Computation Time Minimized Offloading in NOMA-enabled Wireless Powered Mobile Edge Computing

Wenchao Chen, Xinchen Wei, Kaikai Chi, *Senior Member, IEEE*, Keping Yu, *Member, IEEE*, Amr Tolba, *Senior Member, IEEE*, Shahid Mumtaz, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*,

***Abstract*—Wireless powered mobile edge computing (WP-MEC), which combines mobile edge computing (MEC) and wireless power transfer (WPT), is a promising paradigm for coping with the computing power and energy constraints of wireless devices. However, how to realize the online optimal offloading decision and resource allocation in the WP-MEC system is very challenging. This paper studies the system computation completion time (SCCT) minimization problems for WP-MEC networks using non-orthogonal multiple access (NOMA) communication under binary and partial offloading modes. Due to the complexity of the optimization problems and the time-varying nature of the channel state information, we decouple the original problems into a top-problem of optimizing WPT duration and a sub-problem of optimizing resource allocation, and then propose a convolutional deep reinforcement learning online (CDRO) algorithm. For the top-problem, a deep reinforcement learning framework is used to obtain the near-optimal WPT duration, and an incremental exploration policy is designed to balance the exploration accuracy and exploration range to improve the convergence performance of the CDRO algorithm. For the sub-problems, we propose their corresponding low-complexity algorithms based on in-depth analysis and derivation of the optimal offloading decision's properties. Finally, numerical results show that the proposed CDRO algorithm achieves near-optimal SCCT with low computational complexity, enabling online decision-making in time-varying channel environments.**

***Index Terms*—Mobile edge computing, wireless power transfer, deep reinforcement learning, system computation completion time.**

I. Introduction

A growing number of intelligent applications are gaining popularity, such as face recognition, augmented reality, and driverless driving [1]. These intelligent applications frequently require computationally-intensive and latency-sensitive tasks to be performed [2]–[4]. However, due to the size and cost constraints, wireless devices (WDs) are typically equipped with low-power processors and small-capacity batteries, thereby limiting their computing power and energy consumption [5]–[7]. These limitations pose obstacles for WDs to independently handle intelligent applications and provide reliable quality of service (QoS). Therefore, how to solve these two limitations is an urgent challenge for WDs.

Mobile edge computing (MEC) can offload computation tasks from edge devices to MEC servers in a low-latency manner to enhance the computing capabilities of edge devices, meeting the computing resource requirements of applications [8], [9]. Unlike traditional cloud computing, MEC servers are co-located with the information access point (IAP) at the network edge, closer to the edge devices, and can achieve lower latency [10]. Therefore, MEC is promising to overcome the limitations of edge devices' computational power. Furthermore, non-orthogonal multiple access (NOMA) is a key technology for 6G communication systems that can effectively improve the utilization of spectrum resources [11]. NOMA enables multiple users to use the same resource blocks to communicate with the IAP, reducing system delay [12]. Thus, integrating NOMA technology into a multi-user MEC network can effectively address the computation resource limitations of WDs while maintaining low system latency.

The MEC network operates in two modes: binary and partial offloading modes [10]. The binary offloading is used for the case that the computational task is indivisible, and the entire task is either computed locally or offloaded to the MEC server. In contrast, the partial offloading mode breaks down the computation task into two parts, where one part is computed locally, while the other part is offloaded to the MEC server for computation. This paper considers both two offloading modes and explores the performance optimization associated with each mode.

Furthermore, wireless power transfer (WPT) technology, utilizing radio frequency (RF) energy, offers a reliable solution to mitigate the energy consumption constraints of WDs [5], [6]. WPT enables WDs to receive a stable energy supply from the RF energy in the air without affecting the normal operation of the battery, which enhances the service life of WDs [13]. To further leverage the benefits of WPT, the integration of WPT and MEC gives rise to wireless powered

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62272414 and the Researchers Supporting Project Number (RSPD2024R681), King Saud University, Riyadh, Saudi Arabia. *(Corresponding author: Xinchen Wei.)*

W. Chen, X. Wei and K. Chi are with the School of Computer Science and Technology, Zhejiang University of Technology, China (e-mail: {wcchen,weixinchen,kkchi}@zjut.edu.cn).

K. Yu is with the Graduate School of Science and Engineering, Hosei University, Tokyo 184-8584, Japan (e-mail: keping.yu@ieee.org).

S. Mumtaz is with Department of Applied Informatics, Silesian University of Technology, Akademicka 16 44-100 Gliwice, Poland and Department of Computer Sciences, Nottingham Trent University, UK. (e-mail: dr.shahid.mumtaz@ieee.org).

A. Tolba is with the Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia(e-mail: atolba@ksu.edu.sa).

M. Guizani is with the Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), UAE (e-mail: mguizani@ieee.org).

mobile edge computing (WP-MEC) [14], [15]. By combining WPT and MEC, WDs can tap into the computation capabilities of nearby MEC servers while efficiently managing energy resources. Therefore, WP-MEC offers a notable benefit in overcoming the constraints imposed by computing power and energy consumption.

Although the WP-MEC paradigm comes with several advantages, it also introduces certain challenges. One challenge lies in jointly determining reasonable task-offloading decision and resource allocation to optimize system performance. In [16], Bi and Zhang studied a time division multiple access (TDMA) based WP-MEC. They proposed a binary search method and coordinate descent method to obtain the optimal time allocation and offloading decisions, maximizing the computation rate of users. Zhou *et al.* [17] considered an unmanned aerial vehicle (UAV) supported WP-MEC network, using a successive convex approximation (SCA) method to optimize the trajectory of the UAV and maximize the throughput of the network. However, in practical scenarios, the channel conditions of the network are time-varying, necessitating real-time updates of offloading decision and resource allocation in order to adapt to the dynamically changing channel environment. Therefore, the other challenge is how to design low-complexity online algorithms. In recent years, deep reinforcement learning (DRL) has demonstrated great potential in addressing optimization problems in complex scenarios. Yang *et al.* [18] proposed a centralized deep Q-network (DQN) algorithm to minimize energy consumption while satisfying latency requirements in MEC networks. Specifically, the algorithm optimized the user offloading data ratio and resource allocation to achieve the intended objective. Building upon [16], Huang *et al.* [19] proposed a DRL-based online offloading (DROO). DROO algorithm obtained a near-optimal binary offloading decision and upheld the feasibility of online offloading. Due to the exhaustive search of DQN and the binary exploration policy of DROO, these methods are not suitable for finding the optimal action in continuous action space.

