhead are still inevitable.

To address these challenges, the cloud mining mechanism becomes a viable option. Cloud computing can empower resource-constrained IoT devices with extra sufficient storage and computing resource. In this way, more IoT devices are enabled to participate in the blockchain network, so as to increase the whole system utility. Recently, a large and growing body of literature has investigated cloud mining-based BCoT architecture [6], [7], [8]. These works are mainly focusing on the resource allocation between the devices and cloud servers. For example, in [9], Xiong *et al.* formulated the resource allocation problem between cloud services and IoT devices as a Stackelberg game and implemented a backward induction algorithm to search the Nash equilibria of this game.

However, with the exponential growth of the number of IoT devices and hash rate, the probability for a single miner to win the mining competition game tend to be slim. Only a few fortunate miners would obtain large rewards and the majority will get no rewards. To seek a steady reward stream, miners are gradually willing to group into several teams, called mining pools. In the mining pool, miners will share the rewards according to their contributed hash power (i.e., computing resource).

Therefore, in this paper, we design a cloud mining pool-aided BCoT architecture, where IoT devices can rent the computing and storage resources from the cloud mining pool. In terms of this architecture, we discuss the mining pool selection problem among IoT devices. Assuming that the IoT devices are rational (i.e. profit-driven), we model the dynamic mining-pool selection process as an evolutionary game. To search the evolutionary stable strategy (ESS), we design a centralized evolutionary game-based pool selection algorithm, where a centralized controller is used to synchronize information. Besides, considering the non-cooperative relationship among miners, we propose a distributed reinforcement learning algorithm, termed as the 'WoLF-PHC' algorithm.

The major contributions of this paper can be summarized as follows.

- We design a cloud mining pool aided BCoT architecture, where the IoT devices can offload their computing or storage tasks to the cloud servers.
- We discuss the pool selection problem in our system. First, we propose a centralized evolutionary game-based algorithm. A centralized controller is used to synchronize information.
- Moreover, considering the non-cooperation relationship among miners, we introduce a distributed reinforcement learning algorithm, termed as the 'WoLF-PHC', to search for the evolutionary stable strategy. Some extensive simulations are presented to evaluate the convergence of our algorithm and compare it to the other state-of-the-art schemes.

The rest of this paper is organized as follows. In Section 2, we review the related works. In Section 3.2, we present a cloud mining pool aided BCoT architecture and discuss the system model. We present a centralized population evolution algorithm in Section 4, and a distributed reinforcement learning algorithm in Section 5. In Section 6, we present the simulation results, and the conclusion is in Section 8.

2 Related Work

Recently, a considerable amount of literature has been published on BCoT. In [2], Dai *et al.* presented a comprehensive survey on BCoT and discussed the research challenges of this new paradigm. In the following, we will briefly discuss related works from the perspective of cloud mining and the mining pool.

2.1 Cluod Mining

Much of the current literature on BCoT pays particular attention to the cloud mining mechanism. In [6], Qiu *et al.* proposed a cloud mining assisted blockchain-enabled IoT architecture, where the cloud miner act as a mining proxy for physical IoT in cloud services to offload mining tasks. Then, the authors modeled the computing, networking resources allocation as a joint optimization problem, and proposed a dueling deep reinforcement learning to search for the optimal solution. In [10], Yao *et al.* modeled the resource allocation among cloud services and IoT devices as a Stackelberg game and proposed a lightweight multi-agent reinforcement learning algorithm. The experimental results showed that the system can converge to the Nash equilibrium point. In [11], Li *et al.* designed a blockchain-enabled edge-cloud IoT system, and proposed a double auction scheme for computing resource trading. In [12], Liu *et al.* discussed two offloading mechanisms (including the nearby access point, a group of nearby users) in cloud mining. An alternating direction algorithm is introduced to solve the offloading problem in a distributed fashion. In addition, we notice that cloud mining also receives a large amount of attention from industries. Some mining-as-a-service company (e.g., CloudHashing, MineOnCloud) offers cloud mining contracts in exchange of a fee.

2.2 Mining Pools

Recently, a large and growing body of literature has investigated the mining pool mechanism. In [13], Schrijvers *et al.* modeled the reward allocation in Bitcoin mining pools as a non-cooperative game among the miners and the pool managers. The pool manager allocates the amount of reward for each miner according to their contributed computing resources. In [14], Fisch *et al.* designed the utility and social welfare to the pooling strategies for the sake of maximizing the utility of miners. The experimental results showed that a geometric pay pool is able to achieve the optimal steady-state income for the individual miners. In [15], Lewenberg *et al.* formulated the allocation of profit among pool participants as a coalitional game, which is found to be difficult to distribute in a stable way. Besides, considering the peer-to-peer relationship among the miners, Kim *et al.* adopted the group bargaining solution to implement an incentive payment process in [16]. In addition, Luu *et al.* used a computing power-

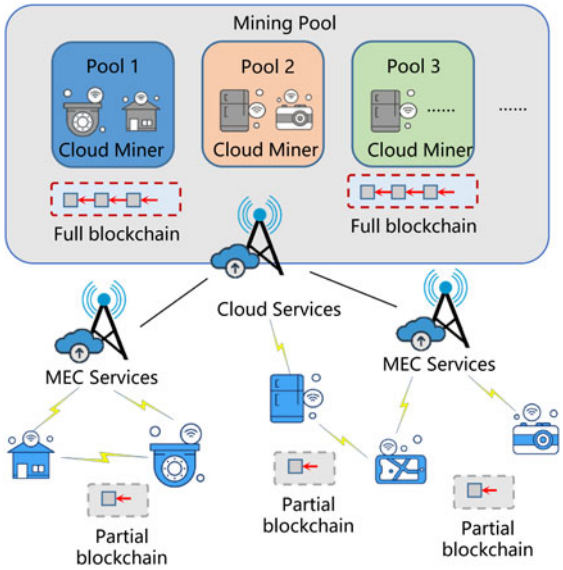


Fig. 1. Cloud mining pool aided BCoT architecture.

splitting game to enable miners' subscription to more than one mining pool in [17]. Within this mechanism, the miners split their computation resources into different pools for the sake of maximizing the expected mining reward. In [18], Liu *et al.* studied the evolutionary stability of the mining pool selection game in a blockchain network and proposed a centralized iterative algorithm to search the evolutionary stable strategy point. However, considering the non-cooperative relationship among miners, this centralized method presents poor scalability and robustness.

3 System Model

In this section, we first design a cloud mining pool-aided BCoT architecture. Then, we present the system model and problem formulation of the mining pool selection problem.

3.1 Cloud Mining Pool Aided BCoT

As discussed above, the blockchain cannot be directly applied to IoT systems. To fulfill the computing and storage resources required in the consensus process, we adopt the cloud mining paradigm in this paper. As shown in Fig. 1, there exist two types of nodes to participate in the blockchain network, including the cloud services nodes and the IoT devices nodes [2]. The cloud services are responsible for storing the entire blockchain data (the full Bitcoin blockchain data occupies about 200G large) and undertaking computational intensive operations (e.g., consensus process), while IoT devices are only responsible for undertaking some simple operations (e.g., initiating transactions). It is worth mentioning that the IoT devices still need to keep a partial blockchain local for validating the authenticity of transactions.

