HIGH TECHNOLOGY LETTERS | Vol. 29 No. 2 | Jun. 2023 | pp. 181-193

doi:10.3772/j.issn.1006-6748.2023.02.009

Joint optimization of serving node selection and wireless resources allocation for transactions data in mobile blockchain enhanced Internet of Things①

YIN Yufeng(尹玉峰), WU Wenjun, GAO Yang②, JIN Kaiqi, ZHANG Yanhua, SUN Teng
(Faculty of Information Technology, Beijing University of Technology, Beijing 100124, P. R. China)

Abstract

With the increased emphasis on data security in the Internet of Things (IoT), blockchain has received more and more attention. Due to the computing consuming characteristics of blockchain, mobile edge computing (MEC) is integrated into IoT. However, how to efficiently use edge computing resources to process the computing tasks of blockchain from IoT devices has not been fully studied. In this paper, the MEC and blockchain-enhanced IoT is considered. The transactions recording the data or other application information are generated by the IoT devices, and they are offloaded to the MEC servers to join the blockchain. The practical Byzantine fault tolerance (PBFT) consensus mechanism is used among all the MEC servers which are also the blockchain nodes, and the latency of the consensus process is modeled with the consideration of characteristics of the wireless network. The joint optimization problem of serving base station (BS) selection and wireless transmission resources allocation is modeled as a Markov decision process (MDP), and the long-term system utility is defined based on task reward, credit value, the latency of infrastructure layer and blockchain layer, and computing cost. A double deep Q learning (DQN) based transactions offloading algorithm (DDQN-TOA) is proposed, and simulation results show the advantages of the proposed algorithm in comparison to other methods.

Key words: Internet of Things(IoT), mobile edge computing(MEC), blockchain, deep reinforcement learning(DRL)

0 Introduction

The Internet of Things (IoT) which is an important technology supporting intelligent application scenarios such as smart cities, smart transportation, smart medical care, and logistics has aroused lots of attention[1]. Due to the massively scattered terminals of IoT and the large amounts of data collected by IoT devices, data security and privacy has become key issue in this field[2]. Recent research shows that blockchain technology is a promising solution as it has the characteristics of decentralization, whole-process traceability, and transparency[3].

However, blockchain is a computing consuming technology, which cannot be directly implemented on resource-limited IoT devices[4]. Integrating mobile edge computing (MEC) with IoT provides the close-to-user computing capabilities to execute blockchain, the problem of computing task unloading decision is introduced, and this problem is widely studied[5-7].

Generally, how to efficiently use edge computing resources to process the computing tasks from blockchain is a matter of prime importance, and auction-based resource allocation algorithms are popular solutions for this kind of problem[8-10]. Considering the characteristics of wireless transmission, the optimization of MEC-enhanced blockchain systems is further researched. Energy efficiency is one of the hot research topics, and a geometric programming method is used to optimize the ratio of overall system throughput to system average power consumption in blockchain-based IoT[11]. With the constraints of probabilistic backhaul and delay constraints, the computation and data caching rewards are optimized by an alternating direction method of a multipliers-based algorithm[12].

Moreover, both the performance measurements of wireless transmission and blockchain systems are integrated to form the optimization objective in the deep reinforcement learning (DRL) based system optimization

① Supported by the National Key Research and Development Program of China (No. 2020YFC1807903) and the Natural Science Foundation of Beijing Municipality (No. L192002).

② To whom correspondence should be addressed. E-mail: cathyshou@emails.bjut.edu.cn.

Received on Sep. 15, 2022

algorithms. Taking the cost of latency in the blockchain system into the reward, a joint optimization algorithm of caching and computation for delay-tolerant data in machine-to-machine communications networks is proposed based on dueling deep Q-network[13]. The rewards obtained by uploading data to the blockchain system are also considered, and the scheduling of data processing task requests[14] and the node selection algorithm of the computing resources providers are proposed[15] based on the policy gradient algorithm, respectively. By modeling both the rewards of executing smart contracts and the latency of the blockchain system in the system reward, the joint optimization of the offloading decisions, the allocation of computing resources and radio bandwidth, and the smart contract usage are studied[16]. Although the latency in the blockchain system is modeled with the consideration of different consensus processes, the communication among different blockchain nodes which are also accessed points in the wireless access networks is not modeled in detail.

Focusing more on the characteristics of blockchain, the historical reputation and the accumulated credit of blockchain nodes have an important impact on service quality[17-19]. However, in the blockchain and MEC-enhanced IoT, only a few studies have considered this in the computing offloading and task scheduling problems[14-15].

In this paper, the MEC-enhanced IoT is considered, and the blockchain is enabled to provide secure and traceable management of IoT applications. The transactions recording the data or other application information are generated by the IoT devices. And they are offloaded to the base stations (BSs) and MEC servers in the heterogeneous network (HetNets) to be processed and packaged in a block to join the blockchain. The selection of serving BS and the allocation of wireless transmission resources are jointly optimized. To ensure the enthusiasm of blockchain nodes to serve the IoT applications, the system utility is defined considering task reward, the credit value, latency of the infrastructure layer and blockchain layer, and computing cost. A double deep Q learning (DDQN) based transaction offloading algorithm is proposed, and simulation results show the advantages of the proposed algorithm in comparison to other methods. The main contribution of this paper is further summarized in the following.

The communication latency among different blockchain nodes is modeled in detail. Due to the large number of BSs in the blockchain-enhanced HetNets, the X2 interface which can forward messages to other BSs may not be built between any two BSs. Therefore, a time-varying angular symmetric matrix is used to model the real-time direct connection status of the BSs, and the real-time communication latency among different BSs is approximately measured by the number of hops of the shortest path.

The credit weight of each BS is defined based on the counter of successfully processed transactions, and this weight is used in the reward to model the impact of credit on service quality. The counter increases if the transaction is successfully uploaded to the blockchain. Otherwise, the counter decreases. Then, the credit ratio is defined with the average counter as the reference. Finally, an exponential function of this ratio is used in the definition of credit weight which is used in the calculation of the profit of uploading a transaction.

The joint optimization of the selection of serving BS and the allocation of wireless transmission resources is modeled as an Markov decision process (MDP), and the solution is designed based on DDQN. To ensure the enthusiasm of blockchain nodes to serve the IoT applications, the cumulative reward is defined based on the long-term system utility relating to task reward, credit value, the latency of the infrastructure layer and blockchain layer, and computing cost. To facilitate the design of the solution, the states and the actions are also unified, respectively. The proposed DDQN-based transactions offloading algorithm is described in detail and evaluated through comparison.

The rest of this paper is organized as follows. First, the system model is proposed in Section 1. Then the problem formulation is proposed in Section 2. And in Section 3, the algorithm flow is proposed. Besides, Section 4 is the simulation result and test result of the algorithm. Finally, Section 5 is the summary of the paper.

1 System Model

1.1 System architecture

In this paper, the system is divided into two layers, i. e. the infrastructure layer and the blockchain layer[20], and the system architecture is shown in Fig. 1.

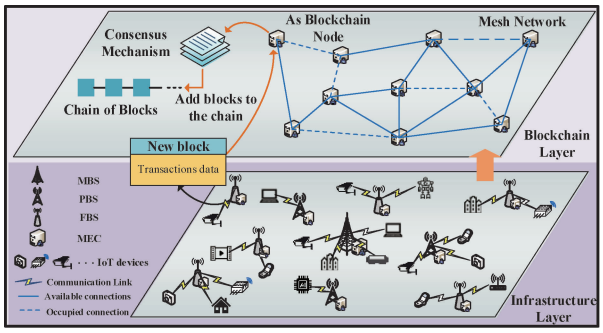

Fig. 1 System model

The HetNets with mobile edge computing (MEC) is the infrastructure layer. There are N base stations (BSs) which are N_{m} macro BSs (MBSs), pico BSs (PBSs) and N_{f} femto BSs (FBSs). The frequency resources are quantified as resource blocks (RB), and the initial number of RBs allocated to the three kinds of BSs are f_{m}, f_{p} and f_{f}, respectively. Assume each BS deploys a MEC server (MECS), and the computing capability of the three kinds of BSs are c_{m}, c_{p} and c_{f}, respectively. The n-th BS in the system is denoted by B_n, and its initial number of RBs and computing capability is denoted by f_{n} and c_{n}, respectively.