Although some offloading algorithms are available, there are still some key issues that need further research. Firstly, most of the existing research does not take into account the computation delay metrics of WP-MEC networks, particularly the minimization of WDs' computation delay while ensuring fairness. Secondly, while NOMA has the potential to enhance connectivity, it poses a significant optimization challenge due to the mutual interference of WDs' signals. Thirdly, most DRL algorithms are not suitable for continuous action space and single-slot optimization problems.

Motivated by the above observations, this paper considers a WP-MEC network using NOMA under both binary and partial offloading modes. Our aim is to minimize the system computation completion time (SCCT) while satisfying a set of constraints. To solve the optimization problems, we propose a convolutional deep reinforcement learning online (CDRO) algorithm. The main contributions of this paper are summarized as follows:

1) The problems of minimizing the SCCT under binary and partial offloading modes are formulated as non-convex problems. To simplify the original complex optimization problems, we decouple the original optimization problems into a common top-problem and two sub-problems. The top-problem focuses on optimizing the WPT duration, while the sub-problem deals with the original optimization problem once the WPT duration is determined.
2) We propose a DRL-based algorithm to efficiently obtain the near-optimal solution for the top-problem. To improve the convergence speed and the algorithm performance, we utilize a convolutional neural network (CNN) to compress the system state into a lower dimension. In addition, we design an incremental step-size exploration policy to balance the exploration precision and the exploration range during the learning process.
3) For the sub-problem of the binary offloading, we first deduce some critical properties. Based on these properties, we introduce an algorithm using both the discrete and continuous bisection search to obtain the optimal binary offloading decisions, the local computation time (LCT) and the offloading time, respectively. For the sub-problem of the partial offloading, we also use the continuous bisection search and the golden section search to obtain the optimal solution.
4) Through performance evaluation, the effectiveness of the CDRO algorithm is verified, which converges fast during the self-learning stage and obtains near-optimal solutions with low computational complexity after converging.

The rest of this paper is organized as follows. The related work is introduced in Section II. Section III and Section IV investigate the system model and problem formulation, respectively. Section V introduces the DRL-based offloading approach. The numerical results are given in Section VI. Finally, Section VII concludes the paper.

II. Related Work

Tran and Pompili [20] studied a multi-server multi-user MEC system, utilizing convex and quasi-convex methods to optimize resource allocation. They used a heuristic algorithm to address the offloading decision problem to maximize user benefits. Yan *et al.* [21] considered a dual-user MEC network in which the computation tasks between users are dependent. They proposed a Gibbs sampling algorithm to minimize the weighted sum of users' energy consumption and computation delay. To minimize the cost of users and edge servers while ensuring network stability, Du *et al.* [22] proposed a Lyapunov-based algorithm and an iterative algorithm based on continuous relaxation and Lagrangian duality to solve the joint optimization problem for servers and clients. Cui *et al.* [23] considered a distributed MEC system and proposed an online anticipatory active network association method to minimize the average task latency under energy consumption constraints.

In the NOMA-based MEC system, Wang *et al.* [24] studied a MEC system that employed multi-carrier NOMA communication. They presented a novel DRL framework to solve the joint resource optimization problem under time-varying channels. The framework consisted of discrete and continuous variable modules, which resulted in a larger structure and

correspondingly more complex training. In uplink NOMA communication MEC networks, Wang *et al.* [25] proposed three frameworks based on DQN and deep deterministic policy-gradient (DDPG) to resolve non-convex joint optimization problems. Their goal was to satisfy the minimum rate requirements of network users while maximizing the energy efficiency of users in the network.

In the WP-MEC system, in order to maximize the computation rate of the backscatter-based WP-MEC network, Nguyen *et al.* [26] proposed a fast and efficient algorithm based on coordinate descent to jointly optimize offloading decisions, resource allocation, and backscatter coefficients. Wang *et al.* [14] considered a beamforming-based WP-MEC system, with the optimization goal of minimizing the energy consumption of the system under the constraint of computational delay. The authors adopted the Lagrangian dual method to jointly optimize the beamforming vector, user computation frequency, offloading duration, and offloading task bits. Li *et al.* [27] considered an IRS-assisted WP-MEC system to achieve better latency performance.

In order to realize the online decision-making, more and more works introduce DRL into the MEC network. Zhou *et al.* [28] investigated a dynamic multi-user MEC network, with the goal of minimizing long-term energy consumption. To address the curse of dimensionality caused by the exponential growth of the action space, the authors proposed an algorithm based on a double deep Q-network (DDQN) to optimize the network's offloading decisions and resource allocation. Chen *et al.* [29] proposed a new two-stage DQN framework to minimize the long-term average energy consumption of WP-MEC systems. Wang *et al.* [30] proposed a probabilistic sampling-based exploration strategy to improve the scalability of the DROO algorithm in large-scale WP-MEC networks, thereby enhancing computation speed. Gao *et al.* [31] considered a mixed task offloading scenario and proposed a multi-agent deep deterministic policy gradient (MADDPG) algorithm based on game. In [32] and [33], the computation rate maximization problem in WP-MEC network based on partial offloading mode was investigated under two multiple access techniques, namely TDMA and frequency division multiple access (FDMA), respectively. To address this problem, a DRL-based algorithm was designed and employed to solve the joint optimization problem. Zhou *et al.* [34] focused on a UAV-assisted WP-MEC network. To address the objective of maximizing network computation bits and ensuring user fairness, they proposed an algorithm that combines Soft Actor-Critic framework with UAV trajectory planning and resource allocation.