During the initializing phase, the IoT devices first register as a legitimate entity (i.e., cloud miner) on cloud servers, and obtain an identity ID and a public/privacy key. These cloud miners act as the proxy nodes of the IoT devices to offload their mining and storage tasks. Then, to secure a steady reward, these miners will group themselves into several mining pools. In mining process, these pools present themselves to the whole system as single powerful proxy nodes. Combing with more computing resources, the mining pools are able to gain a computation advantage over other individual miners. Note that the miner can choose to redirect its hash power to any other mining pool at any time. In each pool, the cloud provider will place a coordinator in charge of managing the miners. They'll work as a task scheduler to guarantee the miners are undertaking different subtasks so that they're not wasting hash power by trying to solve the same sub-cryptographic puzzle. Once successfully mining a block, the coordinator will divide the profit to each miner according to its devoted hash power. We will present the system model in the following.

3.2 System Model

Consider a set of IoT devices that are interested in participating in the consensus process, which is denoted as $\mathcal{N} = \{1, \ldots, N\}$. We assume that these miners are willing to form $\mathcal{M} = \{1, \ldots, M\}$ mining pools, where each mining pool adopts a different pooling strategy with different hash power requirement [18]. Let ω_j represent the hash power required by the pool $j \in \mathcal{M}$. According to consensus protocol, the probability of winning the mining game is related to the ratio between local hash power and the total hash power of the entire blockchain network. Therefore, we define a relative hash power α_j of pool j with respect to the entire hash power of all miners, which can be described as:

$$\boldsymbol{\alpha}_j(\boldsymbol{\omega}, x_j, \boldsymbol{x_{-j}}) = \frac{x_j \omega_j}{\sum_{k \in \mathcal{M}} x_k \omega_k}, \boldsymbol{\alpha}_j > 0, \tag{1}$$

where x_j represents the pool j's population fraction, and the $\boldsymbol{x_{-j}}$ represents the sum of pools' population fraction expect to pool j. Note that $\boldsymbol{\alpha}$ satisfies following condition:

$$\sum_{k \in \mathcal{M}} \boldsymbol{\alpha}_k = 1. \tag{2}$$

During the mining process, mining pools compete with each other in a race to solve the cryptographic puzzle. The appearance of solving the cryptographic puzzle can be formulated as a Poisson process with a mean random variable $\lambda = \frac{1}{T}$, where T denote the complexity of finding a block (e.g., $T = 600\ sec$ in Bitcoin). After successfully mining a block, the winner needs to propagate its solution to the entire network for reaching a consensus. Only the first block, which is confirmed by the majority of the participant, could be accepted as a new block. All other candidate blocks will be discarded, called orphaning. According to previous works [19], the propagation time of a block to reach consensus is mainly determined by the set of transaction size Q included in a block. We denote $\tau(Q) = \xi \times Q$ as the propagation time. Then, the probability of orphaning can be formulated as:

$$P^{orphan}(Q) = 1 - e^{-\lambda \tau(Q)}. \tag{3}$$

The successful probability of mining pool j to win the mining game can be formulated as:

$$P_j(\alpha_j, Q_j) = \alpha_j \times (1 - P_j^{orphan}(Q_j)) = \alpha_j \times e^{-\lambda \tau(Q_j)}. \tag{4}$$

After successfully mining a block, the winner can obtain a reward, which is composed of a fixed reward $R \geq 0$ and a

TABLE 1
Notations

Parameter	Definition
N	Number of miniers
M	Number of pools
ω_j	The hash rate required by pool j
Q_j	The set of transactions size included in a block
x_j	Population fraction of pool j
α_j	The relative computing power of pool j with respect to the all system
R	The fixed reward when mining a block
p	The price of each computing and storage resource unit
ρQ	The variable reward when mining a block
$\tau(Q)$	The time needed for a block to propagation
$P_j(\alpha_j, Q_j)$	The successful probability of mining pool j to win the mining game
$u_j(\alpha_j, Q_j)$	The expected reward of pool j
$r_i(\alpha_j, Q_j)$	The expected reward of the device i in pool j

variable reward ρQ [19]. The variable reward linearly increases with the size of the transaction Q in the block, and the ρ is the linear coefficient. Therefore, the expected reward for pool j can be expressed:

$$u_j(\alpha_j, Q_j) = (R + \rho Q_j)\alpha_j \times e^{-\lambda\tau(Q_j)}. \tag{5}$$

And the expected profit of the miner $i \in \mathcal{N}$ in pool j can be expressed as:

$$R_i(\alpha_j, Q_j) = \frac{(R + \rho Q_j)}{N x_j}\alpha_j \times e^{-\lambda\tau(Q_j)}. \tag{6}$$

Besides, since the miners rent the computation resource from the cloud servers, the miners have to pay for it [20]. We denote the price of each computing resource unit as p. Thus, the expected reward of the miner $i \in \mathcal{N}$ in pool j can be reformulated as:

$$r_i(\alpha_j, Q_j) = \frac{(R + \rho Q_j)}{N x_j}\alpha_j \times e^{-\lambda\tau(Q_j)} - p\omega_j. \tag{7}$$

As shown in Table 1, we list the notations of this paper.

4 Evolutionary Game Formulation of the Pool Selection Problem

In this section, we apply the evolutionary game to the mining pool selection problem [21]. The evolutionary game defines a framework of contests, strategies, and analytic into which colony competition can be modeled. It can capture the strategy adaptation of rational agents according to their fitness. That is, the agent can slowly adjust its strategy (i.e., evolves) based on the environment knowledge. Mathematically, for mining pool selection, the evolutionary game can be formulated as a 4-tuple $\mathcal{G} = < \mathcal{N}, x, \mathcal{M}, R>$, where

- Players: Players are the decision-makers with pre-programmed strategies in the game. In our scenario, each individual miner can be regarded as a player.
- Population: The population $x = [x_1, \ldots, x_M] \in X$ refers to the set of players in a mining pool. The population will present variation among competing players.
- Strategy: The strategy is a set of action $\mathcal{M} = \{1, \ldots, M\}$ that the player can perform. The different strategies will obtain different rewards. The strategy space in our scenario is the all available mining pools.
- Payoff: Payoff r_j reflects the player's expected outcome based on its strategy, where $r_i(\alpha_j, Q_j) = \frac{(R+\rho Q_j)}{N x_j}\alpha_j \times e^{-\lambda Q_j} - p\omega_j$. Note that the reward is not only determined by the local strategy, but also the other players' strategies.

4.1 Replicator Dynamics of Pool Selection

To express the evolutionary dynamics in the game, the replicator dynamics function is introduced. The replicator dynamics function is a non-linear game dynamic used to explain learning as well as evolution in evolutionary game [22]. The core idea of replicator dynamics is that the population will increase (decrease) if fitness is larger (smaller) than the average fitness. In our scenario, the replicator dynamics function of pool j can be described as:

$$\dot{x_j}(t) = \sigma x_j(t)(u_j(\alpha_j(t), Q_j) - \bar{u}(x(t))), \tag{8}$$

where $\dot{x_j}(t)$ is the growth rate of the pool j's population, σ is the speed parameter, and $\bar{u}(x)$ is the network average payoff, which can be formulated as:

$$\bar{u}(x(t)) = \sum_{j \in \mathcal{M}} u_j(\alpha_j, Q_j)x_j. \tag{9}$$

The replicator dynamics functions must satisfy the following condition:

$$\sum_{j \in \mathcal{M}} \dot{x_j}(t) = 0 \tag{10}$$

From the players' perspective, the miners will slowly adjust their selection strategies, if their payoff is less than the average payoff, otherwise the miners will keep their current strategies.