The IoT applications in the system are managed by blockchain. The transactions recording the data or other application information are generated by the IoT devices. Due to the limited capabilities of the IoT devices, these transactions are offloaded to the BSs, and then, processed in the MECS to be packaged in a block to join the blockchain. Therefore, the blockchain layer consists of all the MECS in the system. Assume the practical Byzantine fault tolerance (PBFT) consensus mechanism is used. The MECS of the MBS is the primary node, and the other MECSs are the common nodes. Any MECS which wants to upload a block to the blockchain is considered as the client node of the corresponding consensus process, and the remaining MECSs are the replica nodes.

Due to the large number of BSs in the blockchain-enhanced HetNets, the X2 interface which can forward messages to other BSs may not be built between any two BSs. Thus, a time-varying angular symmetric matrix $\boldsymbol{Y}_t = [y_{i,j,t}]_{N\times N}$ is used to denote the real-time direct connection status of the BSs in the system. $y_{i,j,t} = y_{j,i,t} = 1$ means there is a direct connection between B_i and B_j at time step t, otherwise, $y_{i,j,t} = y_{j,i,t} = 0$. The number of hops $\zeta_{i,j,t}$ between B_i and B_j under the condition of $\boldsymbol{Y}_t$ is used to represent the corresponding transmission latency between the MECSs of B_i and B_j in the consensus process approximately.

1.2 Transactions model

Assume the arrival of the transactions in the system follows the Poisson process with arrival rate λ. Besides, the IoT devices are randomly scattered in the area, and each device only generates one transaction at a location. The unserved transactions and the newly coming transactions are recorded in the queue of the system. $M_t = \{M_{1,t}, M_{2,t}, \cdots, M_{i,t}, \cdots\}$ denotes the queue at the beginning of time step t, where $M_{i,t}$ is the i-th transaction in the queue. Define $M_{i,t} = \{L_{i,t}, D_{i,t}, C_{i,t}, \tau_{i,t}, R_{i,t}\}$, where $L_{i,t}$, $D_{i,t}$, $C_{i,t}$, $\tau_{i,t}$ and $R_{i,t}$ are the location, the data size, the required computing resources, the maximum tolerable latency and the expected profit of transaction $M_{i,t}$.

1.3 Latency in the infrastructure layer

The latency of transaction $M_{i,t}$ in the infrastructure layer mainly consists of the queuing time $\tau^{\mathrm{q}}_{i,t}$ and the wireless transmission time $\tau^{\mathrm{w}}_{i,t}$ from the IoT device to the selected serving BS.

The wireless transmission time of $M_{i,t}$ can be estimated as

$$\tau^{\mathrm{w}}_{i,t} = \frac{D_{i,t}}{v_{i,t}} \tag{1}$$

where $v_{i,t}$ is the wireless transmission rate between the IoT device generating $M_{i,t}$ and the selected serving BS $a^{\mathrm{b}}_{i,t} = B_n$, and

$$v_{i,t} = a^{\mathrm{f}}_{i,t} bt_{\mathrm{s}} \log_2\left(1 + \frac{P_{i,t} h_{i,t,n} (l_{i,t,n})^{-\alpha}}{\sigma^2 + I_{i,t,n}(N-1)a^{\mathrm{f}}_{i,t}}\right) \tag{2}$$

where $a^{\mathrm{f}}_{i,t}$ is the number of RBs allocated for transmitting $M_{i,t}$, b is the bandwidth of a RB, $P_{i,t}$ is the transmitting power of $M_{i,t}$, t_{s} is the interval of a time step, σ^2 is the noise power, $I_{i,t,n}$ is the average co-channel interference on each RB, $h_{i,t,n}$ and $(l_{i,t,n})^{-\alpha}$ are the Rayleigh fading coefficient and the path-loss of the channel between the IoT device generating $M_{i,t}$ and the BS B_n, respectively.

To approximately calculate $I_{i,t,n}$, it can be assumed all the remaining tasks in the queue are served in the system, and thus

$$I_{i,t,n} = \frac{\sum_{M_{j,t}\in M_t, j\neq i} P_{j,t} h_{j,t,n} (l_{j,t,n})^{-\alpha}}{N_{\mathrm{m}} f_{\mathrm{m}} + N_{\mathrm{p}} f_{\mathrm{p}} + N_{\mathrm{f}} f_{\mathrm{f}} - a^{\mathrm{f}}_{i,t}} \tag{3}$$

To quantify the data size of $M_{i,t}$ by the total number of required RBs, the wireless transmission rate between the IoT device generating $M_{i,t}$ and an arbitrary base station B_n is also calculated as

$$v'_{i,t,n} = bt_{\mathrm{s}} \log_2\left(1 + \frac{P_{i,t} h_{i,t,n} (l_{i,t,n})^{-\alpha}}{\sigma^2 + I_{i,t,n}(N-1)}\right) \tag{4}$$

thus, the total number of RBs required by $M_{i,t}$ if accessing to B_n can be estimated as

$$\delta_{i,t,n} = \frac{D_{i,t}}{v'_{i,t,n}} \tag{5}$$

Based on the above-mentioned estimation, the vector $\boldsymbol{\delta}_{i,t} = [\delta_{i,t,1}, \cdots, \delta_{i,t,n}, \cdots, \delta_{i,t,N}]$ is defined as an important characteristic of $M_{i,t}$ considering the wireless transmission environment.

1.4 Latency of the consensus process

Generally, the PBFT consensus process consists of five phases, i.e., the request phase, the pre-prepare

phase, the prepare phase, the commit phase, and the reply phase.

(1) Request phase: the client node signs the *REQUEST* message and sends it to the primary node. In this phase, the delay is $\zeta_{c,p,t}$, where B_c and B_p are the client node and the primary node, respectively.

(2) Pre-prepare phase: the primary node receives the *REQUEST* message and verifies that the *REQUEST* message from the client node is signed correctly. Then, the primary node broadcasts a *PRE – REQUEST* message to all the replica nodes. Thus, the latency of the pre-prepare phase is $\max_{B_r \in R_t} \zeta_{p,r,t}$, where R_t is the set of replica nodes of this consensus process.

(3) Prepare phase: the replica node receives the *PRE – REQUEST* message from the primary node and verifies whether the signature and information are correct. If they are correct, the replica node broadcasts the verification information which is a *REQUEST* message to all the nodes except the client node. The latency of the prepare phase is then calculated as $\max_{B_{r'} \in R_t,\ B_r \in B_t} \zeta_{r',r,t}$, where $B_t = R_t \cup B_p$.

(4) Commit phase: the primary node and the replica nodes receive the *REQUEST* message and check whether the signature is correct. If the primary node or the replica node receives more than $2(N-2)/3$ *REQUEST* messages that have passed the verification, it broadcasts a *COMMIT* message and records the received *REQUEST* messages. The latency of the commit phase is $\max_{B_{r'} \in B_t,\ B_r \in B_t} \zeta_{r',r,t}$.

(5) Reply phase: the primary node and the replica node receive the *COMMIT* message and check whether the signature and information are correct. If the replica node receives more than $2(N-2)/3$ *COMMIT* messages that have passed the verification, most of the nodes in the system have reached a consensus. The *REPLY* message is executed, and a message is returned to the client node. The latency of the reply phase is $\max_{B_r \in R_t} \zeta_{r,c,t}$.