III. System Model

This paper investigates a NOMA-based WPT-MEC network, as shown in Fig. 1, where an IAP communicates with N WDs equipped with rechargeable batteries. This IAP is powered by stable energy resources, such as wired grids and diesel generators. Specifically, it includes an MEC server and an energy beacon, providing edge-assisted computation and WPT service, respectively. Both IAP and MEC servers

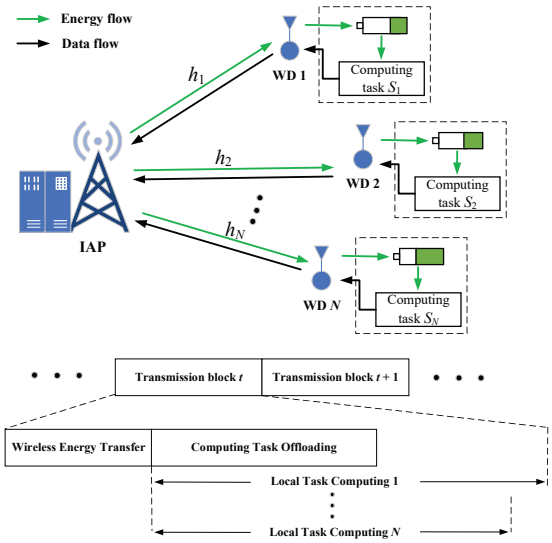


Fig. 1. NOMA-based WPT-MEC network.

carry the same meaning throughout this paper. Each WD operates in half-duplex mode and the WPT-MEC network follows a "harvest-then-transmit" protocol. For such protocol, in each time block, the WD initially captures the RF signal broadcasted by the IAP and stores the RF energy in its battery. Then, the collected energy is utilized by the WD for local computation and edge offloading of computation tasks.

In this paper, we consider both binary and partial computation offloading modes. At the beginning of a time block, each WD receives a computation task, with the amount of computation task for the i-th WD denoted by S_i. During the WPT phase, the IAP provides WPT services to all WDs. In the local task computation phase, WDs can execute their relevant tasks locally. The computation task offloading phase is assigned to offload computation tasks to the IAP based on NOMA and successive interference cancellation (SIC) techniques. Due to the much higher computational power of the IAP compared to the WD and the much smaller size of the task's computation result in comparison to the computation task, the computation time for the IAP and the transmission time of computation results are negligible [16], [19].

A. WPT Model

Let T_W represent the WPT duration which can be expressed as τT_0, where $\tau \in [0,1]$ denotes the WPT duration ratio and T_0 represents the maximum WPT duration. The energy collected by the i-th WD D_i ($i \in \mathcal{N}, \mathcal{N} = \{1,...,N\}$) is given by

$$E_i = \xi P h_i T_W, \tag{1}$$

where h_i is the channel gain between the IAP and D_i, P denotes the transmission power of the RF signal from the IAP, and ξ denotes the energy harvesting efficiency. In (1), this paper considers a linear energy harvesting model to facilitate analysis and gain insights. The proposed algorithm is still appliable when the linear energy harvesting model is replaced by a nonlinear energy harvesting model.

B. Binary Offloading

For the binary offloading mode, the computation tasks can be computed either locally or completely offloaded to the IAP for computation. We denote x_i as the binary offload decision of D_i, such that

$$x_i = \begin{cases} 1, & \text{the task of } D_i \text{ is offloaded to the IAP,} \\ 0, & \text{the task of } D_i \text{ is computed locally.} \end{cases} \tag{2}$$

The details of two offloading modes are provided in the following.

1) Local computing model: Let f_i denote the processor computation speed of D_i in cycles per second, $T^b_{L,i}$ represent the LCT of D_i, and ψ denote the number of cycles needed for the WD's CPU to process one bit of task data. Hence, similar to [16], [32], [35], the amount of local computation $S^b_{L,i}$ can be expressed as

$$S^b_{L,i} = \frac{T^b_{L,i} f_i}{\psi}. \tag{3}$$

The energy consumption constraint is:

$$\gamma_e f_i^3 T^b_{L,i} \leq E_i = \xi P h_i T_W. \tag{4}$$

γ_e represents the CPU effective capacitance coefficient of the WD, which depends on the architecture of the CPU chip. Since our optimization goal aims to minimize the maximum computation time of the WDs in the system, all the available energy must be consumed by the WD, resulting in the inequality (4) being satisfied with equality. Based on the above discussion, we have

$$S^b_{L,i} = \left(\frac{{T^b_{L,i}}^2 \xi P h_i T_W}{\gamma_e \psi^3} \right)^{\frac{1}{3}}. \tag{5}$$

2) Edge computing model: The NOMA technique is employed for offloading WDs, which offload their computation task to the IAP at the same transmission time.

Let the edge offloading time, i.e., the uplink NOMA transmission time, denoted by T^b_E. We ignore the computation time at edge severe and the transmission time of computation results. So the edge computation time (ECT) is T^b_E. The energy consumption constraint of edge computation can be written as

$$p_i T^b_E \leq E_i = \xi P h_i T_W, \tag{6}$$

where p_i represents the transmission power of D_i. Let $\alpha_i E_i$ denote the energy consumed by offloading, and p_i can be expressed as

$$p_i = \frac{\alpha_i \xi P T_W h_i}{T^b_E}, \tag{7}$$

where $0 \leq \alpha_i \leq 1$. Note that $\alpha_i = 1$ is usually not an optimal solution due to interference among offloading WDs.