4.2 Evolutionary Equilibrium and Stability Analysis

As discussed above, the players constantly adjust their strategies (i.e., evolve) for the sake of a higher expected payoff. Along with the players evolves over time, the whole system will finally converge to the evolutionary stable strategy (ESS). The ESS is phenotypes that can persist in populations and cannot be invaded by any other strategies [23]. We can define that the x^* is an ESS if the following condition is satisfied:

$$\sum_{j \in M} x_j^* u_j((1-\sigma)x^* + \epsilon x') \geq \sum_{j \in M} x_j' u_j((1-\sigma)x^* + \sigma x'), \tag{11}$$

where x' is the invade state.

According to this definition, in the ESS, none of the players is willing to deviate its selection strategy (i.e., the rate of strategy adaptation is zero). By solving the replicator dynamics functions (i.e., $\dot{x_j}(t) = 0$), a set of fixed points can be obtained. According to [24], these fixed points are stable (i.e., ESS) if all eigenvalues of the Jacobian matrix have negative real parts. Then, the ESS can be defined as a set of stable fixed points, which can be described as follows.

Definition 1. *A population state x^* is an ESS, if the condition $(x-x^*)^T R(x^*)=0$ implies that:*

$$(x^*-x)^T R(x) \geq 0. \tag{12}$$

where $\forall x \in B-x^*$ is the neighborhood of X.

4.3 Two Mining Pool Study

To demonstrate the evolutionary stable strategy, in this part, we will present a two mining pools case study. We set the population fraction of two pool as $x_1=x$, and $x_2=1-x$. Then, we can obtain the Ordinary Differential Equations:

$$\dot{x_1}(t)=x_1x_2\left(\frac{\omega_1k_1-\omega_2k_2}{N(x_1\omega_1+x_2\omega_2)}-p(\omega_1-\omega_2)\right), \tag{13}$$

$$\dot{x_2}(t)=x_1x_2\left(\frac{\omega_1k_1-\omega_2k_2}{N(x_1\omega_2+x_2\omega_1)}-p(\omega_1-\omega_2)\right), \tag{14}$$

where

$$k_i=(R+\rho Q_i)\times e^{-\lambda\tau(Q_i)}. \tag{15}$$

By solving the above formulas, we can obtain the fixed points as $(x*,1-x*)$, where $x*=\frac{\omega_1k_1-\omega_2k_2}{pN(\omega_1-\omega_2)^2}-\frac{\omega_2}{\omega_1-\omega_2}$. According to the above definition, this fixed point is ESS if all eigenvalues of the Jacobian matrix have negative real parts. For this replicate dynamic system, the Jacobian matrix of the replicator dynamics is

$$J=\begin{pmatrix}\frac{\partial f(x_1)}{x_1} & \frac{\partial f(x_1)}{x_2}\\ \frac{\partial f(x_2)}{x_1} & \frac{\partial f(x_2)}{x_2}\end{pmatrix}$$

After some tedious mathematical manipulations, the rest point with $x*=\frac{\omega_1k_1-\omega_2k_2}{pN(\omega_1-\omega_2)^2}-\frac{\omega_2}{\omega_1-\omega_2}$ is an ESS if the following conditions are satisfied:

$$\begin{cases}\omega_1k_1-\omega_2k_2<0\\ \omega_2\omega_1(k_2-k_1)(\omega_2-\omega_1)>0\end{cases}.$$

4.4 Delay in Replicator Dynamics

As discussed above, the players can adjust their strategies based on the system's average fitness. However, in actual deployment, the latest fitness information may not be available to all players. They can only rely on historical information to make decisions. Therefore, in this paper, we introduce a certain period delayed τ in our system. The replicator dynamics will be reformulated as:

$$\dot{x_j}(t)=x_j(t-\tau)(u_j(\alpha_j(t-\tau),Q_j)-\bar{u}(x(t-\tau))), \tag{16}$$

which is a delay differential equation. To obtain the solution to this equation, the Runge-Kutta method can be applied. Besides, the stability of the delay differential equation has been well studied. In [25], Obando *et al*. investigated the stability of the replicator dynamics with the effect of a time delay using the Lyapunov method. The theoretical results show that the ESS strategy is stable if the time delay is small enough. The detailed analysis can be founded in [25]. We will evaluate the impact of delay in the Section 6.

4.5 Evolutionary Game-based Pool Selection Algorithm

As discussed above, we apply the evolutionary game to the mining pool-selection problem. The miners continually adjust their strategies based on the system average fitness for the sake of a higher expected payoff. Along with the players evolves over time, the whole blockchain network can converge to the ESS. Therefore, in this paper, we propose a centralized population evolutionary strategy algorithm. In this approach, a centralized controller is deployed to calculate the average fitness of all players. Then, the average fitness is issued to each player to evaluate the current strategy based on its current payoff (i.e., switch their strategies or keep them) [26]. The centralized pool-selection algorithm can be described as follows.

Algorithm 1. The Population Evolution Approach for Pools Selection

```
Each pool set ω, Q.
All devices randomly choose the pool.
repeat
  Each device compute the payoff from:
  r_i(α_j, Q_j) = ((R+ρQ_j)/(Nx_j)) α_j × e^{-λQ_j} − pω_i
  The payoff information is sent to the controller.
  The centralized controller computes average payoff and
  broadcast it to all devices.
  r(x̄) = (Σ_{i∈N} r_i(α_j,Q_j)) / N
  for i ∈ N do
    if r_i < r̄ then
      if rand() < (r̄ − r_i)/r̄ then
        Choose other pool.
      end if
    end if
  end for
until
```

5 Distributd Reinforcement Learning Approach

Thus far, the paper has argued an ideal model where a centralized controller cloud be placed to guide behaviors of all players. However, in an actual IoT network, considering the non-cooperative relationship among them, the centralized controller may not be available [27]. Each player has to adapt its pool-selection decision independently. Therefore, how could the individual devices optimize their strategy in the non-stationary system (i.e., multi-agent system) is a critical challenge. Inspired by the recent success of applying reinforcement learning algorithms in multi-agent system, we introduce a distributed reinforcement learning algorithm, named 'WoLF-PHC' [28].

5.1 The Multi-Agent System

With the decision-making shift to each miner, these independent miners constitute a multi-agent system, which can be modeled as a decentralized partially observable Markov decision process (Dec-POMDP) [29]. Formally, a Dec-POMDP can be formalized as a 5-tuple $< N,S,O_i,A_i,R>$, where N is the set of agents, S is the global states space, O_i

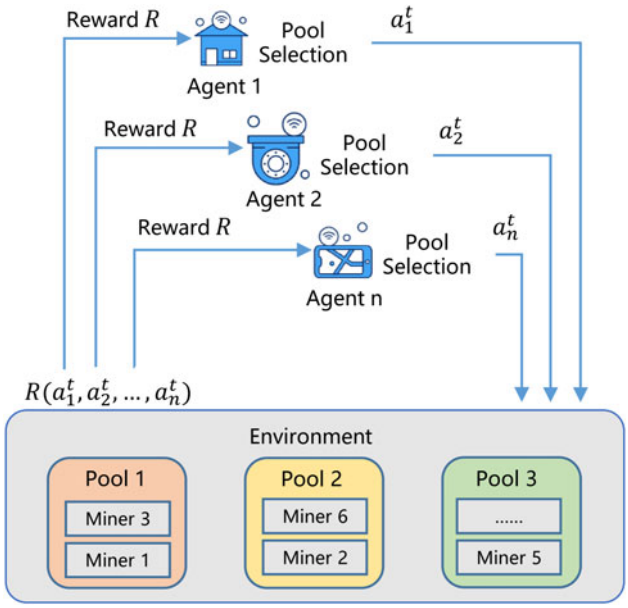


Fig. 2. The multi-agent system.

is the local observations space of agent i, A_i is the action space of agent i, and R is the immediate rewards [30]. As shown in Fig. 2, at each step, each agent takes an action a_i according to current policy $\pi_i(a_i|o_i)$ and its local observation o_i. Then, the system will generate an immediate reward R, and the state s will transit to a new state s'.