Taking all the phases into account, the latency of the consensus process of $M_{i,t}$ can be calculated as:

$$\tau^{\mathrm{b}}_{i,t} = \zeta_{c,p,t} + \max_{B_r \in R_t} \zeta_{p,r,t} + \max_{B_{r'} \in R_t,\ B_r \in B_t} \zeta_{r',r,t} + \max_{B_{r'} \in B_t,\ B_r \in B_t} \zeta_{r',r,t} + \max_{B_r \in R_t} \zeta_{r,c,t}, \tag{6}$$

1.5 Credit model

As the historical reputation of the blockchain node affects its performance in the blockchain system, the credit model is defined, and the reward of successfully uploading a transaction to the blockchain is affected by the credit value.

Generally, the credit of each node increases if a transaction is successfully uploaded to the blockchain. Otherwise, the credit decreases. Use $H_{t,n}$ to denote the counter of B_n at time step t. When the consensus process is accomplished, $H_{t,n}$ is updated before the next time step. If B_n successfully completes the transaction, $H_{t,n}$ increases 1. Otherwise, $H_{t,n}$ decreases 1. To restrict the range of credit, the credit ratio is defined as

$$\eta_{n,t} = \frac{H_{t,n} - \frac{\sum_{n \in N} H_{t,n}}{N}}{\frac{\sum_{n \in N} H_{t,n}}{N}} \tag{7}$$

when the transaction $M_{i,t}$ is successfully uploaded to the blockchain through B_n at time step t, the credit weight

$$\omega_{n,t} = \frac{1}{1 + e^{-\eta_{n,t}}} + 0.5 \tag{8}$$

is used to calculate the real profit. According to the definition of $\omega_{n,t}$, its growth gradually slows down with the increase of credit value.

2 Problem formulation

According to the system model, the selection of serving BS and the allocation of wireless transmission resources for transmitting the transaction to the serving BS is the main factor affecting the service quality. Thus, these two problems are jointly optimized. To ensure the enthusiasm of blockchain nodes to serve the IoT applications, the optimization problem aims at maximizing the system utility related to task reward, credit value, the latency of the infrastructure layer and blockchain layer, and computing cost. The detailed formulation of the problem is given in the following.

2.1 Optimization problem

Assume each observation period of the system lasts for T time steps. The optimization problem is formulated as

$$\begin{aligned} &\max\ G = \mu_{\mathrm{P}} G_{\mathrm{P}} - \mu_{\mathrm{C}} G_{\mathrm{C}} - \mu_{\mathrm{Q}} G_{\mathrm{Q}} - \mu_{\mathrm{W}} G_{\mathrm{W}} - \mu_{\mathrm{B}} G_{\mathrm{B}} \\ &\text{s. t.}\ \ \tau^{\mathrm{w}}_{i,t} + \tau^{\mathrm{b}}_{i,t} + \tau^{\mathrm{q}}_{i,t} < \tau_{i,t} \end{aligned} \tag{9}$$

where, μ_{P}, μ_{C}, μ_{Q}, μ_{W} and μ_{B} are weights, G_{P} is the task profit obtained by all the blockchain nodes, G_{C} is the computing cost, G_{Q} is the queuing cost, G_{W} and G_{B} are the cost of wireless transmission time and the cost of blockchain layer latency, respectively. The definitions of G_{P}, G_{C}, G_{Q}, G_{W} and G_{B} are given as

$$
\begin{aligned}
G_{\mathrm{P}} &= \sum_{t\in T}\sum_{i\in M_t}[R_{i,t}\omega_{n,t}(1-\xi^{\mathrm{q}}_{i,t})\xi^{\mathrm{s}}_{i,t}]\\
G_{\mathrm{C}} &= \sum_{t\in T}\sum_{i\in M_t}[C_{i,t}(1-\xi^{\mathrm{q}}_{i,t})]\\
G_{\mathrm{Q}} &= \sum_{t\in T}\sum_{i\in M_t}\xi^{\mathrm{q}}_{i,t}\\
G_{\mathrm{W}} &= \sum_{t\in T}\sum_{i\in M_t}[\tau^{\mathrm{w}}_{i,t}(1-\xi^{\mathrm{q}}_{i,t})]\\
G_{\mathrm{B}} &= \sum_{t\in T}\sum_{i\in M_{tt}}[\tau^{\mathrm{b}}_{i,t}(1-\xi^{\mathrm{q}}_{i,t})]
\end{aligned}
\tag{10}
$$

where $\xi^{\mathrm{q}}_{i,t}$ and $\xi^{\mathrm{s}}_{i,t}$ are the indicators of queueing status and service status of $M_{i,t}$, respectively. $\xi^{\mathrm{q}}_{i,t}=1$ means that $M_{i,t}$ has not been offloaded to any MECS by the end of time step t, otherwise, $\xi^{\mathrm{q}}_{i,t}=0$. $\xi^{\mathrm{s}}_{i,t}=1$ means $M_{i,t}$ is offloaded to the BS B_n at time step t, otherwise, $\xi^{\mathrm{s}}_{i,t}=0$.

2.2 Markov decision process (MDP) model

Assume the MBS oversees the offloading decision of all the transactions generated by the IoT devices in the area. Both the offloading target BS and the number of RBs allocated for transmitting each transaction need to be selected to maximize the system utility.

As the arrival of the transactions in the system follows the Poisson process for a long time and the available transmission and computing resources are updated every time step, the transactions offloading decision problem can be modeled as an MDP which is defined as a tuple (S, A, P, R), where S and A denote the state space and the action space, respectively. $P(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')$ is the state transition probability from the state $\boldsymbol{s}\in S$ to the state $\boldsymbol{s}'\in S$ by taking the action $\boldsymbol{a}\in A$. R is the reward function and $\boldsymbol{r}=R(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')$ is the reward obtained by taking the action $\boldsymbol{a}\in A$ at the state $\boldsymbol{s}\in S$ and transiting to the state $\boldsymbol{s}'\in S$.

In the following of this section, the state, the action, and the reward function are defined in detail.

(1) State

The state is defined as

$$
\boldsymbol{s}_{i,t}=(t, \boldsymbol{S}^{\mathrm{B}}_t, \boldsymbol{S}^{\mathrm{T}}_{i,t}, \boldsymbol{S}^{\mathrm{T}}_{i+1,t}, \cdots) \tag{11}
$$

where t is index of time, $\boldsymbol{S}^{\mathrm{B}}_t$ is the information of all the BSs, $S^{\mathrm{T}}_{i,t}$ is the information of $M_{i,t}$.

$$
\boldsymbol{S}^{\mathrm{B}}_t=[\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{H}_t] \tag{12}
$$

where $\boldsymbol{H}_t=[H_{1,t}, \cdots, H_{n,t}, \cdots, H_{N,t}]$ is the array of BS's credit value at the time step t, $\boldsymbol{C}_t$ and $\boldsymbol{F}_t$ are arrays of the available computing and transmission resources of all the BSs, respectively. To compromise between accuracy and complexity, the estimated available resources from time step t to $t+30$ are observed.

$$
\boldsymbol{S}^{\mathrm{T}}_{i,t}=[\mathrm{t}^{d}_{i,t}, \Delta_{i,t}, \mathrm{C}_{i,t}, \mathrm{R}_{i,t}] \tag{13}
$$

where $t^{\mathrm{d}}_{i,t}$ is the time remaining before the timeout of $M_{i,t}$ occurs and $\boldsymbol{\Delta}_{i,t}=[\delta_{i,t,1}, \cdots, \delta_{i,t,n}, \cdots, \delta_{i,t,N}]$.

