Blockchain-enabled Charging Scheduling for Unmanned Vehicles in Smart Cities

Chuhao Qin, Pan Li, Jun Liu, Jianqi Liu

School of Information and Communication Engineering, University of Electronic Science and Technology of China, China
chrisitianq@gmail.com, lpuestc9@gmail.com, cnliujun93@gmail.com, jianqi.liu@outlook.com

Abstract

Recently, mobile crowd sensing (MCS) has been widely used in smart cities with the help of unmanned aerial vehicles (UAVs) and autonomous cars. To ensure long-term, long-distance sensing tasks of UAVs, mobile unmanned charging stations, called "*Carriers*", are scheduled to reach preset charging locations in a city to provide charging services. However, existing methods not only have difficulty in serving a large number of UAVs, but also face a big challenge of potential security vulnerabilities in the charging transactions. To address the above issues, this paper explores the blockchain technology to design an auction-based framework for scheduling charging services for UAVs while improving the security of charging transactions. In particular, considering the revenues of *Carriers* and the utilities of UAVs, we develop a new auction mechanism based on the emerging deep reinforcement learning technique to improve the auction performance. Experimental results demonstrate that our method can enhance charging transactions' security and optimize the system performance.

Keywords: Smart cities, Blockchain, Vehicles, Security, Auction

1 Introduction

Mobile crowd sensing (MCS), is an emerging paradigm for enabling smart cities, which avails assistance from unmanned vehicles, such as UAVs and unmanned cars. They are equipped with different types of high-precision sensors (e.g., GPS, gyroscope, and camera), and utilized for long-distance surveillance, delivery service, and data collection [1-3]. However, UAVs are constrained by their sensing range and lifetime because of the limited battery capacity, and it is also inadvisable to increase the battery size in each UAV as its weight will become another drawback. In order to perform a long-term, long-distance task, charging stations have been introduced to extend the working hours of UAVs in an MCS system [4-5]. Yet, cities are usually too crowded to set stationary stations on the road. Thus, unmanned cars can be used as mobile unmanned charging stations, namely, *Carriers*. After embarking from the master station, a *Carrier* heads toward preset charging location, waiting for UAVs to charge and continue working, as indicated in Figure 1.

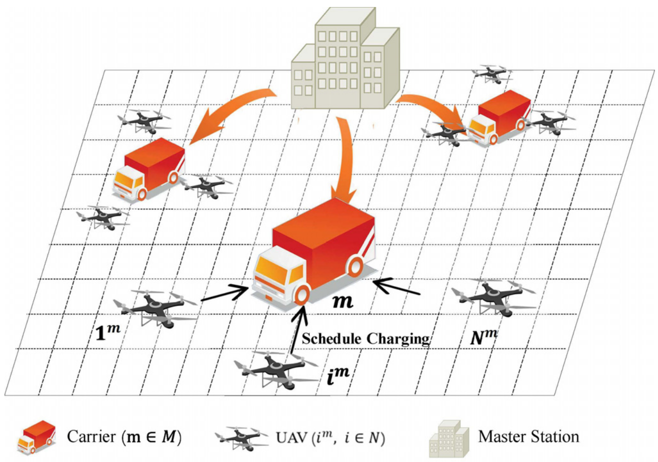


Figure 1. Charging scheduling with unmanned vehicles in an MCS system

The application of mobile unmanned charging stations faces a critical challenge. Since the *Carrier* is mobile, its size is comparably smaller than that of a fixed charging station. Therefore, the amount of energy resources available is limited, and hence, a relatively smaller number of UAVs can be charged simultaneously [5]. Recently, an economical method has been introduced to require mobile charging stations to schedule the charging time slot for UAVs, which is interpreted as an auction problem [6]. In an auction, buyers (or bidders, i.e., UAVs in the system) estimate their private and individual bids based on the emergency of their charging demand, and submit their bids to access services that are periodically auctioned by a seller (or auctioneer, i.e., *Carrier* in the considered setting). The UAV with the highest value of the bid wins the charging priority, and should pay an amount of cryptocurrency to the *Carrier*. The auction approach is remarkably feasible when each buyer is not assumed to know private vaues (or charging demands) of other buyers; and the seller is not aware of the actual values

*Corresponding Author: Pan Li; E-mail: lpuestc9@gmail.com
DOI: 10.3966/160792642021032202008

related to buyers before auctioning. This is the reason why the auction approach we take is especially suitable to solve the problem of time scheduling to UAVs in this non-cooperative, information-limited and distributed charging system.

However, in order to distribute the MCS system throughout cities, it is problematic to manage the currency in charging auctions composed of a large number of unmanned vehicles. Equally important, the currency transactions between *Carriers* and UAVs can be threatened by malicious behaviors such as data leakage and tampering due to being exposed to a trustless environment [7]. Recently, blockchain has emerged as a viable approach to record transactions in a distributed and verifiable manner by utilizing many technologies such as a distributed ledger, consensus mechanism, and smart contracts [8-11]. Inspired by blockchain technologies with advantages of security, transparency, and decentralization, we leverage it and build a trust, decentralized and automatic framework for *Carriers* to address the UAV charging problem.

The revenue-optimal auction mechanism is also considered. UAVs in a single-item auction tend to increase their bids to win the charging opportunity, but it is impermissible for them to pay more than its charging demand. Considering that neither *Carriers* nor blockchain developers operate blockchain sacrificially, we note that the utilities of UAVs (i.e., the difference between actual payment and bid in the auction) and the revenue of *Carriers* (i.e., payment received by the UAVs) should both be optimized. Here, the Myerson auction [12-13] is one of the most efficient revenue-optimal single-item auctions. This auction approach transforms the bid value monotonically, followed by the determination of the winner UAV and its payment. However, the main challenge of applying this approach is that it lacks any distributed prior knowledge or assumptions (i.e., distribution of UAVs). In the current environment of uncharged UAVs, primary factors affecting the charging demands, such as residual energy distribution, position distribution, and residual operation distribution, are desirable to be extracted in a system. Recently, deep learning has widely demonstrated that neural network structures can automatically learn important features from data and estimate complex nonlinear functions [14-16]. We utilize feature data with the help of neural networks under this circumstance.

Towards this end, based on the concept of the Myerson auction, we present a method that leverages deep reinforcement learning (DRL) [17] for auctioning between one *Carrier* and several UAVs, which we term as auctioning mechanism based on DRL (AM-DRL). Due to DRL's state-of-the-art performance while learning tasks with specific constraints, we believe it can approximate the monotonic distribution function without a prior knowledge. Then, we implement the overall framework by smart contracts to test cryptocurrency management and trading process. Finally, security analysis proves the security of the designed framework. Simulation results indicate *Carriers* obtain optimal revenues by using powerful deep neural networks (DNNs), while irrational bidders are punished for paying more than their charging demands.