Specifically, in our scenario, the observation can be described as $O^t = [x_1^{t-1}, \ldots, x_M^{t-1}]$ (i.e., current mining pools' state, which is determined by all agents' action in the last time $t-1$), the action of each agent can be described as $A^t = [\mathcal{M}]$, where M is the set of the mining pools, and the immediate reward can be formulated as the expected profit r_i. In the following, we will define the three components of the miners:

Definition 2. *The three components of the miners:*

- *Observation:*

$$O^t = [x_1^{t-1}, \ldots, x_M^{t-1}].$$

- *Action:*

$$A^t = [\mathcal{M}].$$

- *Immediate reward:*

$$R_i^t(a_1^t, a_2^t, \ldots, a_n^t) = \frac{(R + \rho Q_j)}{N x_j} \alpha_j \times e^{-\lambda Q_j} - p\omega_j.$$

Note that the reward is only related to the joint action $(a_1^t, a_2^t, \ldots, a_n^t)$.

5.2 Policy Generation

Learning in a multi-agent system is much more difficult than in a single-agent system [31]. One of the critical challenges is the moving target problem (i.e., non-stationary learning problem), which is caused by the noise signal brought by other agents [32]. Directly applying single-agent reinforcement learning (e.g., Q-learning, Policy gradient) will suffer seriously no-convergence problem [33]. In this paper, we introduce an enhanced policy gradient algorithm, termed as the WoLF Policy Hill Climbing (WoLF-PHC). It adopts the 'wining or learning fast' scheme (i.e. learn slowly while winning or quickly while losing), where a variety of learning rates are used to encourage convergence.

In the WoLF-PHC, the updating rule of the Q value can be described as [34]:

$$Q_i(a_t) \leftarrow (1-\alpha)Q_i(a_t) + \alpha(R_i + \delta \max_{a \in A} Q_i(a_{t+1})), \tag{17}$$

where $\delta \in (0, 1]$ is the discount factor, and $\alpha \in (0, 1]$ is the learning rate. The discount factor determines the importance of future rewards and the learning rate determines what extent new knowledge overrides the old knowledge. During the training process, agents continually update their strategies, i.e. $\pi_i(a) :\rightarrow Pr(A)$, for the sake of maximizing the cumulative reward by learning from the environment.

To update the $\pi_i(a)$, the WolF-PHC adopts two learning rates θ^{win} and θ^{lose}, where $\theta^{win} > \theta^{lose}$ (i.e., learn slowly while winning or quickly while losing). They are used to update agents' policy depending upon if the agent is winning or losing [35]. To determine the winning or loss of current policy, a baseline is designed. The baseline is the expected reward of the average policy $\overline{\pi}_i(a_t)$, which can be formulated as:

$$\overline{\pi}_i(a_t) \leftarrow \overline{\pi}_i(a_t) + \frac{\pi_i(a_t) - \overline{\pi}_i(a_t)}{N_i(t)}, \ \forall a_t \in A, \tag{18}$$

where

$$N_i(t+1) \leftarrow N_i(t) + 1. \tag{19}$$

Then, the θ^{win} is applied to update the policy cautiously in the condition of win, otherwise, θ_m^{lose} is used, i.e.

$$\theta = \begin{cases} \theta^{win}, & \Pi, \\ \theta^{lose}, & o.w, \end{cases} \tag{20}$$

where Π :

$$\sum_{a \in A_i} \pi_i(a_t) Q_i(a_t) > \sum_{a \in A_i} \overline{\pi}_i(a_t) Q_i(a_t) \tag{21}$$

In the learning process, the agents constantly learn and adapt their policy toward maximizing the expected reward, followed by the decrease of the other actions [36]. The update of the pool selection policy of the mining pool can be formulated as:

$$\pi_i(a_t) \leftarrow \pi_i(a_t) + \triangle_{a_t}, \ \forall a \in A, \tag{22}$$

where

$$\triangle_{a_t} = \begin{cases} -\min\left(\pi_i(a_t), \frac{\theta_i}{M-1}\right), \Pi', \\ \Sigma_{a' \neq a} \min\left(\pi_i(a'_t), \frac{\theta_i}{M-1}\right), \ o.w. \end{cases} \tag{23}$$

where

$$\Pi' : a_t \neq \arg\max_{a'_t \in A} Q_i(a'_t), \tag{24}$$

and M is a constant coefficient.

TABLE 2
List of Parameters Setting

Parameter	Value
Fixed reward R	1000
Number of IoT devices N	5000
Number of agent	30
The price of computation resource unite p	0.01
The variable reward parameter ρ	0.01
The maximum episode numbers	10000
The learning rate α	0.2
The discount factor β	0.8
The learning rates (win) θ^{win}	0.0025
The learning rates (lose) θ^{lose}	0.01

Based on this, the WoLF-PHC algorithm based pool selection policy is described in Algorithm 2.

Algorithm 2. The WoLF-PHC algorithm for the Pool Selection

Set $\alpha, \delta, \theta^{win}, \theta^{lose}$
Initialization
repeat
  for $t = 1, 2, 3$ **do**
    Select action a_t according to current policy π_i
    Each miner observes the immediate reward R
    Update $Q_i(a_t)$ by:
    $Q_i(a_t) \leftarrow (1-\alpha)Q_i(a_t) + \alpha(R_i + \delta \max_{a \in A} Q_i(a_{t+1}))$
    Update $\overline{\pi}_i(a)$ and $\pi_i(a)$ by:
    $\overline{\pi}_i(a_t) \leftarrow \overline{\pi}_i(a_t) + \frac{\pi_i(a_t) - \overline{\pi}_i(a_t)}{N_i(t)}, \ \forall a_t \in A$
    $\pi_i(a_t) \leftarrow \pi_i(a_t) + \triangle_{a_t}, \ \forall a_t \in A,$
  end for
until

6 Performance Evaluation

In this section, we first analyze the colony behaviors of IoT devices in the pool selection problem. Then, we present the experimental results to evaluate the performance of our proposed WoLF-PHC based algorithms.

6.1 Evolution Analysis

In our experiments, we simulate a blockchain network with 5000 IoT devices (i.e., $N = 5000$). These resources constrained IoT devices are willing to rent the computing resource from the cloud services and evolve to form several mining pools. For the blockchain, we set the fixed reward R as 1000, and the variable reward parameter ρ as 0.01. For the cloud server, we set the price p of computing and storage resources unites as 0.01. The parameters setting can be found in Table 2.

We first investigate the dynamic behavior of the players' population. In this case, we deploy two mining pools, where the hash power requirement of two pools is $\omega_1 = 10$ and $\omega_2 = 30$, and the size of transactions size of both two pools is 100. As shown in Fig. 3, we plot the phase plane of the replicator in our system. The figure shows that the direction of the adaptation in mining pool selection to the ESS point (i.e. $Population_1 : 0.3, Population_2 : 0.7$). For example, when the initial population state is $x = [0.5, 0.5]$, the trajectory of replicator dynamics follows the arrows to reach the ESS.