(2) Action

The action consists of two parts which are the selected offloading target BS $a^{\mathrm{b}}_{i,t}$ and the number of RBs $a^{\mathrm{f}}_{i,t}$ allocated for transmitting $M_{i,t}$, and $\boldsymbol{a}_{i,t}=[a^{\mathrm{b}}_{i,t}, a^{\mathrm{f}}_{i,t}]$. In order to decrease the size of action space, $a^{\mathrm{f}}_{i,t}$ is quantified into 5 levels.

(3) Reward

To be consistent with the system utility optimization problem, the cumulative reward of the whole MDP should be G given in Eq. (9). The system utility is decomposed into the reward of each offloading decision at each time step. For the offloading decision of transaction $M_{i,t}$, the corresponding reward is

$$
\begin{aligned}
r_{i,t}(\boldsymbol{a}_{i,t}) &= [\mu_{\mathrm{P}}R_{i,t}\omega_{n,t}\xi^{\mathrm{s}}_{i,n,t}(a^{\mathrm{b}}_{i,t}) - \mu_{\mathrm{C}}C_{i,t} - \mu_{\mathrm{W}}\tau^{\mathrm{w}}_{i,t}\\
&(\boldsymbol{a}_{i,t}) - \mu_{\mathrm{B}}\tau^{\mathrm{b}}_{i,t}(a^{\mathrm{b}}_{i,t})]\\
&[1-\xi^{\mathrm{q}}_{i,t}(a^{\mathrm{b}}_{i,t})] - \mu_{\mathrm{Q}}\xi^{\mathrm{q}}_{i,t}(a^{\mathrm{b}}_{i,t})W
\end{aligned}
\tag{14}
$$

where the queueing status indicator $\xi^{\mathrm{q}}_{i,t}$, the service status indicator $\xi^{\mathrm{s}}_{i,n,t}$, the estimated wireless transmission time $\tau^{\mathrm{w}}_{i,t}$, and the latency of the consensus process $\tau^{\mathrm{b}}_{i,t}$ are affected by the action, W is the penalty of every queuing.

3 Deep reinforcement learning-based transactions offloading algorithm

According to the MDP model-based transactions offloading decision problem, the state space is extremely large, and the action space is also complicated. Thus, it is difficult to use conventional methods to solve the MDP. Luckily, recent researches show that the DRL-based method can solve the complicated MDP, and the Double DQN[21-23] is adopted as the transactions offloading decision agent in this paper.

Generally, the number of transactions in M_t varies at each decision step. To ensure the stable structure of the Double DQN, the state input to the neural networks is modified as

$$
\hat{\boldsymbol{s}}_{i,t}=(t, \boldsymbol{S}^{\mathrm{B}}_t, \boldsymbol{S}^{\mathrm{T}}_{i,t}, \cdots, \boldsymbol{S}^{\mathrm{T}}_{i+X,t}) \tag{15}
$$

which means only the first X transactions in M_t are observed for each decision. If the number of transactions in M_t is less than X, the corresponding positions in $\hat{\boldsymbol{s}}_{i,t}$ will be filled with 0.

As for the action, this work computationally maps the two-dimensional decision action $\boldsymbol{a}_{i,t}$ to one-dimensional action $\hat{a}_{i,t}$ to simplify the design of neural networks, where $\hat{a}_{i,t}=5a^{\mathrm{b}}_{i,t}+a^{\mathrm{f}}_{i,t}$.

The workflow of the proposed DDQN-based transactions offloading algorithm (DDQN-TOA) is shown in Fig. 2 and the detailed pseudo-code is given in Algo-

rithm 1. In Fig. 2, first, the state $\hat{s}_{i,t}$ is input to the Q-evaluation network. And then, the action $\hat{a}_{i,t}$ is selected using the ε-greedy policy. Finally, the parameter θ of the Q-evaluation network is updated as

$$\theta_{\text{new}} = \theta_{\text{old}} + \alpha\,(y_k - Q(\hat{s}_k, \hat{a}_k; \theta_{\text{old}}))\,\nabla_{\theta_{\text{old}}} Q(\hat{s}_k, \hat{a}_k; \theta_{\text{old}}) \quad (16)$$

where Q is the value function represented by the neural network, α is the learning rate,

$$y_k = r_k + \gamma Q(\hat{s}_k, \underset{\hat{a}_k}{\operatorname{argmax}} Q(\hat{s}_k, \hat{a}_k; \theta_{\text{old}}); \theta'_{\text{old}}) \quad (17)$$

where, γ is the discount ratio, and θ' is the parameter of the Q-target network. $(\hat{s}_k, \hat{a}_k, r_k, \hat{s}_k')$ is obtained by sampling from replay memory.