The contributions of this paper are summarized as follows:

(1) To the best of our knowledge, this is the first work that leverages the blockchain technology to address the UAV charging problem in an MCS system.

(2) We design a secure, decentralized, and automatic auction-based framework for UAVs and mobile charging stations on the blockchain.

(3) A novel, truthful, and highly effective auction mechanism based on DRL, named AM-DRL, is proposed to optimize the revenues of mobile charging stations while improving the utilities of UAVs.

2 Designed Framework

In this section, we introduce the auction process of charging trading between one mobile charging station and a certain number of UAVs at first. Then, we describe the entire system, i.e., the system with the support of blockchain technologies. Table 1 illustrates the list of important notations used in this paper.

Table 1. Notations

Notation	Explanation
m, M	The index of a *Carrier*, the total number of *Carriers* in the networks
u, U	The index of a UAV, the total number of UAVs
i^m	The index of a UAV that requests charging to the *Carrier m*
N	The total number of UAVs that request charging to a *Carrier*
E, t, T	The amount of charge required, flight time with current battery residue, and remaining flight time that the total sensing tasks required
u_i, d_i, b_i	The utility, charging demand, bid of UAV i^m
x_i, p_i	The allocation rules and payment rules of UAV i^m
s_t, a_t, r_t	State, action and reward at period t
$\pi(\cdot), Q(\cdot), r(\cdot), L(\cdot)$	Policy function, Q function, reward function, loss function
$\boldsymbol{B}, \boldsymbol{B}_s$	The input bid smaples, s-th bid sample
ε, δ, T	The number of episodes, bid samples, periods

2.1 Auction-based Framework

The framework for charging scheduling based on auction (see Figure 2) mainly includes two entities: *Carrier* and UAVs.

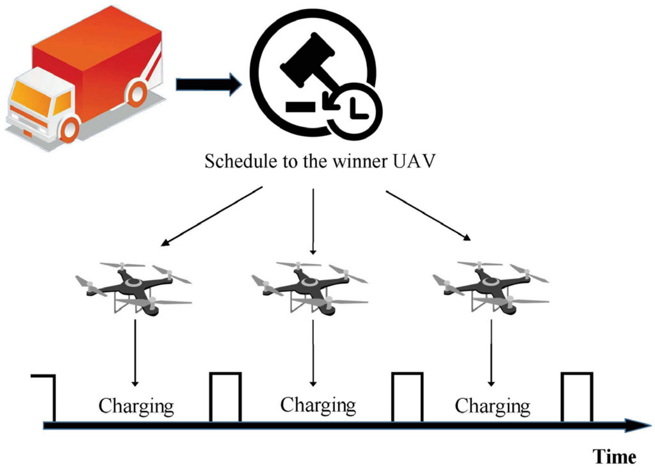


Figure 2. Auction-based scheduling process with one Carrier and several UAVs

Mobile charging stations—*Carrier*. The *Carrier* is instructed to arrive at a preset charging point in advance and wait, while charging all UAVs within a limited area in real time. This paper assumes that only one UAV can be charged by the *Carrier* in each time slot. For this reason, the *Carrier* uses an auction approach, allowing UAVs to compete for the energy resource. During an auction, a *Carrier*, as an auctioneer and a resource owner, receives all bids from UAVs, calculates allocation probabilities and payments, assigns a charging time slot for the winner who bids the largest amount corresponding to the highest allocation probability, informs the winner UAV regarding the actual amount of transaction coins it must pay, and then receives the payment. As the auction proceeds, the *Carrier* schedules UAVs continuously while accumulating the revenue. Finally, the revenue of the charging service is stored and used to pay the *Carriers*' operators and Blockchain developers in the master station.

Unmanned aerial vehicles. UAVs are the bidders in the considered setting. Each UAV searches for the nearest *Carrier* and requests scheduling for a charging slot once it detects a low battery level, and subsequently submits its bid privately according to the value of its charging demand. Note that a UAV with higher charging demand submits a higher amount of bid to the *Carrier*, and has higher probability to win the charging auction. The winner UAV selected by the *Carrier* can access the resource preferentially; it can be scheduled in the granularity of charging time slot till the next bidding. A UAV that fails the bidding game can choose to continue bidding for the same *Carrier*, or seek a charging opportunity from another carrier nearby till its battery charge is above its private threshold. Subsequently, it continues sensing tasks and operates the next charging iteration.

We assume that there are M *Carriers*, indexed by $\boldsymbol{M} = 1, 2, ..., m, \ldots, M$. There are U UAVs, denoted by $\boldsymbol{U} = 1, 2, ..., u, \ldots, U.$. If a UAV requests charging to the *Carrier* m, it is grouped into $\boldsymbol{N}^m = 1^m, 2^m, ..., i^m, \ldots, N^m$, where i denotes the index of the UAV requests charging to the *Carrier m*, N symbolizes the largest number of UAVs request charging to the *Carrier m*, $N \le U$. In this paper, we consider a short-sized charging scheduling with short-distance communications based on WLAN; and thus we neglect potential problems induced by the time delay between the UAVs' current locations and the preset charging position. Therefore, key properties that determine the charging demand of UAVs are categorized as the amount of charge required E, flight time with current battery residue t, and remaining flight time that the total sensing tasks required T, wherein T can represent the emergency of a UAV that has to finish the total sensing tasks in an MCS system, which plays a decisive role in the priority of charging scheduling.

2.2 Operation Details With Blockchain

As shown in Figure 3, the entire framework, which is supported by blockchain technologies, consists of a large number of unmanned vehicles. There are three types of nodes in the blockchain, namely, candidate, follower, and proposer.