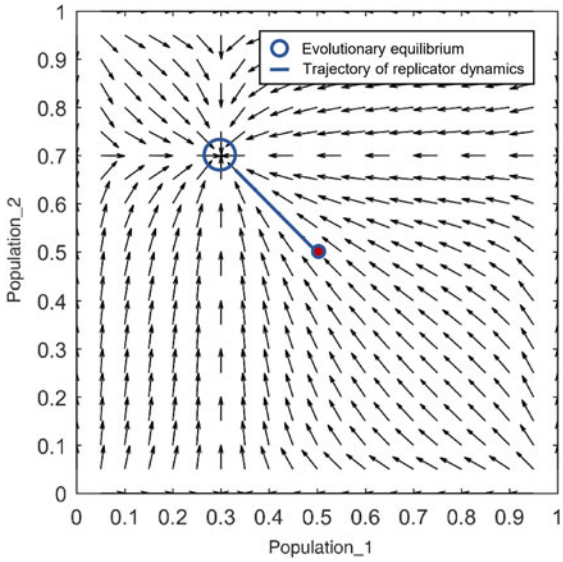


Fig. 3. The phase plane of replicator dynamics.

6.2 Evolution Analysis With Different Pooling Strategies

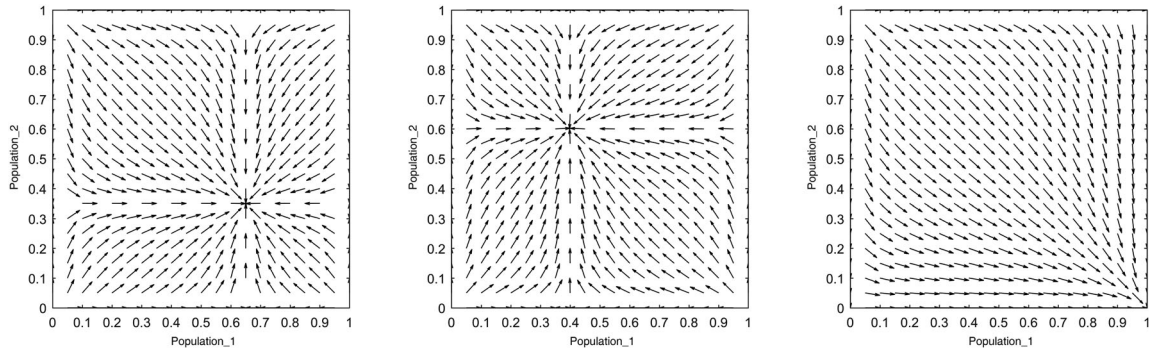
Then, we evaluate the evolution behavior of the players population with different pooling strategies. In this case, we design three groups of experiments, where the mining pools' configuration and the ESS point can be found in Table 3. As shown in Fig. 4, we notice that the miner is more willing to join in the pool with less hash power requirement. The increasing hash requirement will reduce the number of players who are willing to joining in. This is mainly caused by the cost of renting the cloud-computing resource. The slim profit gained from mining a block can not meet the exorbitant cost of resource renting. Therefore, to attract more miners to join in, the pool's coordinator should lower the threshold of the hash requirement for each miner.

6.3 Evolutionary Game-Based Pool Selection Algorithm

In this section, we will evaluate the convergence of the centralized evolutionary game-based pool selection algorithm. The trajectories of players' strategies adaptation over time are shown in Fig. 6. We can find that the players with our algorithm can quickly converge to the ESS point. This is mainly because that the centralized controller can constantly revise the player behavior. The average payoff $\bar{r}$ is issued to each IoT device to evaluate its current strategy. Then, these players can adjust their strategies based on their current payoff (i.e., switch their strategies or keep them).

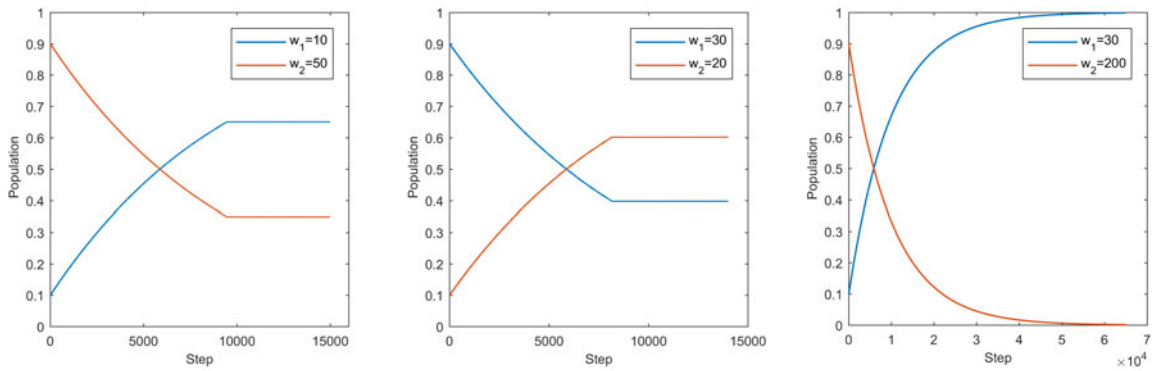
TABLE 3
Experiments Configuration

Mining Pool	Setting	ESS Point
Pool 1	$\omega_1 = 10, \omega_2 = 30$	(0.3, 0.7)
Pool 2	$\omega_1 = 10, \omega_2 = 50$	(0.35, 0.65)
Pool 3	$\omega_1 = 30, \omega_2 = 20$	(0.4, 0.6)
Pool 4	$\omega_1 = 30, \omega_2 = 200$	(1, 0)



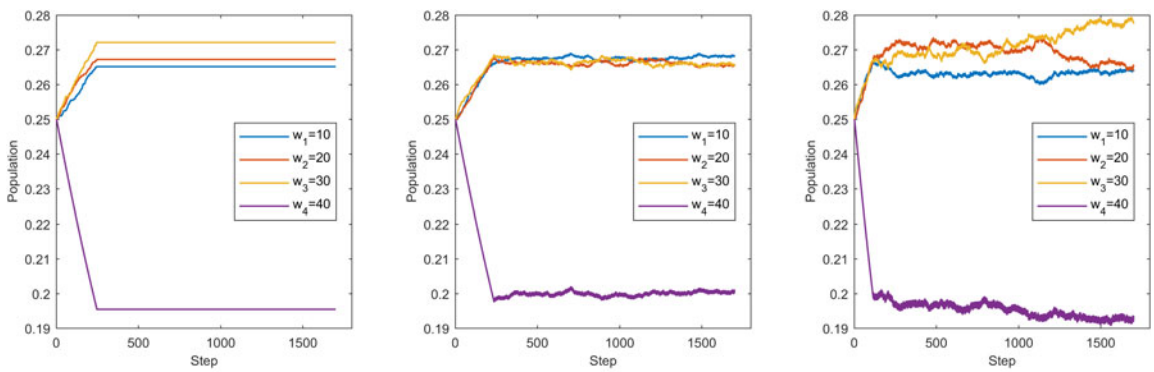
(a) $\omega_1 = 10, \omega_2 = 50$ (b) $\omega_1 = 30, \omega_2 = 20$ (c) $\omega_1 = 30, \omega_2 = 200$

Fig. 4. The phase plane of replicator dynamics with different pooling strategies.