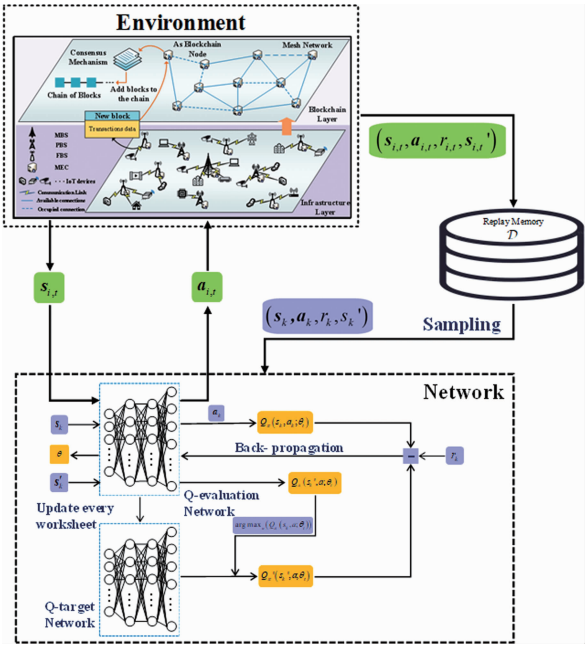

Fig. 2 DDQN-TOA workflow

Algorithm 1 The process of one iteration simulation in DDQN-TOA

1. Input: T, D, θ and θ'.
2. Initialize the state of RBs. $\boldsymbol{F} = [\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_T]$, where $\boldsymbol{F}_t = [F_{t,1}, F_{t,2}, \cdots, F_{t,N}] = [f_n]_{1\times N}$ for all $t \in [1, T]$.
3. Initialize the state of computing capabilities. $\boldsymbol{C} = [\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_T]$, where $\boldsymbol{C}_t = [C_{t,1}, C_{t,2}, \cdots, C_{t,N}] = [c_n]_{1\times N}$ for all $t \in [1, T]$.
4. Initialize the credit counter of all the BSs $\boldsymbol{H} = [H_1, H_2, \ldots, H_N]$.
5. Initialize $t = 0$.
6. Initialize temporary variables $M' = \varphi$ and $\boldsymbol{H}' = [0]_{1\times N}$.
7. **for** t in T **do**:
8. $M_t \leftarrow M'$ and update the information of all the transactions in M_t.
9. Add new arrival transactions to the end of M_t.
10. **for each** $M_{i,t} \in M_t$ **do**:
11. Observe the current state $s_{i,t}$ according to Eq. (11).
12. Input $s_{i,t}$ to the Q-evaluation network and obtain the action $\boldsymbol{a}_{i,t} = [a^{\text{b}}_{i,t}, a^{\text{f}}_{i,t}]$ according to the ε-greedy policy.
13. **if** $a^{\text{r}}_{i,t} > F_{t,a^{\text{b}}_{i,t}}$ or $C_{i,t} > C_{t,a^{\text{b}}_{i,t}}$ **then**
14. $\xi^{\text{q}}_{i,t} = 1, \xi^{\text{s}}_{i,t} = 0$.
15. Remove $M_{i,t}$ form M_t and add it to the end of M'.
16. **else**
17. $\xi^{\text{q}}_{i,t} = 0$.
18. Consume resource of BS $B_{a^{\text{b}}_{i,t}}$ and update $\boldsymbol{F}$ and $\boldsymbol{C}$.
19. Compute $\tau^{\text{w}}_{i,t}$ and $\tau^{\text{b}}_{i,t}$ according to Eqs (1) and (6), respectively.
20. Remove $M_{i,t}$ form M_t.
21. **if** $\tau^{\text{w}}_{i,t} + \tau^{\text{b}}_{i,t} < t^{\text{d}}_{i,t}$ **then**
22. $\xi^{\text{s}}_{i,t} = 1$ and $H'_{a^{\text{b}}_{i,t}} = H'_{a^{\text{b}}_{i,t}} + 1$.
23. **else**:
24. $\xi^{\text{s}}_{i,t} = 0$ and $H'_{a^{\text{b}}_{i,t}} = H'_{a^{\text{b}}_{i,t}} - 1$.
25. **end if**
26. **end if**
27. Compute $\boldsymbol{r}_{i,t}(\boldsymbol{a}_{i,t})$ using Eq. (14).
28. Observe the next state $\boldsymbol{s}_{i+1,t}$ and $\boldsymbol{s}'_{i,t} = \boldsymbol{s}_{i+1,t}$, add experience $(\boldsymbol{s}_{i,t}, \boldsymbol{a}_{i,t}, \boldsymbol{r}_{i,t}, \boldsymbol{s}'_{i,t})$ to D.
29. Randomly select a minibatch of experiences D' from D
30. **for each** $\boldsymbol{d}_k = (\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{r}_k, \boldsymbol{s}'_k) \in D'$ **do**:
31. **if** $\boldsymbol{s}_k$ is the terminal of an episode **then**
32. $y_k = r_k$
33. **else**
34. Calculate y_k using Eq. (16).
35. **end if**
36. **end for**
37. Update θ using Eq. (15).
38. **end for**
39. $\boldsymbol{H} = \boldsymbol{H} + \boldsymbol{H}'$
40. **end for**
41. Update the parameters of the Q-target network $\theta' \leftarrow \theta$.
42. Output: D, θ and θ'.

4 Performance evaluation

To evaluate the performance of the proposed DDQN-TOA, three basic transactions offloading algorithms which are named MAX-SINR, MAX-Credit and Greedy are implemented for comparison. MAX-Credit algorithm selects the BS with the highest credit value to serve the transaction. MAX-SINR algorithm selects the BS with the best channel quality to serve the transaction. Greedy algorithm selects the action which can obtain the highest reward to offload the transaction.

The main parameters of the evaluation scenario are given in Table I. As for the DDQN-TOA, two three-layer neural networks are used as the Q-evaluation and Q-target networks, respectively. The hidden layer of each

neural network has 256 neurons.

To verify the generalization ability of the algorithm, this work uses the different arrival rate $\lambda = \pi R_l^2 \times \gamma$, ($\gamma = 0.000\ 05$, $0.000\ 06$, $0.000\ 07$, $0.000\ 08$).

Table 1 Parameters of the evaluation scenario

Symbol	Parameter	Setting
R_l	Radius of two-dimensional Poisson distribution and task generated	200 m
$D_{i,t}$	Data size of the task $M_{i,t}$ uniformly and randomly generated from the setting	[5, 30]
$C_{i,t}$	Required computing resource of the task $M_{i,t}$ uniformly and randomly generated from the setting	[5, 10]
$\tau_{i,t}$	The maximum tolerable delay of the task $M_{i,t}$ uniformly and randomly generated from the setting	[3, 10]
$R_{i,t}$	The reward of the task $M_{i,t}$ uniformly and randomly generated from the setting	[20, 25, 30, 35, 75, 80, 85, 95, 100]
N	The total number of BSs	10
N_m, N_p, N_f	The number of MBS, PBS, FBS	1, 3, 6
M	The total number of frequency resource blocks consumed by BSs	275
T	Time length of system observed	300
f_m, f_p, f_f	MBS, FBS, PBS frequency resource blocks number	50, 25, 25
c_m, c_f, c_p	MBS, FBS, PBS computing resource blocks number	100, 80, 60
σ^2	Noise power of single RB	7.2e-13 mW
α	Path loss exponent	2
α_l	Learning rate	0.000 1
μ_P, μ_W, μ_B, μ_C	The weight of pure reward, latency of wireless transmission, latency of blockchain layer and computing cost	0.2, 0.3, 0.3, 0.2
μ_Q	The cost of queuing once	10
t_s	The interval of a time step	10 ms
b	The bandwidth of single RB	180 kHz
W	The penalty of once queuing	10
$P_{i,t}$	The transmitting power	20 mW

4.1 Training phase

In this section, the DDQN performing as the transactions offloading decision agent is trained with different transactions arrival rates which are $\gamma \in \{0.000\ 05, 0.000\ 06, 0.000\ 07, 0.000\ 08\}$. For each transactions arrival rate, 10 independent transactions flows are generated following the Poisson process within 300 time steps. The training results are given from Fig. 3 to Fig. 6.

In general, the training process is convergent under different system settings, and the proposed DDQN-TOA performs the best among all the algorithms evaluated in this paper. When $\gamma = 0.000\ 07$, the cumulative reward converges to the largest value. Obviously, with the increase of transactions arrival rate, the task profit obtained by all the blockchain nodes increases. However, the computing cost, the queuing cost, the cost of wireless transmission time and the cost of blockchain layer latency increase. Besides, the number of failed transactions also increases. Thus, the system obtains higher task profit at the cost of more intensive resource consumption and worse service quality. Taking all the performance indicators into consideration, $\gamma = 0.000\ 07$ is the best match with the setting of system resources in our simulation, and the best comprehensive reward is obtained when $\gamma = 0.000\ 07$.

The Greedy algorithm is the second best, and it tends to obtain better task profit obtained by all the blockchain nodes as shown in all the training results. Since the MAX-SINR algorithm selects the BS with the best channel quality to serve the transaction, it obtains the least cost of wireless transmission time. When $\gamma = 0.000\ 05$, the MAX-Credit algorithm can obtain similar overall reward to the Greedy algorithm, and its cost of infrastructure layer ltency is even smaller than that of the Greedy algorithm. As the MAX-Credit algorithm has the problem of load imbalance, the performance of the queuing cost and the cost of wireless transmission time increases dramatically with the increase of γ. When $\gamma > 0.000\ 06$, the MAX-Credit algorithm even cannot run properly.

Since the cost of blockchain layer latency highly depends on the real-time direct connection status which are not included in the state of the MDP, all the algorithm obtains the similar performance of this factor. The computing costs of all the algorithm are similar. This is because no matter which base station is selected as the service node, the computing costs of a given transaction is fixed.