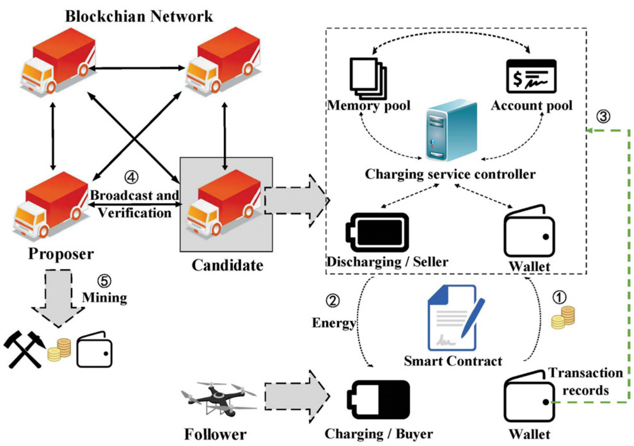


Figure 3. Charging scheduling framework supported by blockchain

Each *Carrier* behaves as a candidate, and a *Carrier* contains an account pool, a memory pool, and a charging service controller. It acts as a transaction server, by considering an auction process between a *Carrier* and a UAV as a transaction. Each candidate generates a transaction, including transaction records and a timestamp, among other features; encrypts the information; and subsequently constructs it into a block by solving hash problems. Authorized candidates work as validators to audit and verify the transaction records. Each UAV acts as follower, whose private account stores all transaction records and a corresponding wallet that manages transaction coins in the account. All followers can access, transmit, and receive the account data, but they are not permitted to manage transaction records. Herein, we use random pseudonyms as public keys of a node's wallet, named wallet addresses, to replace the true address of the wallet for security protection. The mapping relationships regarding wallets, the corresponding wallet addresses, and transaction coin accounts are stored in candidate's account pools. The memory pool stores all transaction records of candidate and follower nodes.

There is only one proposer selected from candidates based on a consensus mechanism, named as proof-of-work (PoW), which is similar to the conventional consensus in Bitcoins, which requires the calculation of a difficult hash value. The candidate with higher computing power has a greater opportunity of becoming the proposer, who has the bidding right of the blockchain, i.e., the proposer is permitted to transmit a block proposal to the network and receive extra mining revenue. Therefore, there are two competitions in the entire system. Competition for billing is booked among *Carriers*. UAVs, however, compete for the priority of energy charging. The total operation details of the system are listed as follows.

Registration and Smart Contracts Deployment. In the designed framework, all unmanned vehicles register on a trusted authority, such as Certificate Authority (CA), and become legitimate entities before setting off from the master station. Each *Carrier* and UAV with true identity gains public and private keys (PK^m, SK^m), $\forall m = 1, \ldots, M$, and (PK_u, SK_u), $\forall u = 1, \ldots, U$ from the authority respectively. During the initialization, both the blockchain rules and the trained auction algorithm network are written as smart contracts and deployed on the blockchain network for the automatic execution of the total circulation. More details about the auction algorithm are given in Section 3.

Auction of Energy. Each UAV detects the nearest *Carrier m*, and then binds to it by denoting i^m ($\forall i = 1, \ldots, N$) before it is validated by the *Carrier m*. Next, all UAVs download the latest data about the wallet from the memory pool, and upload bids to the *Carrier m*. After a bidding competition, the winner UAV i^m transfers transaction coins from its wallet to the obtained wallet address. The *Carrier m* validates the payment by acquiring the last blockchain from the memory pool before allocating the charging time for the UAV i^m. Transaction records are generated by UAV i^m, and validated by the *Carrier m*.

Building Blocks and PoW Consensus. *Carriers* in the network collect local transaction records and store them in the memory pool within a certain period. They encrypt and sign these records for the purpose of guaranteeing authenticity and accuracy, after which they construct them into blocks that have a similar Merkle-tree structure to Bitcoin. Each candidate calculates the hash value of its block based on a random nonce value, timestamp, previous block hash value, and so on, trying to find a correct *nonce* to solve the PoW problem. This problem is considered as solved when a calculated answer value is smaller than *Difficulty*, which can govern the speed of finding solutions in the system. The fastest node in the candidate group, which solves the cryptographic puzzle first, becomes the proposer of the current consensus process. The proposer broadcasts block data to other authorized candidates for validation and audit. After auditing them with a signature, comparing them, and sending feedback, the proposer analyzes the acquired replies from authorized candidates. Consensus is considered to be reached once the result is successfully validated; the newly generated block is appended to candidates' local copy of the blockchain in a linear and chronological order. Finally, the system calculates the corresponding rewards to the proposer.

3 Proposed Method

In this section, we introduce the proposed method, including auction mechanism, DRL networks, AM-DRL, and the overall framework.

3.1 Auction Mechanism

A single-item auction mechanism, based on DRL, is proposed to maximize the *Carrier*'s revenue while improving optimal utilities for UAVs. We use the second-price auction (SPA) as a baseline to design the auction mechanism. In SPA, the auctioneer receives an amount of bids from all bidders, and determines that the highest bidder is the winner; and the winner only pays the amount that is equal to the second highest bid. We consider the situation where UAVs $\boldsymbol{N}^m$ compete for one *Carrier m*. It is guaranteed that the winner UAV must have the highest charging demand d_i based on SPA, $\forall i = 1, \ldots, N$. Additionally, each UAV i^m has the charge required E_i, the current flight time remained t_i, and the total flight time remained T_i. If E_i is larger, or t_i is smaller, UAV i^m is willing to pay a larger amount for the charging service, evolving into a larger charging demand. The higher T_i also enables UAV i^m to give a higher d_i, so as to complete the whole sensing tasks. Therefore, d_i can be expressed as $d_i = E_i * T_i / t_i$.

Each UAV i^m reports its demand d_i by bidding an amount b_i to the *Carrier m*; an auction determines the allocation rules of items to UAVs and charges a payment to them. We denote an auction as the allocation rules x_i and the payment rules p_i, $\forall i = 1, \ldots, N$. The allocation rule x is used to determine which UAV should be scheduled, and the payment rules p decide the actual payment of the winner UAV. Thus, the auction revenue of a *Carrier* with N UAVs can be expressed as follows:

$$\text{Auction Revenue} = \sum_{i=1}^{N} p_i * x_i(b_i) \quad \textbf{(1)}$$

However, an adverse case may exist wherein a UAV participating in an auction reports a bid untruthfully, to maximize its utility and equip it with more competitive auctioning power, which destabilizes the system [12-15], [18]. Given bids $\boldsymbol{b} = (b_1, \ldots, b_N)$, the auction computes $x(\boldsymbol{b})$ and $p(\boldsymbol{b})$. As a result, the utility of the UAV i^m, regarded as the private profile for a bidder, can be calculated as $u_i = d_i x_i - p_i,\ \forall i = 1, \ldots, N$. Thus, bidders are strategic and seek to maximize their utilities by reporting bids that are different from their demands. To ensure truthful action of bidders, two characteristics of auction, i.e., the dominant strategy incentive compatibility (DSIC) and individual rationality (IR), are considered in this paper, and they are guaranteed by the Myerson auction. DSIC is defined such that each bidder's utility is maximized by reporting truthfully regardless of what other bidders report. Specifically, for each bidder i^m, every demand d_i, every bid b_i, and every possible report of the other bidders bid b_{-i}, $u_i(d_i, b_{-i}) \geq u_i(b_i, b_{-i})$. An auction is IR if each bidder achieves a non-zero utility: for all i^m, d_i and b_i, we have $d_i \geq b_i$, i.e., no bidder pays more than its charging demand.