With loss of generality, assume that $D_1, D_2, ..., D_N$ satisfy

$$h_1 \geq h_2 \geq ... \geq h_N. \tag{8}$$

SIC is utilized at the IAP to decode the received signals from WDs [36], [37]. Specifically, the IAP first detects the signal of the stronger WD with higher channel gain, and then decodes and subtracts it from the received signal. The SIC process is carried out sequentially until the weakest WD's signal is decoded without interference from other WDs.

Let $v_u \geq 1$ denote the proportion of communication overhead during the transmission of computation task, such as packet header, packet trailer or encryption overhead. The amount of transmitted task $S^b_{E,i}$ for D_i is

$$\begin{aligned} S^b_{E,i} &= \frac{BT^b_E}{v_u} \log_2 \left(1 + \frac{p_i h_i}{\sum_{j=i+1}^{N} x_j p_j h_j + N_0} \right) \\ &= \frac{BT^b_E}{v_u} \log_2 \left(1 + \frac{\alpha_i \xi P T_W h_i^2}{\sum_{j=i+1}^{N} x_j \alpha_j \xi P T_W h_j^2 + T^b_E N_0} \right), \end{aligned} \tag{9}$$

where B is communication bandwidth and N_0 is the received noise power.

C. Partial Offloading

For the partial offloading mode, the computation task of the WD can be split arbitrarily, which means that any portion of computation task can be offloaded to the IAP for computation. Let $\alpha_i E_i$ and $(1-\alpha_i)E_i$ denote the energy consumed by offloading computation task and the energy consumed in local computation, respectively.

1) Local computing model: The LCT for D_i is denoted as $T^p_{L,i}$, and thus, the energy consumption constraint is:

$$\gamma_e f_i^3 T^p_{L,i} \leq (1-\alpha_i) E_i = (1-\alpha_i) \xi P h_i T_W. \tag{10}$$

Accordingly, the amount of local computation task $S^p_{L,i}$ is given by

$$S^p_{L,i} = \frac{T^p_{L,i} f_i}{\psi}. \tag{11}$$

Similar to the binary offloading mode, WDs should consume all allocated energy when computing locally. Therefore, the amount of local computation task $S^p_{L,i}$ can be expressed as

$$S^p_{L,i} = \left(\frac{{T^p_{L,i}}^2 (1-\alpha_i) \xi P h_i T_W}{\gamma_e \psi^3} \right)^{\frac{1}{3}}. \tag{12}$$

2) Edge computing model: Without loss of generality, assume that

$$h_1 \geq h_2 \geq ... \geq h_N. \tag{13}$$

The ECT is defined as the task transmission (i.e., offloading) time T^p_E. The transmission power p_i of D_i can be expressed as

$$p_i = \frac{\alpha_i \xi P T_W h_i}{T^p_E}. \tag{14}$$

The transmission task amount $S^p_{E,i}$ for D_i is

$$\begin{aligned} S^p_{E,i} &= \frac{BT^p_E}{v_u} \log_2 \left(1 + \frac{p_i h_i}{\sum_{j=i+1}^{N} p_j h_j + N_0} \right) \\ &= \frac{BT^p_E}{v_u} \log_2 \left(1 + \frac{\alpha_i \xi P T_W h_i^2}{\sum_{j=i+1}^{N} \alpha_j \xi P T_W h_j^2 + T^p_E N_0} \right). \end{aligned} \tag{15}$$

IV. Problem Formulation

Motivated by the importance of task computation latency and user fairness, a min-max problem is considered, aiming to minimize the SCCT, i.e., minimize the maximal task completion time among WDs while satisfying the relevant constraints of the system. In particular, the set of constraints are related to computation amount, energy ratio for offloading and offloading-mode selection indicators.

A. Binary Offloading

The task computation completion time T_i^b for D_i can be expressed as

$$T_i^b = T_W + (1 - x_i)T_{L,i}^b + x_i T_E^b. \tag{16}$$

Lemma 1. *For the offloading WDs, the amount of offloaded computation $S_{E,i}^b$ of WDs increases strictly monotonically with T_E^b and α_i.*

Proof. The partial derivative of $S_{E,i}^b$ with respect to α_i can be expressed as

$$\frac{\partial S_{E,i}^b}{\partial \alpha_i} = \frac{BT_E^b}{v_u \ln 2} \cdot \frac{\xi T_W P h_i^2}{\sum_{j=i+1}^{N} x_j \alpha_j \xi T_W P h_j^2 + T_E^b N_0 + \alpha_i \xi T_W P h_i^2} > 0. \tag{17}$$

The partial derivative of $S_{E,i}^b$ with respect to T_E^b can be expressed as

$$\frac{\partial S_{E,i}^b}{\partial T_E^b} = \frac{B}{v_u \ln 2} \left[\ln\left(1 + \frac{\alpha_i \xi T_W P h_i^2}{\sum_{j=i+1}^{N} x_j \alpha_j \xi T_W P h_j^2 + T_E^b N_0}\right) - \frac{T_E^b}{1 + \frac{\alpha_i \xi T_W P h_i^2}{\sum_{j=i+1}^{N} x_j \alpha_j \xi T_W P h_j^2 + T_E^b N_0}} \cdot \frac{\alpha_i \xi T_W P h_i^2 N_0}{\left(\sum_{j=i+1}^{N} x_j \alpha_j \xi T_W P h_j^2 + T_E^b N_0\right)^2} \right]. \tag{18}$$

For values of x greater than 1, the inequality $\ln x > (x-1)/x$ holds [38]. Therefore, we have $\partial S_{E,i}^b / \partial T_E^b > 0$. This completes the proof. □

Lemma 1 reveals that the computation amount $S_{E,i}^b$ in the edge computing mode is monotonically increasing relative to α_i and T_E^b, and increasing α_i can effectively reduce T_E^b. However, due to the interference caused by the un-demodulated WD signals to the currently demodulated device in NOMA, it is not optimal to consume all the energy for offloading. Therefore, α_i is also one of the variables to be optimized in this paper.