4.2 Test phase

To evaluate the generalization capability of the proposed DDQN-TOA, the three DDQNs trained with $\gamma \in \{0.000\ 06, 0.000\ 07, 0.000\ 08\}$ are tested with γ

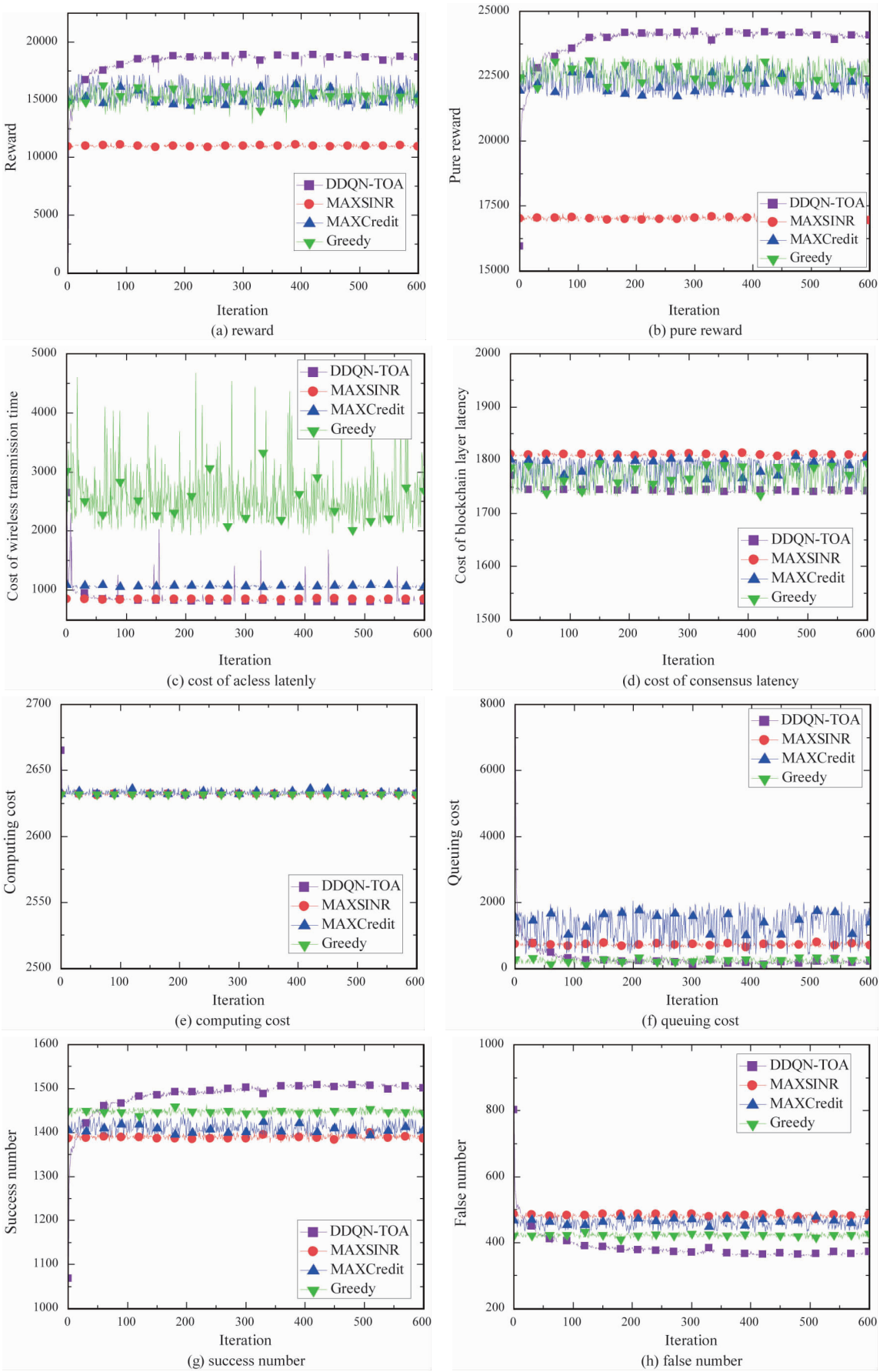

Fig. 3 Training phase ($\gamma = 0.000\ 05$)

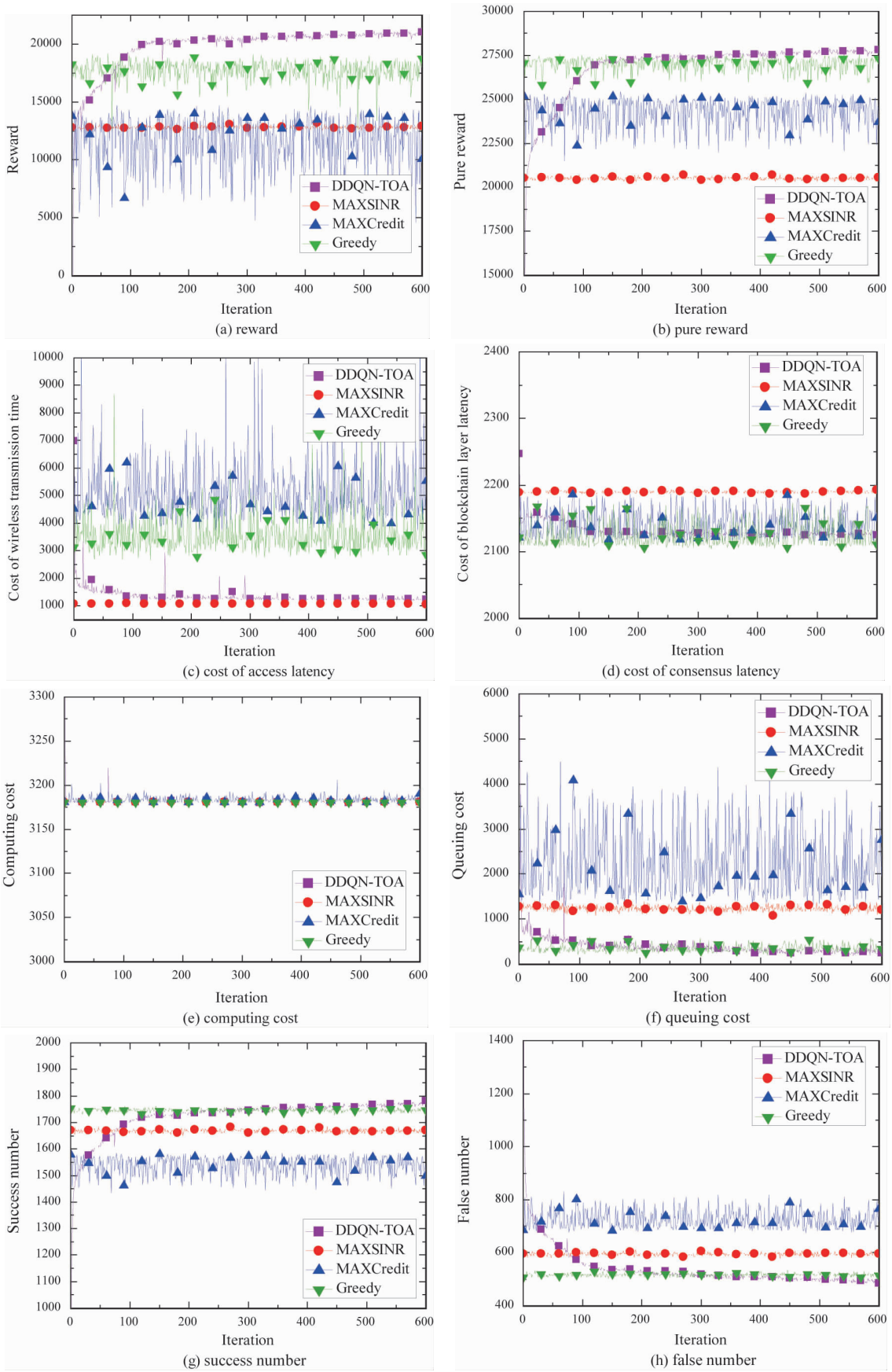

Fig. 4 Training phase ($\gamma = 0.000\ 06$)