The Myerson auction satisfies DSIC and IR, encouraging each UAV to bid truthfully to maximize its own utility in an energy auction [12]. In addition, the Myerson auction guarantees auctioneer's revenue optimality. Thus, to maximize the *Auction Revenue* shown in (1), we leverage the monotonically non-decreasing transform functions, denoted as φ_i, $\forall i = 1, \ldots, N$, from Myerson auction. As presented in [12], input bids b_i, $\forall i = 1, \ldots, N$ is converted into $\overline{b_l} = \varphi_i(b_i),\ \forall i = 1, \ldots, N$. Then, SPA with zero reserve price (SPA-0) is used to calculate allocation rules x and payment rules p as per the following theorem [12].

Theorem 1: For any set of strictly monotonically increasing functions $\phi_1, \ldots, \phi_N : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, an auction which is defined by allocation rule x_i and payment rule $p_i = \phi_i^{-1}(\max_{j \neq i}(\overline{b_i}))$.

With this theorem, the proposed Myerson auction-based algorithm can not only maximize the *Carriers* revenue while improving expected utility of each UAV, but also satisfy sufficient conditions for DSIC and IR. Nevertheless, it is hard to obtain ϕ to convert b into $\overline{b}$; and thus we propose to leverage a recent DRL method to approximate the transform function $\phi(\boldsymbol{b})$ Previous research results [6, 14] have shown that deep learning with neural networks can estimate the function and get the optimal revenue. Our method with DRL leverages the action to the state and reward feedback, being updated as bids sets input and episode iterates. Note that the proposed method is only in consideration of one *Carrier* and several UAVs, but it is applicable to the entire system, which can be proved by the numerical results in Section 4.

3.2 DRL Networks

The proposed DRL-based method for auction mechanism was used to calculate the *Auction Revenue*. Given a state and a set of possible actions to choose from, the goal is to find a control policy $\pi(\boldsymbol{s})$ that maximizes the accumulated reward. First, based on the virtual valuation function $\phi(\boldsymbol{b})$ we define the state, action, and reward at current period t $(\forall t = 1, \ldots, T)$ as follows:

(1) State: $s_t = (s_{1,t}, \ldots, s_{N,T})$ at period t is the current result of $\phi(\boldsymbol{b})$, which means that the function is calculated once it enters the state in the next timeslot $t+1$. Thus, the initial state s_1 is equal to the input bids $\boldsymbol{b} = (b_1, \ldots, b_N)$, and the final state s_T refers to $\overline{\boldsymbol{b}} = \overline{b_1}, \ldots, \overline{b_N}$.

(2) Action: $a_t = (a_{1,t}, \ldots, a_{N,t})$ is equivalent to function $\phi(\boldsymbol{b})$. After the execution of the action, the old state s_t would change to the new state s_{t+1}. Suppose that each action represents a component of the transform function denoted as ϕ_t; then, we express the formula $\overline{b_1} = \phi_T(b_i) = \phi_T(\phi_{T-1}(\ldots\phi_1(b_i))),\quad \forall i = 1, \ldots, N$. Since action is undertaken continuously, the method is a continuous control task.

(3) Reward: The reward r_t is defined as

$$r_t = \sum_{i=1}^{N} p_{i,t} * x_i(s_{i,t+1}), \quad \textbf{(2)}$$

where $p_{i,t}$ is the actual payment of winner bidder i_m at period t, and $x_i(s_{i,t+1})$ is the allocation determined by the next state. Since the reward is equivalent to the revenue of the *Carrier*, maximizing the cumulative reward is equivalent to maximizing the *Auction revenue*.

Note that we can decide the expression of $p_{i,t}$ and $x_i(s_{i,t+1})$ based on Theorem 1. We convert the input vector of the allocation rules (x_i) into a probability vector; the result of x_i is output in a probabilistic manner. At each period, the allocation rules assign the highest wining probability to the highest bidder, whose state is in the next timeslot. The allocation of the winner i^m can be calculated as follows:

$$x_i(s_{i,t+1}) = \frac{e^{rs_{i,t+1}}}{\sum_{j=1}^{N} e^{rs_{j,t+1}}}, \quad \textbf{(3)}$$

where constant parameter r determines the quality of approximation, deeply relevant to the *Auction Revenue*. For the payment rules p_i for the winner i^m, we refer to the formula in Theorem 1 and a *ReLU* activation unit to express the function of payment with the new state $s_{i,t+1}$:

$$p_{i,,t} = \phi_t^{-1}(ReLU(\max_{j \neq 1} s_{i,t+1})), \quad \textbf{(4)}$$

where $ReLU(b) = \max(b, 0)$ ensures that the payment is non-negative; the inverse transform function is expressed as $\phi_t^{-1} = \phi_1^{-1}(...(\phi_t^{-1}(\cdot)))$, $t \geq 2$, or $\phi_t^{-1} = \phi_1^{-1}(\cdot)$, $t = 1$. Note that the transform ϕ_t and the inverse transform ϕ_t^{-1} at period t are defined as $\phi_t(y) = \psi_t y + \sigma_t$ and $\phi_t^{-1}(z) = (\psi_t)^{-1}(z - \sigma_t)$, whose parameters ψ_t and σ_t are calculated by the action a_t.

Since we are dealing with a continuous control task, a state-of-the-art actor-critic method, called DDPG, was selected for the operation [19-20]. The detailed training process of the network is presented as Algorithm 1. The input contains a group of bids sets. Following the calculation of allocation rules and payment rules, the algorithm minimizes loss function to update weights $\theta^{Q'}$ and $\theta^{\pi'}$ in target networks.