In particular, the aim is to minimize the SCCT by jointly optimizing the variables (i.e., the binary offload decision, time allocation for different phases, and energy ratio) subject to a set of constraints, which can be formulated as the following problem:

$$\begin{aligned}
(\text{Pb}): T_b(\mathbf{h}, \mathbf{S}) = \min_{\mathbf{x}, \boldsymbol{\alpha}, \tau, \mathbf{T}_L^b, T_E^b} \ & \max \ \{T_i^b : i \in \mathcal{N}\} && \text{(19a)} \\
\text{s.t.} \quad & 0 \le \tau \le 1, && \text{(19b)} \\
& 0 \le T_E^b, && \text{(19c)} \\
& 0 \le T_{L,i}^b, \ \forall i \in \mathcal{N}, && \text{(19d)} \\
& 0 \le \alpha_i \le 1, \ \forall i \in \mathcal{N}, && \text{(19e)} \\
& x_i \in \{0, 1\}, \ \forall i \in \mathcal{N}, && \text{(19f)} \\
& (1 - x_i) S_i \le S_{L,i}^b, \forall i \in \mathcal{N}, && \text{(19g)} \\
& x_i S_i \le S_{E,i}^b, \ \forall i \in \mathcal{N}, && \text{(19h)}
\end{aligned}$$

where $\mathbf{h} = [h_1, h_2, \ldots, h_N]$, $\mathbf{S} = [S_1, S_2, \ldots, S_N]$, $\mathbf{x} = [x_1, x_2, \ldots, x_N]$, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ and $\mathbf{T}_L^b = [T_{L,1}^b, T_{L,2}^b, \ldots, T_{L,N}^b]$. The constraint $(19b)$ imposes an upper bound on the duration of WPT. The constraints $(19c)$ and $(19d)$ indicates uplink NOMA transmission time and LCT constraints. The constraint $(19e)$ ensures that the energy consumed by offloading computation tasks does not exceed the collected energy. The constraint $(19f)$ refers to the binary offloading decision constraint. The constraints $(19g)$ and $(19h)$ ensure that the amount of locally computed data or offloaded data is greater or equal to the task data for each WD.

B. Partial Offloading

The task computation time of D_i is determined by the larger one of the LCT and the ECT. The task completion time T_i^p for D_i is

$$T_i^p = T_W + \max\{T_{L,i}^p, T_E^p\}. \tag{20}$$

In particular, the WPT duration, time allocation, and energy ratio for all devices to minimize the SCCT are jointly optimized. This min-max design based on the partial offloading mode can be formulated as:

$$\begin{aligned}
(\text{Pp}): \ T_p(\mathbf{h}, \mathbf{S}) = \min_{\boldsymbol{\alpha}, \tau, \mathbf{T}_L^p, T_E^p} \ & \max \ \{T_i^p : i \in \mathcal{N}\} && \text{(21a)} \\
\text{s.t.} \quad & 0 \le \tau \le 1, && \text{(21b)} \\
& 0 \le T_E^p, && \text{(21c)} \\
& 0 \le T_{L,i}^p, \ \forall i \in \mathcal{N}, && \text{(21d)} \\
& 0 \le \alpha_i \le 1, \ \forall i \in \mathcal{N}, && \text{(21e)} \\
& S_i \le S_{L,i}^p + S_{E,i}^p, \forall i \in \mathcal{N}. && \text{(21f)}
\end{aligned}$$

where $\mathbf{T}_L^p = [T_{L,1}^p, T_{L,2}^p, \ldots, T_{L,N}^p]$. The constraints $(21b)$-$(21e)$ represent the constraints on WPT duration, NOMA communication time, LCT and energy consumption constraints, respectively. The constraint $(21f)$ ensures that the sum of the locally computed data and offloaded data must be greater or equal to the task data for each WD.

Given the coupling relationship between different optimization variables in (Pb) and (Pp), such as the interdependence between $\boldsymbol{\alpha}$, τ and T_E, and the fact that the optimization objective involves minimization and maximization, both (Pb) and (Pp) are non-convex fractional optimization problems. Furthermore, in the binary offloading mode, (Pb) becomes a