(a) $\omega_1 = 10, \omega_2 = 50$ (b) $\omega_1 = 30, \omega_2 = 20$ (c) $\omega_1 = 30, \omega_2 = 200$

Fig. 5. Convergence analysis of the evolutionary game-based pool selection algorithm.



(a) delay = 0 (b) delay = 10 (c) delay = 20

Fig. 6. The impact of delay in strategy adaptation.

6.4 Impact of Delay in Strategy Adaptation

pt?>As discussed above, considering the communication latency between miners and the centralized controller, we investigate a certain period of time delay τ in our system. In this section, we evaluate the impact of delay in the process of strategy adaptation. We set four mining pools in our system with the different hash power requirement, where $\omega_1 = 10$, $\omega_2 = 20$, $\omega_3 = 30$ and $\omega_4 = 40$. Also, we set three groups of experiments, where the time delay τ are separately set as 0, 10, and 20. Note that the units of τ are steps in our experiments.

As shown in Fig. 6, when the time delay τ is 0, the trajectory of strategy adaptation is relatively smooth. The system can quickly converge to the evolutionary equilibrium. And when the delay is introduced, we notice fluctuating dynamics of strategy adaptation over time toward the ESS. Especially, with the time delay of τ becoming larger, the more fluctuating will be brought. This is because that when outdated knowledge is used by the players, the decisions tend to be inaccurate. But although the trajectory of strategy adaptation is fluctuating, the system can also converge to the near ESS, which means the system still be stable if the time delay is not very large [37].

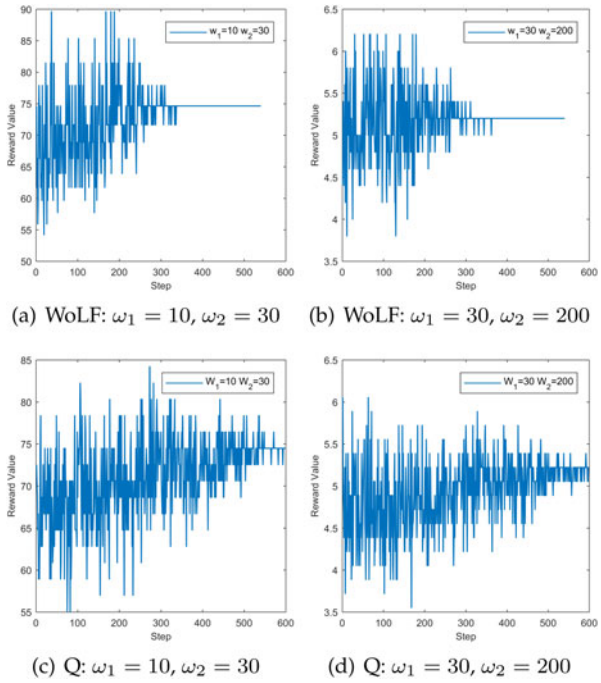


Fig. 7. Convergence analysis of Q-learning and WoLF-PHC algorithm.

7 WoLF-PHC based Pool Selection

Then, in this section, we evaluate the performance of the WoLF-PHC based algorithm in the mining pool selection problem. We construct an trading environment with 30 IoT devices (i.e., agents) and two mining pools. The blockchain environment setting is consistent with previous experiments. We set the maximum training episode numbers as 10000, the learning rate α as 0.2, the discount factor β as 0.8, the θ^{win} as 0.0025 and θ^{lose} as 0.01.

7.1 Performance Convergence of WoLF-PHC

First, we evaluate the convergence of our algorithm. We use Q-Learning as the baseline algorithm, where the learning rate α and the discount factor β are also set as 0.2 and 0.8. 17. As shown in Fig. 7, the learning process of WoLF-PHC and Q-learning algorithm are demonstrated. We notice that the agents with the Q-learning algorithm exhibit a poor convergence performance. This is caused by the moving target problem in multi-agent system. By contrast, benefiting from the 'wining or learning fast' scheme, the WoLF-PHC algorithm present a much better convergence performance.

TABLE 4
Convergence Performance

Algorithm	Steps	Conv Point
WoLF-PHC	320	(0.3, 0.7)
Q-Learning	860	(0.33, 0.67)
Policy Gradient	1020	(0.3, 0.7)
DQN	2470	(0.27, 0.73)
DDPG	1930	(0.3, 0.7)

Besides, as shown in Fig. 8, we present the trajectories of agents' strategies adaptation. In this case, we design three groups of experiments, where the mining pools' configuration can be found in Table 3. We notice that while the adaptation process exists fluctuation, after about 400 steps, the system will converge. By comparing to the phase plane of the replicator dynamics, these convergence points are the ESS of the system.

Moreover, we evaluate our proposed algorithm in comparison with other state-of-the-art reinforcement learning algorithms, including Policy Gradient (PG), Deep Q-learning (DQN), and Deep Deterministic Policy Gradient (DDPG). In this experiment, we set the mining pool's hash power requirement as $\omega_1 = 10$ and $\omega_2 = 30$, and the value of learning rate α and the discount factor β as 0.2 and 0.8. The other algorithm parameters setting of DQN and DDPG can be found in [38]. As shown in Table 4, we notice that all algorithms can converge to a small neighborhood of the ESS point $(0.3, 0.7)$. This demonstrates that reinforcement algorithms can adapt to the non-stationary system and converge to the system's ESS point. But different algorithms present different rates of convergence. The WolF-PHC algorithms exhibit the best performance. It can converge in around 320 steps. In contrast, the convergence of DQN and DDPG are the worst. This is because that they have too many parameters that need to be updated during the learning process. Therefore, we can draw the conclusion that while the complex neural network design enables them to solve complex tasks, simple learning algorithms may more efficient for simple tasks.

7.2 Reward versus Pooling Strategies

In the following, we evaluate the impact of the pooling strategies to the agent's reward. In this case, we fixed the pool1's hash power requirement as $\omega_1 = 10$, and the set of transactions

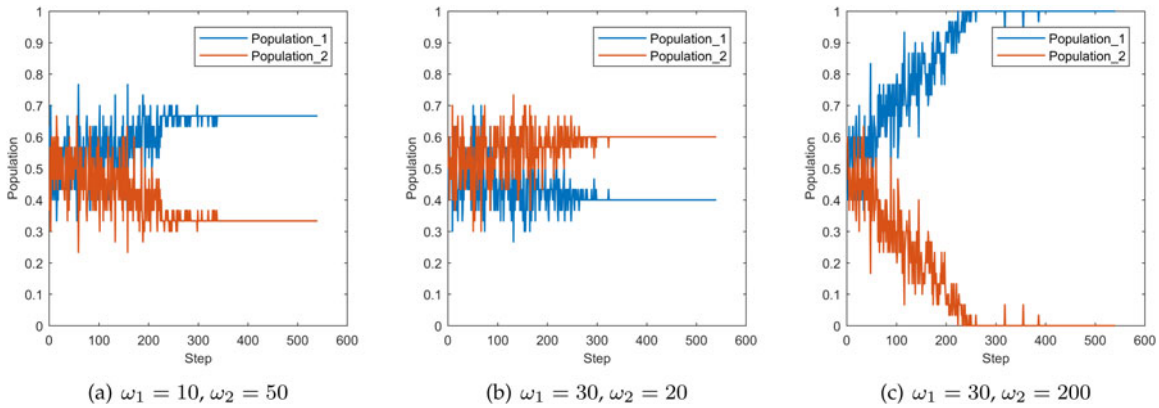


Fig. 8. Convergence analysis of Wolf-PHC based pool selection algorithm.