Algorithm 1. Deep Reinforcement Learning Training

Input: $N, k, \boldsymbol{B}_s = (B_1, ..., B_N)$ in the entire input samples $\boldsymbol{B} = [\boldsymbol{B}_1, ..., \boldsymbol{B}_\delta]$

Output: Optimized weights θ^Q, θ^π, $\theta^{Q'}$ and $\theta^{\pi'}$

1. Initialize actor network $\pi(s \mid \theta^\pi)$, critic network $Q(s, a \mid \theta^Q)$ with weights θ^Q, θ^π, and their two target networks $Q'(\cdot)$, $\pi'(\cdot)$ with parameters $\theta^{\pi'} := \theta^\pi, \theta^{Q'} := \theta^Q$;
2. Initialize replay buffer and exploration noise *N*;
3. **for** *episode*:=1,..., ε **do**
4.   Reset the environment and obtain the initial state s_1;
5.   **for** *Period t*:=1,..., T **do**
6. $a_t = \in \pi(s_t) + 1(1 - \in)N$, $\in$ decays over time;
7.   Execute a_t to acquire s_{t+1} by calculating:
8. $s_{i,t+1} = \phi_t(s_{i,t})$;
9.   Calculate r_t, x_i and $p_{i,t}$ according to equations (2), (3), and (4);
10.   **end**
11.   Store transition sample (s_t, a_t, r_t, s_{t+1}) into buffer;
12.   Sample a random minibatch of *H* samples (s_j, a_j, r_j, s_{j+1}) from buffer;
13. $y_j = r_j + rQ'(s_{j+1}, \pi'(s_{j+1} + \theta^{\pi'}) \mid \theta^{Q'})$;
14.   Update θ^Q by minimizing the loss function:
15. $L(\theta^Q) = \frac{1}{H}\sum_{j=1}^{H}[y_j - Q(s_j, a_j)]$;
16.   Update θ^π by using the gradient:
17. $\nabla_{\theta^\pi} J \approx \frac{1}{H}\sum_{j=1}^{H} \nabla_a Q(s, a \mid \theta^Q)|_{s=s_j, a=\pi(s_j)} \cdot \nabla_{\theta^\pi} J(s \mid \theta^\pi)|_{s=s_j}$;
18.   Update $\theta^{Q'}$ and $\theta^{\pi'}$:
19. $\theta^{\pi'} = \xi\theta^\pi + (1 - \xi)\theta^{\pi'}$;
20. $\theta^{Q'} = \xi\theta^Q + (1 - \xi)\theta^{Q'}$;
21. **end**

First, the algorithm initializes 4 DNNs, which serve as an actor network $\pi(s \mid \theta^\pi)$, a critic network $Q(s, a \mid \theta^Q)$, and two target networks with parameters $\theta^{\pi'} : \theta^\pi, \theta^{Q'} : \theta^Q$ (Lines 1). The fundamental idea is to maintain an actor function to derive the best action from an initial state and a critic network to train and evaluate the network based on realistic and estimated Q value.

After initializing the environment and obtaining the initial state (Line 4), the exploration process is implemented (Lines 5-10). Actions are derived from both the output of current actor network and adjustable parameter $\in$ (Line 6). $\in$ can tradeoff exploration and exploitation by determining the probability of adding a random noise *N*to the actor network. In this paper, exploitation is implemented over the first 200 episodes. $\in$ is initialized to 1 and decays with a rate of 0.9995 over episodes; *N* follows a function by adding a random distribution between -0.2

and 0.2. Then, the next state $s_{i,t+1}$, allocation rule x_i, and payment rule $p_{i,t}$ of the winner i^m are calculated; then, the reward is obtained through (2).

Following the exploration, the algorithm focuses on how to update the neural networks (Lines 12-20). In DDPG, experience-driven replay and target network are used to ad- dress the issue of system instability. DDPG uses a mini-batch from a buffer, which contains abundant state transmission samples. Besides, it uses an additional target network to estimate the target value y_t. The critic network is subsequently updated through minimizing a loss function $L(\theta^Q)$, as [21-22]:

$$L(\theta^Q) = \mathbb{E}[y_t - Q](s_t, a_t \mid \theta^Q)], \qquad \textbf{(5)}$$

$$y_t = r(s_t, a_t) + rQ(s_{t+1}, \pi(s_{t+1} \mid \theta^\pi) \mid \theta^Q] \qquad \textbf{(6)}$$

whereas the actor network can then be updated by using the gradient as:

$$\nabla_{\theta^\pi} J \approx \mathbb{E}[\nabla_a Q(s, a \mid \theta^Q)|_{s=s_t, a=\pi(s_t)} \cdot \nabla_{\theta^\pi} \pi(s \mid \theta^\pi)|_{s=s_t}] \qquad \textbf{(7)}$$

In our design, we sample H = 64 groups of transitions as mini-batches in a replay buffer with z size of 10^6 and set the discount factor $r = 0.99$. After updating the actor and critic networks, the weights of target networks, $\theta^{\pi'}$ and $\theta^{Q'}$, are then slowly updated with controlled updating rate ξ and the original networks weights. Specifically, we utilize a 2-layer fully-connected feedforward neural network to serve as the actor network, including 400 and 300 neurons in the first and second layers, respectively. For the critic network, on the other hand, we use a 3-layer neural network with 400, 100, and 300 neurons in the first, second, and third layers. respectively. Additionally, both networks had the *ReLU* function and $\tan h(\cdot)$ for activation in addition to the L_2 weight decay to prevent overfitting. In addition, we set the initial ξ as 0.001, and the learning rate of actor network and critic network as 0.0001 and 0.001 respectively.

3.3 AM-DRL

The overall auction mechanism of charging scheduling in DRL computation is summarized in Algorithm 2, namely, AM-DRL. When a *Carrier m* has a charging vacancy, the auction is initiated. Each UAV computes its private charging demand d_i and submits its bid b_i to the *Carrier* by determining the value of a_i. a_i can be a higher number when the UAV i^m intends to request a larger amount of energy. If each UAV reports low charging demand to the *Carrier* and the payment output is decreased to 0, the *Carrier* assigns no time slot to UAVs. Otherwise, the corresponding payment probabilities are calculated through a period T of exploration and exploitation in the pre-trained network. (Lines 3-8) Finally, the *Carrier* assigns the payment to the winner UAV i^m with the highest x_i, and allocates it charging time. (Lines 9-13) The UAV accepts the admission, reaches the *Carrier*, and occupies it for the duration of the slot. After the winner UAV leaves, the next iteration of auction begins if the *Carrier* is unoccupied.

Algorithm 2. AM-DRL

Input: $N, k,\ \boldsymbol{b}_i = (b_1, ..., b_N)$

Output: allocation $\boldsymbol{x} = (x_1, ..., x_N)$, payment $\boldsymbol{p} = (p_1, ..., p_N)$

1. **while** The *Carrier m has a charging vacancy* **do**
2. Resert the environment and obtain the initial state s_1;
3. **for** *Period* t:=1,..., T **do**
4. $a_t = \pi(s_t)$;
5. Execute a_t to acquire s_{t+1} by calculating:
6. $s_{i,t+1} = \phi_t(s_{i,t})$;
7. Calculate r_t, x_i and $p_{i,t}$ according to equations (2), (3), and (4);
8. **end**
9. $x_i(s_{i,T+1}) = \dfrac{e^{ks_{i,T+1}}}{\sum_{J=1}^{n} e^{ks_{i,T+1}}}$;
10. $p_i = \phi_T^{-1}(ReLU(\max_{j \neq 1} s_{i,T+1}))$; ;
11. **if** $p_i \neq 0$ **then**;
12. Determine the winner and calculate payment;
13. Allocate charging service to the winner;
14. **end**;
15. **end**

3.4 The Overall Framework

In this part, based on the smart contracts of Ethereum, we implement the overall framework integrating AM-DRL and aforementioned desinged system. Figure 4 indicates the state transition of *Carriers* defined in the CPoC (Charging Process of the *Carrier*) smart contract. It includes three states: “Init”, “Mining” and “Charging”. Each state transition requires the role R to invoke the interface T (expressed as $R \to T$), where the representation of the R are C for *Carrier* and V for UAVs. The details of state transition is as follows.