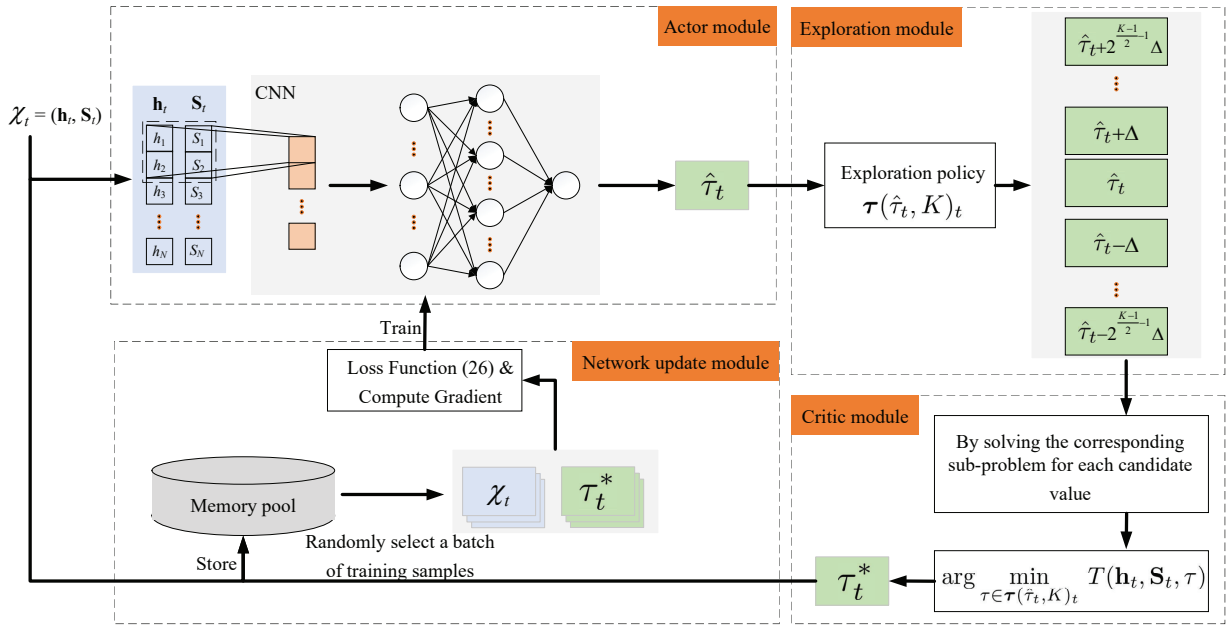


Fig. 2. The framework of the CDRO algorithm.

mixed integer nonlinear programming (MINLP) problem due to the inclusion of the binary decision variable $\mathbf{x}$. Therefore, it is very challenging to quickly solve problems (Pb) and (Pp) under time-varying channels.

V. DRL-based Online Algorithm

To cope with the non-convexity issues in these problems, this paper proposes the decoupling-based CDRO algorithm to decompose the original problem into top-problem and sub-problem.

- *Top-problem*: Optimize the WPT duration ratio τ. This top-problem can be accomplished by employing an online DRL-based CNN model to determine the WPT duration ratio τ.
- *Sub-problem*: Determine the remaining optimization variables of problems (Pb) and (Pp) when the duration of WPT is given. To tackle the sub-problem, this paper proposes an efficient algorithm that leverages a deep understanding of the intrinsic properties of the optimization problem.

A. Top-problem : DRL-based CNN model

To solve the top-problem, this paper aims to quickly generate the near-optimal WPT duration ratio τ of the problem (Pb) and (Pp) through a CNN model. Reinforcement learning (RL) algorithms are generally categorized into three types: actor-only (policy-based method), critic-only (value-based method), and actor-critic (AC) [39], [40]. The AC algorithm integrates both actor-only and critic-only algorithms. The actor can generate discrete or continuous actions based on the current state without the need to optimize the value function directly. The critic provides a lower variance estimate of the value function, which is then used to update the policy function for the actor. Consequently, the AC framework has emerged as a promising approach to RL. Thanks to the advantages of the AC algorithm, we can use past experience to train the CNN model, and then continuously optimize it to output the near-optimal τ. Fig. 2 shows the framework of CDRO algorithm, consisting of four main modules: actor module, critic module, exploration module, and network update module.

1) Actor module: The system state can be defined as $\mathcal{X}_t = (\mathbf{h}_t, \mathbf{S}_t)$, where $\mathbf{h}_t$ is the channel gain at the t-th time block and $\mathbf{S}_t$ is the amount of computation task for WDs at the t-th time block. Considering that $\mathcal{X}_t$ is represented as an $N \times 2$ matrix, we employ the CNN model as the actor module. Compared to the fully connected DNN model, the CNN model's convolution operation is advantageous for extracting essential features from the state matrix. This attribute contributes to a substantial reduction in the number of parameters within the network model and, in turn, enhances the convergence performance of the network. This will be demonstrated and further elaborated upon in the simulation section. The CNN model is defined as:

$$\hat{\tau}_t = \pi_\theta(\mathcal{X}_t), \tag{22}$$

where θ denotes the network parameters of CNN.

2) Exploration module: The output $\hat{\tau}_t$ of the CNN model can be exploited to generate more candidate values in the exploration module. Since θ is initialized randomly, the $\hat{\tau}_t$ generated by the CNN model may not work well in the early time blocks. To achieve the reinforcement learning, we generate multiple candidate values based on $\hat{\tau}_t$. It is worth noting that the fixed-step exploration policy is limited by the predetermined step size. When the step size is small, the accuracy of action exploration increases, but the exploration

range becomes limited. Consequently, finding the optimal WPT duration becomes challenging, leading to prolonged convergence time for the actor module. Conversely, increasing step size expands the exploration range but diminishes the accuracy of mobile exploration. To address these challenges, we use the incremental exploration policy $\boldsymbol{\tau}(\hat{\tau}_t, K)_t$ based on increasing exploration steps to generate $K-1$ additional candidate values, as shown below.

$$\begin{aligned}\boldsymbol{\tau}(\hat{\tau}_t, K)_t = \Big\{ &\hat{\tau}_t + 2^{\frac{K-1}{2}-1}\Delta, \ldots, \hat{\tau}_t + 2^1\Delta, \hat{\tau}_t + 2^0\Delta, \hat{\tau}_t, \\ &\hat{\tau}_t - 2^0\Delta, \hat{\tau}_t - 2^1\Delta, \ldots, \hat{\tau}_t - 2^{\frac{K-1}{2}-1}\Delta \Big\},\end{aligned} \tag{23}$$

where Δ is the exploration step length. Compared with the fixed-step exploration policy, the incremental exploration policy can better balance action exploration accuracy and exploration range. It is necessary to ensure that all the generated τ values satisfy the constraint (19b). In cases where the generated candidate value fails to meet this constraint, we need to discard it and continue generating candidate values based on the exploration policy until K candidate values are obtained.