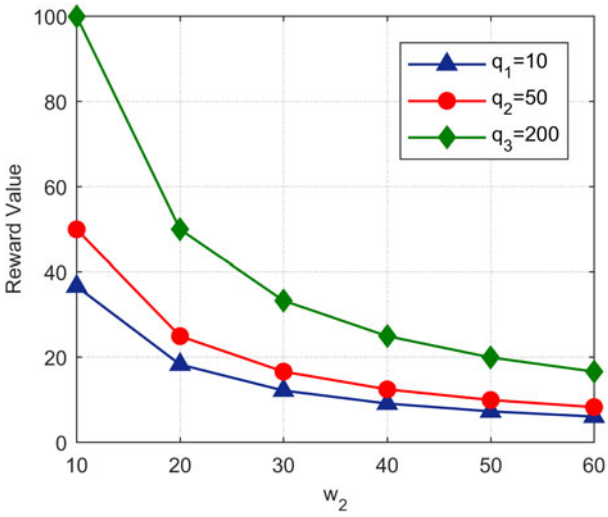


Fig. 9. Reward versus pooling strategies.

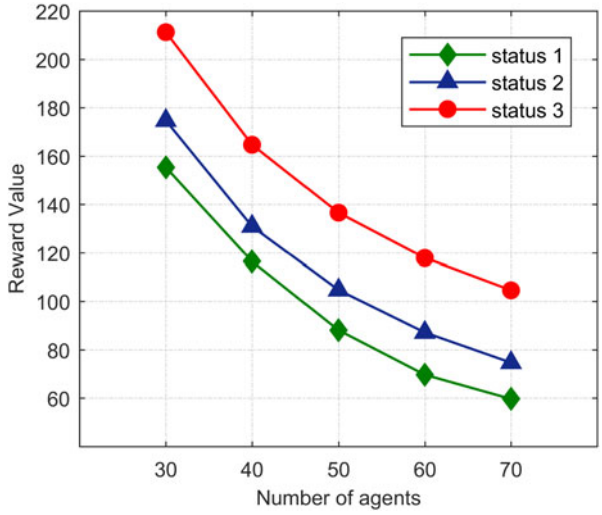


Fig. 10. Reward versus number of miners.

size as $q = 100$. As shown in Fig. 9, as the hash power requirement of pool2 ω_2 increasing, the total reward reduces. This is caused by the cost of renting the cloud computing resource. Because the total reward gained from the blockchain network remains unchanged, renting more computing resources will reduce the total profit of the miners. Besides, we evaluate the impact of the variable reward q on the agent's reward. With the size growing, the more reward will bring to the whole system, and therefore improve the agent's profit.

7.3 Reward versus Number of Miners

Next, we evaluate the impact of the number of miners and the number of pools to the agent's reward. In this experiment, we design three groups of experiments, where the mining pools' configuration can be found in Table 5. We set the size of transactions size in a block of all pools as 30. As shown in Fig. 10, as the number of miners increasing, the agent's reward reduces, which is caused by the competition among miners. Due to the total gain from the mining block is constant, the single agent's profit will decrease with the number of miners increasing. Besides, we find that the agent's reward will increase with the number of pools growing. This result may be explained by the fact that more pools will offer more opportunities for each agent. More choices will reduce the competition among miners, therefore improving the individual agent's profit.

8 Conclusion

In this paper, we propose a cloud mining pool aided BCoT architecture, where the IoT devices can rent the computing resource from the cloud services to offload their mining tasks. In addition, to seek a steady income, the miners are grouped into several mining pools. In the blockchain, these mining pools present themselves as single powerful proxy nodes to gain an advantage over other miners during the mining process. Based on this architecture, we discuss the mining pool selection problem. We first propose a centralized evolutionary game-based pool selection algorithm. A centralized controller is used to guide behaviors of all players. Besides, considering the non-cooperative relationship among miners, we propose a WoLF-PHC based pool selection algorithm. The WoLF-PHC adopts the 'wining or learning fast' scheme to encourage convergence. The agents can constantly adjust their strategies by only interacting with the environment and other agents. The experimental results show that both algorithms can quickly converge to the ESS.

TABLE 5
Experiments Configuration

Mining Pool	Setting
status 1	$\omega_1 = 20, \omega_2 = 30$
status 2	$\omega_1 = 20, \omega_2 = 30, \omega_2 = 40$
status 3	$\omega_1 = 20, \omega_2 = 30, \omega_2 = 40, \omega_2 = 50$

Acknowledgments

This work was supported by the funding from Hong Kong RGC Research Impact Fund (RIF) with the Project No. R5060-19, General Research Fund (GRF) with the Project No. 152221/19E and 15220320/20E, the National Natural Science Foundation of China under Grant 61872310, Shenzhen Science and Technology Innovation Commission (R2020A045), Artificial Intelligence and Smart City Joint Laboratory (BUPT-TGSTII) (B2020001), Future Intelligent Networking and Intelligent Transportation Joint Laboratory (BUPTCTTIC) (B2019007), and the BUPT Excellent Ph.D. Student Foundation under Grant CX2020108.

References

[1] X. Huang, R. Yu, J. Kang, Z. Xia, and Y. Zhang, "Software defined networking for energy harvesting Internet of Things," *IEEE Internet of Things J.*, vol. 5, no. 3, pp. 1389–1399, Jun. 2018.
[2] H. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.
[3] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
[4] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.