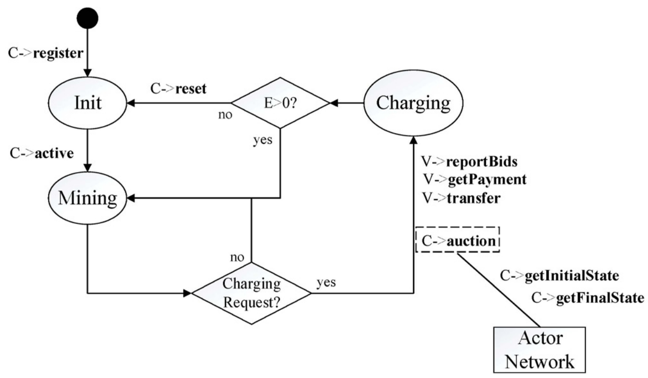


Figure 4. State Transition Diagram of the *Carrier* in the CPoC smart contract

After registration in the master station, the *Carrier* turns its state into “Init” and enters the operation. Once it is charged and activated through performing active interface, its state changes as “Mining”, where it can calculate the hash value, build blocks, receive mining revenues, and validate other transactions. During the mining process, the *Carrier* monitors whether it is received charging requests from UAVs. If it is requested by at least one UAV, all UAVs have to invoke the “reportBids” interface and send the bids to the *Carrier*. Following the winner selection and payment calculation in the auction, the winner UAV invokes “getPayment” and transfer the amount to the address of the *Carrier*. Afterwards, the *Carrier*’s state turns into “Charging” where it has to charge the winner UAV while performing the mining. After charging, the supervision of residual energy E of the *Carrier* determines whether it can continue working, or reset to “Init” state.

Note that “auction” interface includes complicated actor network with trained parameters. We use “getInitialState” and “getFinalState” interfaces to pass parameters outside the contract, simulating the state changes in AM-DRL. More details about performance evaluation of the contract will be given in the next section.

4 Performance Evaluation

In this section, we first analyze the security about our blockchain-enabled system, and then present the numerical results of auction mechanism with DRL.

4.1 Security Analysis

Through blockchain technology, the proposed system framework can protect against many traditional attacks and malicious behaviors that may threaten the networks security.

Impersonators Prevention. A malicious participant may substitute an unmanned vehicle and act as a node in the net- works. With the Certificate Authority, all unmanned vehicles are registered with keys before setting off to from the master station. It is difficult for an impersonator to forge a private key and be involved in transactions that are profitable to them multiple times.

Auction Authenticity. A malicious UAV cannot refuse to pay the *Carrier* and steal the power after being selected as the winner. It is impossible to compromise auction rules, given that the whole auction process is written on smart contracts and deployed on the blockchain at the beginning of tasks. Besides, the proposed auction mechanism is proved to be truthful and rational in the following subsection, which prevents the malicious UAV from misrepresenting its bid.

Transaction Unforgeability. All transactions are recorded in the blockchain, and are validated fairly by the authorized candidates in the networks with digital signature. Since the transaction data is blocked through a highly difficult hash computation with the help of PoW consensus, the attackers would face a huge challenge to tamper the transaction and corrupt the designed system.

4.2 Numerical Results

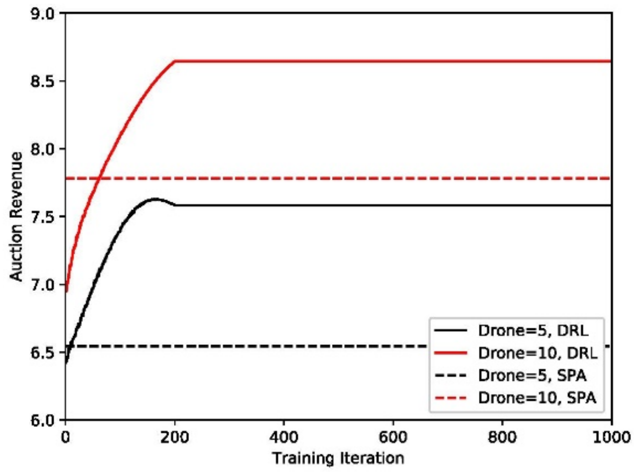
We evaluate the proposed method by simulations implemented in Python 3.6 and TensorFlow 1.10 with 2 NVIDA TITAN XP GPUs. As shown in Table 2, we train the method for 1000 episodes and 10 periods, which is conducted with 100,000 generated data sets. In this experiment, 70% of data sets are used for training, and the rest are used for testing the networks. Suppose that the value of time consumed and energy consumption among UAVs are uniformly distributed. Each UAV calculates the bid based on its private charging demand; and thus we generat the bids set in the range of $0-10$ as the input of network training. We also set 5 or 10 UAVs and the parameter $r = 3$ for approximation quality in the consideration of smoothness in the allocation network and the maximum number of UAVs that can be charged without being discharged. SPA-0 is used as a baseline method to compare with our valuation results.