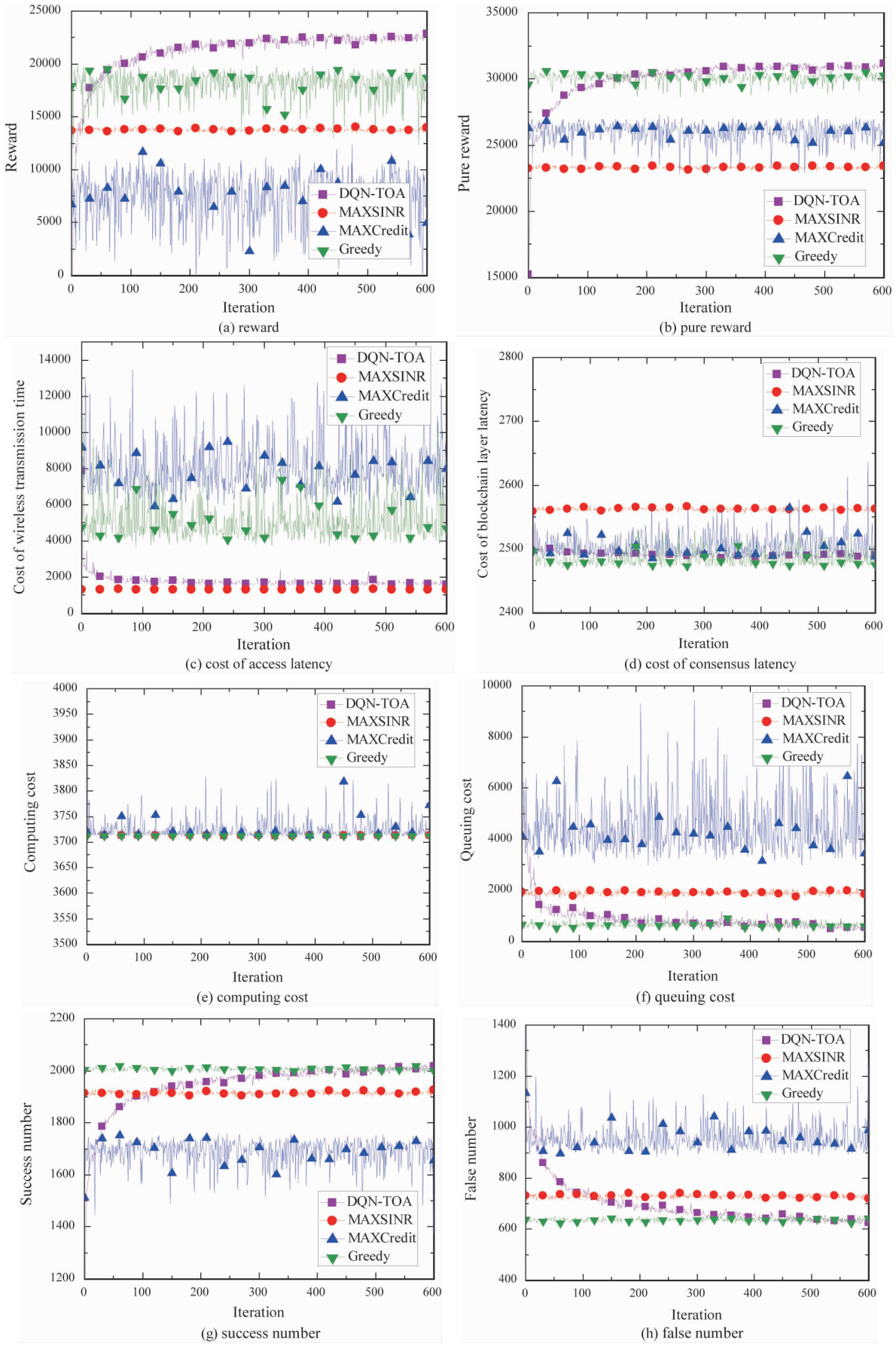

Fig. 5 Training phase ($\gamma = 0.000\ 07$)

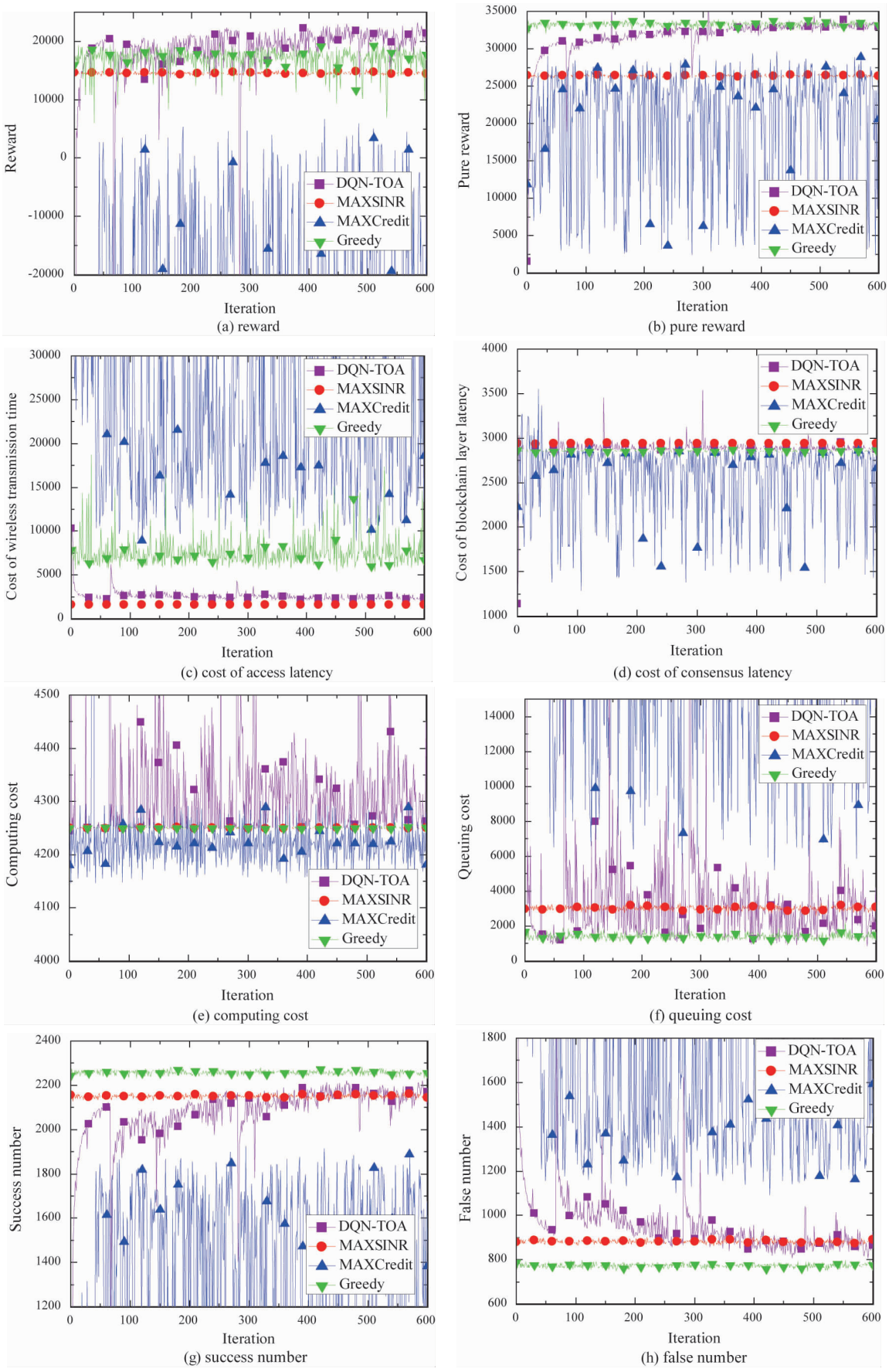

Fig. 6 Training phase ($\gamma = 0.000\ 08$)

varying from 0.000 03 to 0.000 08. For each test value of γ, 50 independent transactions flows are generated following the Poisson process within 300 time steps.

Test results given in Fig. 7 show that the DDQN trained with a higher transactions arrival rate can work properly when the transactions arrival rate of the test samples is lower. This benefits from the fact that the system simulation begins with an empty transactions queue, and the DDQN can experience the system state with light traffic in the training process. However, when the transactions arrival rate of the test samples is larger than that of the training samples, the DDQN has no chance to experience a heavy or overloaded traffic load. Thus, it cannot work properly.