[5] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2016, pp. 142–157.
[6] C. Qiu, H. Yao, C. Jiang, S. Guo, and F. Xu, "Cloud computing assisted blockchain-enabled Internet of Things," in *Proc. IEEE Trans. Cloud Comput.*, 2019, p. 1, doi: 10.1109/TCC.2019.2930259.
[7] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?," *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan./Feb. 2018.
[8] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. S. Shen, "Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach," *IEEE Network*, vol. 31, no. 5, pp. 14–20, 2017.
[9] 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 J.*, vol. 6, no. 3, pp. 4585–4600, Jun. 2019.
[10] 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. Inform.*, vol. 15, no. 6, pp. 3602–3609, Jun. 2019.
[11] Z. Li, Z. Yang, and S. Xie, "Computing resource trading for edge-cloud-assisted Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 15, no. 6, pp. 3661–3669, Jun. 2019.
[12] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11 008–11 021, Nov. 2018.
[13] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden, "Incentive compatibility of bitcoin mining pool reward functions," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2017, pp. 477–498.
[14] B. Fisch, R. Pass, and A. Shelat, "Socially optimal mining pools," in *Proc. Int. Conf. Web Internet Economics*, 2017, pp. 205–218.
[15] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2015, pp. 919–927.
[16] S. Kim, "Group bargaining based bitcoin mining scheme using incentive payment process," *Trans. Emerg. Telecommun. Technologies*, vol. 27, no. 11, pp. 1486–1495, 2016.
[17] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, "On power splitting games in distributed computation: The case of bitcoin pooled mining," in *Proc. IEEE 28th Comput. Secur. Foundations Symp.*, 2015, pp. 397–411.
[18] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary game for mining pool selection in blockchain networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 760–763, Oct. 2018.
[19] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis, "Blockchain mining games," in *Proc. ACM Conf. Economics Computation*, 2016, pp. 365–382.
[20] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for IoT-based energy management in smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 111–117, Mar./Apr. 2019.
[21] J. Hofbauer and K. Sigmund, "Evolutionary game dynamics," *Bull. Amer. Math. Soc.*, vol. 40, no. 4, pp. 479–519, 2011.
[22] C. Taylor, D. Fudenberg, A. Sasaki, and M. A. Nowak, "Evolutionary game dynamics in finite populations," *Bull. Math. Biol.*, vol. 66, no. 6, pp. 1621–1644, 2004.
[23] D. Friedman, "Evolutionary games in economics," *Econometrica*, vol. 59, no. 3, pp. 637–666, 1991.
[24] R. Cressman, C. Ansell, and K. Binmore, *Evolutionary Dynamics and Extensive Form Games*, vol. 5. Cambridge, MA, USA: MIT Press, 2003.
[25] F. Mazenc and S. Niculescu, "Lyapunov stability analysis for nonlinear delay systems," *Syst. Control Lett.*, vol. 42, no. 4, pp. 245–251, 2001.
[26] T. Mekki, I. Jabri, A. Rachedi, and M. B. Jemaa, "Vehicular cloud networking: Evolutionary game with reinforcement learning based access approach," *Int. J. Bio-Inspired Computation*, vol. 13, no. 1, pp. 45–58, 2019.
[27] W. He, Y. Liu, H. Yao, T. Mai, N. Zhang, and F. R. Yu, "Distributed variational bayes-based in-network security for the Internet of Things," *IEEE Internet of Things J.*, vol. 8, no. 8, pp. 6293–6304, Apr. 2021.
[28] L. Busoniu, R. Babuska, and B. De Schutter, "Multi-agent reinforcement learning: A survey," in *Proc. 9th Int. Conf. Control Automat. Robot. Vis.*, 2006, pp. 1–6.
[29] M. T. J. Spaan, "Partially observable markov decision processes," in *Proc. Reinforcement Learn.*, 2012, pp. 387–414.
[30] X. Yuan, H. Yao, J. Wang, T. Mai, and M. Guizani, "Artificial intelligence empowered QoS-oriented network association for next-generation mobile networks," *IEEE Trans. Cognitive Commun. Netw.*, vol. 7, no. 3, pp. 856–870, Sep. 2021.
[31] H. Yao, B. Zhang, P. Zhang, S. Wu, C. Jiang, and S. Guo, "RDAM: A reinforcement learning based dynamic attribute matrix representation for virtual network embedding," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 2, pp. 901–914, Second Quarter 2021.
[32] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, Third Quarter 2020.
[33] P. Hernandezleal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Auton. Agents Multi-Agent Syst.*, vol. 33, no. 6, pp. 750–797, 2019.
[34] J. Wang, C. Jiang, K. Zhang, X. Hou, Y. Ren, and Y. Qian, "Distributed q-learning aided heterogeneous network association for energy-efficient IIoT," *IEEE Trans. Ind. Inform.*, vol. 16, no. 4, pp. 2756–2764, Apr. 2020.
[35] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, "Evolutionary dynamics of multi-agent learning: A survey," *J. Artif. Intell. Res.*, vol. 53, no. 1, pp. 659–697, 2015.
[36] Y. Zhang, R. Yu, M. Nekovee, Y. Liu, S. Xie, and S. Gjessing, "Cognitive machine-to-machine communications: visions and potentials for the smart grid," *IEEE Netw.*, vol. 26, no. 3, pp. 6–13, May/Jun. 2012.
[37] D. Niyato and E. Hossain, "Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2008–2017, May 2009.
[38] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.

Tianle Mai (Student Member, IEEE) is working toward the PhD degree at the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing. His research interests include future network architecture, network artificial intelligence, multi-agent system, space-terrestrial integrated network, network resource allocation and dedicated networks.

Haipeng Yao (Senior Member, IEEE) received the PhD from the Department of Telecommunication Engineering, University of Beijing University of Posts and Telecommunications, in 2011. He is currently an associate professor with the Beijing University of Posts and Telecommunications. His research interests include future network architecture, network artificial intelligence, networking, space-terrestrial integrated network, network resource allocation and dedicated networks. He has published more than 100 papers in prestigious peer-reviewed journals and conferences. He has served as an editor of *IEEE Network*, *IEEE Access*, and a guest editor of *IEEE Open Journal of the Computer Society* and Springer Journal of Network and Systems Management. He has also served as a member of the technical program committee as well as the Symposium chair for a number of international conferences, including IWCMC 2019 Symposium chair, ACM TUR-C SIGSAC2020 publication chair.

Ni Zhang received the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2007. He currently serves with the Sixth Research Institute of China Electronic Corporation, Beijing, China. His research interests include the area of future internet architecture, network security and artificial intelligence.

Lexi Xu (Member, IEEE) received MS degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009, and the PhD degree from the Queen Mary University of London, London, United Kingdom, in 2013. He is currently a senior engineer with China Unicom Research Institute. He is also a China Unicom delegate in ITU, ETSI, CCSA. He served as Workshop Chair of ICSINC, IEEE ISCIT, 5GWN, NGDN, IEEE IUCC. His research interests include big data, self-organizing networks, radio resource management.

Mohsen Guizani (Fellow, IEEE) received the BS (Hons.) and MS degrees in electrical engineering and the MS and PhD degrees in computer engineering from Syracuse University, Syracuse, NY, U.S. in 1984, 1986, 1987, and 1990, respectively. He is currently a professor with the Computer Science and Engineering Department, Qatar University, Qatar. Previously, he has served in different academic and administrative positions with the University of Idaho, Western Michigan University, University of West Florida, University of Missouri-Kansas City, University of Colorado Boulder, and Syracuse University. He is the author of nine books and more than 600 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is a senior member of ACM. He also served as a member, the chair, and the general chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He was a recipient of 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, 2018 Ad Hoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks, and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award for outstanding contributions to the technological advancement of security. He was the chair of IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He has served as the IEEE Computer Society Distinguished Speaker. He is currently the IEEE ComSoc distinguished lecturer. He guest edited a number of special issues in IEEE journals and magazines. He is also the editor-in-chief of IEEE Network Magazine. He serves on the editorial boards for several international technical journals and the Founder and the editor-in-chief for Wireless Communications and Mobile Computing (Wiley).

Song Guo (Fellow, IEEE) is currently a full professor with the Department of Computing, The Hong Kong Polytechnic University. He also holds a changjiang chair professorship awarded by the Ministry of Education of China. He is a fellow of the Canadian Academy of Engineering and a fellow of the IEEE (Computer Society). His research interests include big data, edge AI, mobile computing, and distributed systems. He published many papers in top venues with wide impact in these areas and was recognized as a Highly Cited Researcher (Clarivate Web of Science). He is the recipient of over a dozen best paper awards from IEEE/ACM Conferences, Journals, and Technical Committees. He is the editor-in-chief of IEEE Open Journal of the Computer Society and the chair of IEEE Communications Society (ComSoc) Space and Satellite Communications Technical Committee. He was an IEEE ComSoc distinguished lecturer and a member of IEEE ComSoc Board of Governors. He has served for IEEE Computer Society on fellow Evaluation Committee, and been named on editorial board of a number of prestigious international journals like *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Emerging Topics in Computing*, etc. He has also served as chairs of organizing and technical committees of many international conferences.