Table 2. System Parameterws

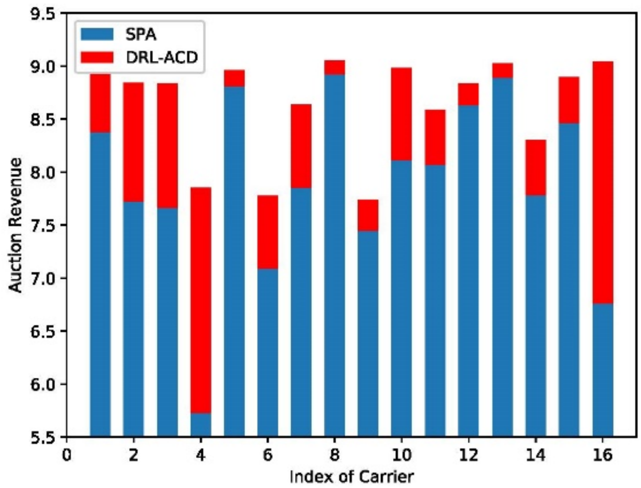
Parameter	Value	Parameter	Value
N	5, 10	r	3
ε	1000	S	100,000
T	10	M	16

Figure 5 shows the optimal effect and high adaptability of our algorithm; the numerical results are presented in Table 3 and Figure 5(a) illustrates the comparison of the *Auction Revenue* results generated by DRL training and SPA-0 auctions. The revenues shown in both graphs are enhanced over the first 200 episodes because of the exploration and exploitation, before achieving stability till the end. The revenue gap between the proposed auction and SPA-0 is around 15.85% when the number of UAVs is 5, whereas the disparity is 11.09% with 10 UAVs. Note that the allocation probability of the high bid can increase as the number of UAVs participating in the auction becomes larger; and thus the second highest bid value in the auction can also be increased. Considering the fact that the *Auction Revenue* is equivalent to the payment, the value increases along with the payment of the winner UAV. Consequently, when UAVs are set from 5 to 10, the results of SPA-0 increase from 6.5460 to 7.7813, while an upward from 7.5836 to 8.6445 belongs to that of proposed auction. From Figure 5(b), we show the comparison between the testing results among 16 auctions (16 *Carriers* in the total networks) and the network results using SPA-0 when UAV is 10. We run AM-DRL by inputting 16 times and receive revenues of different *Carriers*, as shown in Table 4. For each *Carrier*, it can be witnessed that the revenue disparity between AM-DRL and SPA-0 is disproportionally high. The increasing percentage from SPA to proposed method, for example, is 1.47% in the *Carrier* 13 albeit 33.77% in the *Carrier* 16. This happens because the actual charging demand of each UAV restricts the maximum payment to the *Carrier*. Generally, the experimental result not only verifies that AM-DRL has a larger value than SPA-0, but also proves that AM-DRL is applicable to the entire system in charging scheduling.

AM-DRL is proposed in the condition of satisfying DSIC and IR; UAVs in each auction act truthfully, or they will be given penalty. In Figure 6, the results show a UAV fail to win the auction when it bids up to 0.6-2.0 times larger than its charging demand 9.3763. Through the graph, both bid and payment values are monotonic increasing. Suppose that the fake bid is reported when untruthful bidding coefficient is larger 1, it can be seen that the higher fake bid leads to the larger advantage to win the auction, escalating into the higher charging payment for the winner UAV. For instance, Table 5 illustrates that, if coefficient is 2.0 and the reporting bid is as twice as the bidder's true charging demand, the actual payment clearly outnumbers the demand with a significant surplus of 5.8654. On the other hand, if the



(a) Auction revenue statistics comparison



(b) Auction revenues of *Carriers*

Figure 5. Auction revenues of AM-DRL

Table 3. Revenue Optimization

Number of UAVs	5	10
SPA	6.5460	7.7813
DRL training	7.5836	8.6445

Table 4. AM-DRL versus SPA in Revenue of *Carriers*

Index of *Carrier*	1	4	8	13	16
SPA	8.3767	5.7268	8.9221	8.8946	6.7592
AM-DRL	9.0527	7.8510	9.0546	9.0252	9.0417

bidder intends to reduce the payment burden by bidding low price (e.g., coefficient is less than 1), it will lose the opportunity to win the auction when the fake bid is smaller than the second highest bid when coefficient is 0.8, i.e., the payment is smaller than second highest bid. This proves that UAVs avoid getiing a payment loss or losing the game when they compete for charging eligibility in our proposed auction mechanism. Moreover, the figure shows the amount of paymnet cannot be larger than that of bid when coefficient is set to 0.6-2.0. As a consequence, no

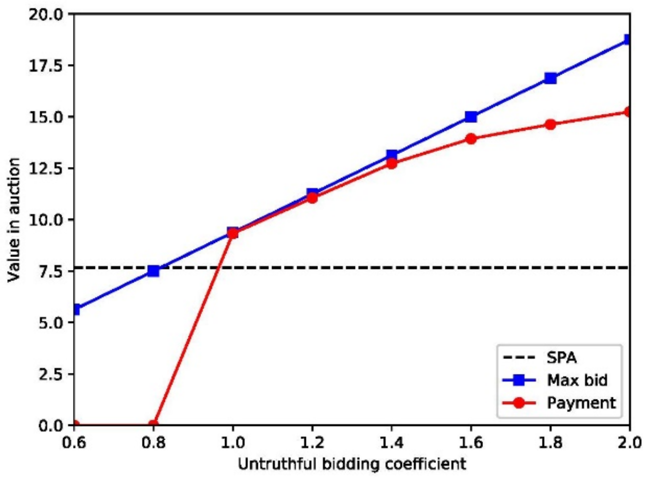


Figure 6. Bid and payment changes by untruthful bidding coefficient

Table 5. Comparison with Untrustful Bidding Coefficient

Coefficient	1.2	1.4	1.6	1.8	2.0
SPA	7.631	7.631	7.631	7.631	7.631
Payment	11.0504	12.7266	13.9325	14.6262	15.2417
Max bid	11.2516	13.1268	15.0020	16.8773	18.7526

bidder is asked to pay more than its charging demand in the condition that the auction mechanism is truthful, which guarantees the individual rationality. This experiment shows that the proposed method can ensure each UAV bids truthfully when it achieves the optimization of its own utility.

Finally, in order to test all functionality in the framework, we leverage the programming, Solidity, to program the CPoC smart contract, and deploy it on the test net of Ethereum blockchain, "Rinkeby". We generate several accounts on "Rinkeby" to simulate different roles, i.e, the *Carrier* and UAVs. As the cryptocurrency of Ethereum without real value, "Ether" is leveraged to invoke the interfaces and serve as the transaction coins stored in all accounts. Apart from this, we test and validate all functionality of interfaces, and the result demonstrate our framework accomplish the total operation. As for the performance analysis of our experiment study, we refer to the complexity of each interface in the CPoC smart contract, since the larger complex one force the role to execute program defined in the interface by consuming higher electricity power, which requires more transaction fee. Hence, the gas consumption, a typical representative of transaction fee cost in Ethereum, is considered to evaluate the experimental performance as shown in Figure 7. The results illustrate each UAV tends to require less amount of gas in the entire operation than the *Carrier*. Because in most cases, the *Carrier* is the greatest beneficiary to gain the payment of charging service. Nevertheless, our implementation for the *Carrier* still exists space to develop, considering the operation of UAVs requesting multiple *Carriers*.