Test results also confirms that $\gamma = 0.000\ 07$ is the best match with the setting of system resources in our simulation since the cumulative reward reaches the largest value when $\gamma = 0.000\ 07$. The DDQN trained with $\gamma = 0.000\ 07$ can obtain the best cumulative reward when $\gamma \leqslant 0.000\ 07$. As for the DDQN trained with $\gamma = 0.000\ 08$, it performs better than the Greedy algorithm when $\gamma = 0.000\ 06$, $0.000\ 07$, $0.000\ 08$. However, when γ is further reduced, the DDQN trained with $\gamma = 0.000\ 08$ performs worse than the Greedy algorithm.

The test experimental tells an important experience for intelligent network optimization, i. e. the network optimization agent should be trained at the system traffic load being full but not overloaded.

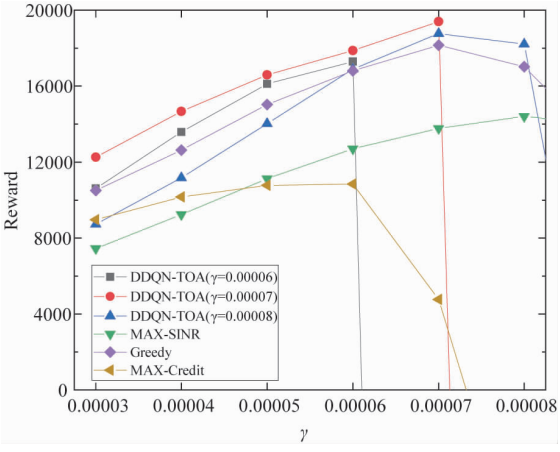

Fig. 7 Test phase

5 Conclusion

In this paper, the selection of serving BS and the allocation of wireless transmission resources in the MEC and blockchain-enhanced IoT is jointly optimized using the proposed DDQN-TOA algorithm. The real-time direct connection status of the BSs is modeled as a time-varying angular symmetric matrix, and the real-time communication latency among different BSs is approximately measured by the number of hops of the shortest path in the calculation of PBFT consensus latency. The training result shows that, under the same transaction arrival rate, the proposed DDQN-TOA can obtain the best long-term system utility, which is related to task reward, credit value, infrastructure layer delay, blockchain layer delay, and computing cost, among all the algorithms as MAX-SINR, MAX-Credit and greedy. Testing results show that the proposed DDQN-TOA has good generalization capability, and the DDQN trained with a higher transactions arrival rate can work properly when the transactions arrival rate of the test samples is lower.

References

[1] VAEZI M. Cellular, wide-area, and non-terrestrial IoT: a survey on 5G advances and the road toward 6G[J]. IEEE Communications Surveys and Tutorials, 2022, 24(2): 1117-1174.

[2] IQBAL W, ABBAS H, DANESHMAND M, et al. An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security [J]. IEEE Internet of Things Journal, 2020, 7 (10): 10250-10276.

[3] SADAWI A A, HASSAN M S, NDIAYE M. A survey on the integration of blockchain with IoT to enhance performance and eliminate challenges[J]. IEEE Access, 2021, 9: 54478-54497.

[4] ABDELLATIF K, ABDELMOUTTALIB C. Graph-based computing resource allocation for mobile blockchain[C]// 2018 6th International Conference on Wireless Networks and Mobile Communications. Marrakesh: IEEE, 2018: 1-4.

[5] ZUO Y, JIN S, ZHANG S. Computation offloading in untrusted MEC-aided mobile blockchain IoT systems [J]. IEEE Transactions on Wireless Communications, 2021, 20 (12): 8333-8347.

[6] CHEN S, CHEN J, MIAO Y. Deep reinforcement learning-based cloud-edge collaborative mobile computation offloading in industrial networks[J]. IEEE Transactions on Signal and Information Processing over Networks, 2022, 8: 364-375.

[7] CHEN S G, TANG B, WANG K. Twin delayed deep deterministic policy gradient-based intelligent computation offloading for IoT [J]. Digital Communications and Networks, 2020, doi: 10.1016/j.dcan.2022.06.008.

[8] XIE Z, WU R, HU M. Blockchain-enabled computing resource trading: a deep reinforcement learning approach[C] // 2020 IEEE Wireless Communications and Networking Conference. Seoul: IEEE, 2020: 1-8.

[9] LUONG N C, XIONG Z, WANG P, et al. Optimal auction for edge computing resource management in mobile block-

chain networks: a deep learning approach[C]//2018 IEEE International Conference on Communications. Kansas City: IEEE, 2018: 1-6.
[10] JIAO Y, WANG P, NIYATO D, et al. Social welfare maximization auction in edge computing resource allocation for mobile blockchain[C]//2018 IEEE International Conference on Communications. Kansas City: IEEE, 2018: 1-6.
[11] GU S, LUO X, GUO D. Joint chain-based service provisioning and request scheduling for blockchain powered edge computing[J]. IEEE Internet of Things Journal, 2021, 8(4): 2135-2149.
[12] LIU M, YU F R, TENG Y, et al. Computation offloading and content caching in wireless blockchain networks with mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2018, 67(11): 11008-11021.
[13] LI M, YU F R, SI P, et al. Resource optimization for delay-tolerant data in blockchain-enabled IoT with edge computing: a deep reinforcement learning approach[J]. IEEE Internet of Things Journal, 2020, 7(10): 9399-9412.
[14] GAO Y, WU W, NAN H, et al. Deep reinforcement learning based task scheduling in mobile blockchain for IoT applications[C]//2020 IEEE International Conference on Communications. Dublin: IEEE, 2020: 1-7.
[15] GAO Y, WU W, DONG J. et al. Deep reinforcement learning based node pairing scheme in edge-chain for IoT applications[C]//2020 IEEE Global Communications Conference. Taibei: IEEE, 2020: 1-6.
[16] NGUYEN D C, PATHIRANA P N, DING M, et al. Secure computation offloading in blockchain based IoT networks with deep reinforcement learning[J]. IEEE Transactions on Network Science and Engineering, 2021, 8(4): 3192-3208.
[17] LAHBIB A, TOUMI K, LAOUITI A, et al. Blockchain based trust management mechanism for IoT[C]//2019 IEEE Wireless Communications and Networking Conference. Marrakesh: IEEE, 2019: 1-8.
[18] YUN J, GOH Y, CHUNG J M. DQN-based optimization framework for secure sharded blockchain systems[J]. IEEE Internet of Things Journal, 2021, 8(2): 708-722.
[19] ZHANG H, LIU J, ZHAO H, et al. Blockchain-based trust management for Internet of vehicles[J]. IEEE Transactions on Emerging Topics in Computing, 2021, 9(3): 1397-1409.
[20] GAO Y, WU W, SI P, et al. B-rest: blockchain-enabled resource sharing and transactions in fog computing[J]. IEEE Wireless Communications, 2021, 28(2): 172-180.
[21] NGUYEN D C, PATHIRANA P N, DING M, et al. Secure computation offloading in blockchain based IoT networks with deep reinforcement learning[J]. IEEE Transactions on Network Science and Engineering, 2021, 8(4): 3192-3208.
[22] ZHANG J, BHUIYAN M Z A, YANG X, et al. Trustworthy target tracking with collaborative deep reinforcement learning in Edge AI-aided IoT[J]. IEEE Transactions on Industrial Informatics, 2022, 18(2): 1301-1309.
[23] GUO S, DAI Y, XU S, et al. Trusted cloud-edge network resource management: DRL-driven service function chain orchestration for IoT[J]. IEEE Internet of Things Journal, 2020, 7(7): 6010-6022.

YIN Yufeng, born in 1998. He received the B. E. degree in communication engineering from Beijing University of Technology in 2020. He is currently pursuing the M. S. degree at Faculty of Information Technology, Beijing University of Technology. His research interests include Internet of Things and blockchain.