3) Critic module: The critic module is utilized to evaluate K candidate values and select the optimal WPT duration τ_t^* from them. Unlike the existing AC model that employs a network to assess the candidate values, we evaluate the best candidate value by solving the sub-problems associated with each candidate value in set $\boldsymbol{\tau}(\hat{\tau}_t, K)_t$. This novel approach can lead to more rapid convergence in the actor module and improved accuracy of the critic module in evaluating the candidate values. The details of how to solve the sub-problems are presented in the following two subsections. Therefore, the best WPT duration ratio τ in the t-th time block is

$$\tau_t^* = \arg \min_{\tau \in \boldsymbol{\tau}(\hat{\tau}_t, K)_t} T(\mathbf{h}_t, \mathbf{S}_t, \tau). \tag{24}$$

4) Network update module: We maintain a memory pool with capacity M for storing training samples. In each time block, we combine the state $\mathcal{X}$ and the optimal WPT duration ratio τ^* obtained by the critic module into an input-output sample $(\mathcal{X}, \tau^*)$, and then store it in the memory pool. It is worth noting that a new sample will replace the oldest sample when the memory pool is full. For every fixed training interval ϖ, we randomly select a batch of training samples Γ from the memory pool, and then use the Adam algorithm [41] to update the parameter θ of the CNN. As a result, the loss function can be defined as

$$L(\theta) = \frac{1}{|\Gamma|} \sum_{\gamma=1}^{|\Gamma|} \left[\pi_\theta(\mathcal{X}_\gamma) - \tau_\gamma^*\right]^2, \tag{25}$$

where $|\Gamma|$ denotes the size of the training sample set.

The DRL-based CDRO algorithm is summarized in Algorithm 1. Benefiting from the portability of the DRL framework, for different access schemes, we can also use an online DRL-based CNN model to determine the WPT duration, simply by redesigning the optimization algorithm of the sub-problem.

Algorithm 1: The CDRO algorithm for solving top-problem.

input : For the t-th time block, the wireless channel gain $\mathbf{h}_t$ and the amount of computation task $\mathbf{S}_t$;
output: For the t-th time block, τ^*;
Initialize the CNN parameters θ randomly;
Define the training interval ϖ, the capacity M of memory pool, and the number K of candidate values of τ ;
for $t = 1, 2, \ldots$ **do**
  Generate the WPT duration ratio $\hat{\tau}_t = \pi_\theta(\mathcal{X}_t)$ according to system state $\mathcal{X}_t = (\mathbf{h}_t, \mathbf{S}_t)$;
  Generate K candidate WPT duration ratios $\boldsymbol{\tau}(\hat{\tau}_t, K)_t$ by (23);
  Solve the sub-problems for each τ in the set $\boldsymbol{\tau}(\hat{\tau}_t, K)_t$;
  Select the best WPT duration ratio $\tau_t^* = \arg \min_{\tau \in \boldsymbol{\tau}(\hat{\tau}_t, K)_t} T(\mathbf{h}_t, \mathbf{S}_t, \tau)$;
  Update the memory pool by adding the input-output sample $(\mathcal{X}_t, \tau_t^*)$;
  if $t \bmod \varpi = 0$ **then**
    Randomly select a batch of training samples $\{(\mathcal{X}_\gamma, \tau_\gamma^*) \mid \gamma \in \Gamma\}$ from the memory pool;
    Update network parameters θ by (25) to train CNN;
  end
end

B. Sub-problem : Binary Offloading Mode

For a given WPT duration ratio τ, the sub-problem of the binary offloading mode can be expressed as follows:

$$\begin{aligned}(\text{Pb-S1}):\ T_b(\mathbf{h}, \mathbf{S}, \tau) = &\min_{\mathbf{x}, \boldsymbol{\alpha}, \mathbf{T}_L^b, T_E^b} \max \quad \{T_i^b : i \in \mathcal{N}\} \\ &\text{s.t.} \qquad\quad (19c) - (19h).\end{aligned} \tag{26}$$

Although the DRL framework is effective in finding the near-optimal τ, the sub-problem under the binary offloading mode is still a MINLP problem. In this paper, the action space for binary offloading decisions is related to the number of WDs, and its size is 2^N. This means that the action space increases exponentially with the number of WDs. The simplest method is to find the optimal binary offloading decision through an exhaustive search among all possible decisions. However, the time complexity is intolerable for time-varying channels. Therefore, we propose an approach to greatly reduce the search space of optimal offloading decision.

Theorem 1. *For a given WPT duration ratio τ, the minimal LCT of D_i (if it conducts local computation) $T_{L,i}^{b*}$ can be achieved when $S_{L,i}^b = S_i$.*

$$T_{L,i}^{b*} = (S_i \psi)^{\frac{3}{2}} \left(\frac{\gamma_e}{\xi P h_i T_W}\right)^{\frac{1}{2}}. \tag{27}$$

Proof. When the WPT duration ratio τ is given, the harvested energy of the WD is a fixed value as $E_i = \xi P h_i \tau T_0$. Moreover, the local computation bits $S_{L,i}^b$ strictly increases with $T_{L,i}^b$. Therefore, if the WD utilizes the energy E_i to

compute beyond S_i bits, $T^b_{L,i}$ can be further reduced by decreasing the local computation bits $S^b_{L,i}$ to S_i. Then (27) is obtained by transforming (5). □

Note that there are 2^N possible offloading decisions. In order to reduce the search space of optimal offloading decision, we present an efficient approach. Our algorithm firstly assumes that all WDs perform local computing, and obtains their corresponding LCTs $T^{b*}_{L,i}$ by Theorem 1. Secondly, we reorder N WDs according to the ascending order of the LCT, that is, $T^{b*}_{L,1} \leq T^{b*}_{L,2} \leq \cdots \leq T^{b*}_{L,N}$. It is worth noting that using partial order sorting does not affect the conclusions and proof process of subsequent theorems.

Theorem 2. *There exists one optimal offloading decision* $\mathbf{x}^*$ *where the devices from* D_1 *to* D_{i-1} *use the local computing mode, while the devices from* D_i *to* D_N *use the edge computing mode, i.e.,* $\mathbf{x}^* = [x_1 = 0, \ldots, x_{i-1} = 0, x_i = 1, \ldots, x_N = 1]$.