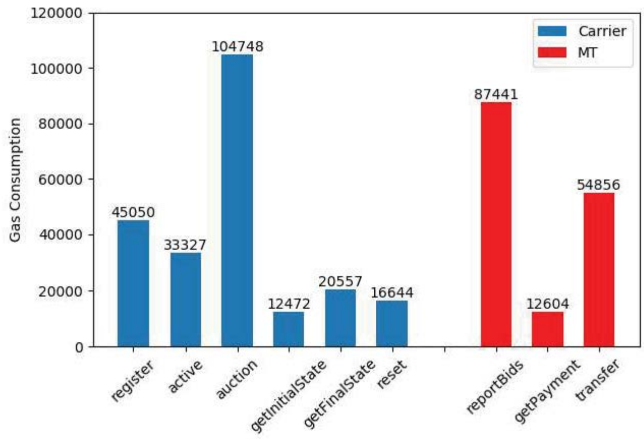


Figure 7. The gas consumption of interfaces in CPoC smart contract

5 Conclusion

In this paper, we report the design of a blockchain-supported charging scheduling framework using the proposed AM-DRL (auction mechanism based on DRL). In this framework, an unmanned charging station that schedule the charging of UAVs was transformed into auction problem in the entire system, wherein each UAV bids its private charging demand; the *Carrier* provides charging service for UAVs by scheduling them according to their demands. Blockchain technology is introduced, whereby *Carriers* and UAVs can automatically trade based on smart contracts. Besides, we prove that this technology improves the system security and provides a platform to manage cryptocurrency. The proposed AM- DRL, based on a state-of-the-art actor-critic method, DDPG, optimizes the *Carriers*' revenues in terms of dominant strategy incentive compatibility and individual rationality, enabling each UAV to bid individually-rationally and maximize its utility. For the further work, we plan to develop the CPoC smart contract and enhance the coordination of the bidding process for unmanned vehicles.

References

[1] N. Nigam, S. Bieniawski, I. Kroo, J. Vian, Control of multiple uavs for persistent surveillance: Algorithm and flight test results, *IEEE Transactions on Control Systems Technology*, Vol. 20, No. 5, pp. 1236-1251, September, 2012.

[2] K. Kuru, D. Ansell, W. Khan, H. Yetgin, Analysis and optimization of unmanned aerial vehicle swarms in logistics: An intelligent delivery platform, *IEEE Access*, Vol. 7, pp. 15804-15831, January, 2019.

[3] C. Dai, X. Liu, W. Chen, C. Lai, A low-latency object detection algorithm for the edge devices of iov systems, *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 10, pp. 11169-11178, October, 2020.

[4] M. Simic, C. Bil, V. Vojisavljevic, Investigation in wireless

power transmission for uav charging, *Procedia Computer Science*, Vol. 60, pp. 1846-1855, 2015.
[5] B. Zhang, C. H. Liu, J. Tang, Z. Xu, J. Ma, W. Wang, Learning- based energy-efficient data collection by unmanned vehicles in smart cities, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 4, pp. 1666-1676, April, 2018.
[6] M. Shin, J. Kim, M. Levorato, Auction-based charging scheduling with deep learning framework for multi-drone networks, *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 5, pp. 4235-4248, May, 2019.
[7] S. Garg, A. Singh, S. Batra, N. Kumar, L. T. Yang, Uav-empowered edge computing environment for cyber-threat detection in smart vehicles, *IEEE Network*, Vol. 32, No. 3, pp. 42-51, May/June, 2018.
[8] L. Lau, Z. Li, S. Hou, B. Xiao, S. Guo, Y. Yang, A Survey of IoT Applications in Blockchain Systems: Architecture, Consensus, and Traffic Modeling, *ACM Computing Surveys (ACM CSUR)*, Vol. 53, No. 1, pp. 1-32, May, 2020.
[9] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, Y. Zhang, Consortium blockchain for secure energy trading in industrial internet of things, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 8, pp. 3690-3700, August, 2018.
[10] Y. Wang, Z. Su, N. Zhang, Bsis: Blockchain-based secure incentive scheme for energy delivery in vehicular energy network, *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 6, pp. 3620-3631, June, 2019.
[11] C. H. Liu, Q. Lin, S. Wen, Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learning, *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 6, pp. 3516-3526, June, 2019.
[12] R. B. Myerson, Optimal auction design, *Mathematics of Operations Research*, Vol. 6, No. 1, pp. 58-73, February, 1981.
[13] J. Hartline, Lectures on Optimal Mechanism Design, 2006.
[14] N. C. Luong, Z. Xiong, P. Wang, D. Niyato, Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach, in *2018 IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, May 2018, pp. 1-6.
[15] P. Dütting, Z. Feng, H. Narasimhan, D. C. Parkes, S. S. Ravindranath, *Optimal auctions through deep learning*, https://arxiv.org/abs/1706.03459, June, 2017.
[16] C. Dai, X. Liu, J. Lai, Human action recognition using two-stream attention based lstm networks, *Applied Soft Computing*, Vol. 86, p. 105820, January, 2020.
[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, *Nature*, Vol. 518, No. 7540, p. 529-533, February, 2015.
[18] A. M. Khan, X. Vilaça, L. Rodrigues, F. Freitag, A distributed auctioneer for resource allocation in decentralized systems, in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, Nara, Japan, June 2016, pp. 201-210.
[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, *Continuous control with deep reinforcement learning*, https://arxiv.org/abs/1509.02971, September, 2015.
[20] C. H. Liu, Z. Chen, J. Tang, J. Xu, C. Piao, Energy-efficient uav control for effective and fair communication coverage: A deep reinforcement learning approach, *IEEE Journal on Selected Areas in Communications*, Vol. 36, No. 9, pp. 2059-2070, September, 2018.
[21] H. van Hasselt, A. Guez, D. Silver, *Deep reinforcement learning with double q-learning*, https://arxiv.org/abs/1509.06461, September, 2015.
[22] Z. Wang, T. Schaul, M. Hessel, H. V. Hasselt, M. Lanctot, N. D. Freitas, Dueling network architectures for deep reinforcement learning, *International conference on machine learning*, New York City, NY, USA, June 2016, pp. 1995-2003.

Biographies

Chuhao Qin is currently pursuing the master's degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include blockchain, Internet of Things, and deep learning

Pan Li is currently a professor with the School of Communication and Information Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include big data networks, cloud computing, and artifical intelligence

Jun Liu is currently pursuing the doctor's degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include wireless communication, edge computing, and deep learning.

Jianqi Liu is currently pursuing the doctor's degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include bid data analysis, deep learning, and edge computing.