Proof. Given any one optimal offloading decision $\mathbf{x}^* = [x_1 = 0, \ldots, x_j = 1, \ldots, x_{i-1} = 0, x_i = 1, \ldots, x_N = 1]$ where $x_{i-1} = 0$ and $x_i = x_{i+1} = ... = x_N = 1$, we reset $x_1 = x_2 = ... = x_{i-1} = 0$ and then obtain a new offloading decision $\hat{\mathbf{x}} = [x_1 = 0, \ldots, x_{i-1} = 0, x_i = 1, \ldots, x_N = 1]$. Note that for $\hat{\mathbf{x}}$, the largest LCT among local-computing WDs remains unchanged. Furthermore, as shown in (9), since the number of NOMA devices is reduced, the interference during the offloading may decrease, resulting in a possible reduction in ECT. Therefore, $\hat{\mathbf{x}}$ is definitely another optimal offloading decision. This completes the proof. □

Definition 1. *Call the optimal offloading decision in Theorem 2 as the special optimal offloading decision (SOOD).*

There are $N + 1$ possible SOODs:

$$\begin{cases} \mathbf{x}_1 = [x_1 = 1, \ldots, x_i = 1, \ldots, x_N = 1] \\ \mathbf{x}_2 = [x_1 = 0, x_2 = 1, \ldots, x_N = 1] \\ \qquad\qquad . \\ \qquad\qquad . \\ \qquad\qquad . \\ \mathbf{x}_N = [x_1 = 0, \ldots, x_{N-1} = 0, x_N = 1] \\ \mathbf{x}_{N+1} = [x_1 = 0, \ldots, x_i = 0, \ldots, x_N = 0]. \end{cases} \tag{28}$$

Note that there exist only one or more real SOODs.

Theorem 3. *There exists one and only one SOOD* $\mathbf{x}^* = [x_1 = 0, \ldots, x_{i-1} = 0, x_i = 1, \ldots, x_N = 1]$ *whose minimized* T^{b*}_E *satisfying* $T^{b*}_{L,i-1} \leq T^{b*}_E < T^{b*}_{L,i}$.

Proof. Given any SOOD $\mathbf{x}_m$ with $T^{b*}_E \geq T^{b*}_{L,m}$, we let the m-th WD conduct local computing. Obviously, for this new offloading decision, its minimal transmission time T^{b*}_E remains unchanged and the m-th WD has lower or same computation time. Thus, this is a new SOOD $\mathbf{x}_{m+1}$ with a smaller WDs' average computation time.

If this new SOOD's T^{b*}_E is still greater than or equal to $T^{b*}_{L,m+1}$, we continue to conduct the above operation to obtain another new SOOD until we obtain the SOOD $\mathbf{x}_k$ satisfying $T^{b*}_E < T^{b*}_{L,k}$.

Furthermore, for $\mathbf{x}_{N+1}, ..., \mathbf{x}_{k+1}$, clearly they have a LCT larger than T^{b*}_E of $\mathbf{x}_k$ and are not optimal offloading decision. □

Definition 2. *Call the SOOD of Theorem 3 as the WDs' average computation time minimized SOOD (ACTM-SOOD).*

Finally, our algorithm aims to find the ACTM-SOOD among $N + 1$ possible SOODs (a small offloading-decision space). Furthermore, the ACTM-SOOD has the minimal WDs' average computation time not only among all SOODs (if there are multiple SOODS), but also among all optimal offloading decisions.

As the offloading-decision space is already quite small, the exhuastive search method is acceptable. However, we give one more efficient way (bisection search) to find the ACTM-SOOD based on the following observation.

For the ACTM-SOOD $\mathbf{x}_k$, clearly if its $T^b_E = T^{b*}_{L,k}$, there exist feasible $\alpha_k, .., \alpha_N$. However, for $\mathbf{x}_i$ with $i = k-1, ..., 1$, when $T^b_E = T^{b*}_{L,i}$, there do not exist feasible $\alpha_i, .., \alpha_N$. The reason is as follows. For $\mathbf{x}_i$ with $i = k-1, ..., 1$, they have a smaller LCT. If there exist feasible $\alpha_i, .., \alpha_N$ for $T^b_E = T^{b*}_{L,i}$, they have an minimial ECT $T^{b*}_E < T^{b*}_{L,i}$ which is lower than the ACTM-SOOD. This contradicts with the definition of the ACTM-SOOD. Additionally, for $\mathbf{x}_i$ where $i = k+1, ..., N$, for $T^b_E = T^b_{L,i}$, there exist feasible $\alpha_i, .., \alpha_N$.

Given any $\mathbf{x}_m$, if $T^b_E = T^b_{L,m}$ has feasible $\alpha_m, .., \alpha_N$ (the approach for determining whether $T^b_E = T^b_{L,m}$ has feasible $\boldsymbol{\alpha}$ will be introduced later), the ACTM-SOOD is among $\mathbf{x}_m, ..., \mathbf{x}_1$; otherwise, the ACTM-SOOD is among $\mathbf{x}_{N+1}, ..., \mathbf{x}_m$.

Above all, we use the bisection search method to find the ACTM-SOOD among $\mathbf{x}_N, ..., \mathbf{x}_1$.

Now, we introduce the approach for determining whether there exist feasible $\boldsymbol{\alpha}$ when τ, $\mathbf{x}$ and T^b_E are given.

When τ, $\mathbf{x}$ and T^b_E are given, we only need to determine whether $\boldsymbol{\alpha}$ has feasible solution when $S^b_{E,i} = S_i$ for each offloading D_i.

Theorem 4. *When* τ, $\mathbf{x}$ *and* T^b_E *are given, after reordering* Q *offloading WDs from 1 to* Q *according the descending order of their channel qualities, let*

$$A_i = 2^{r_i} A_{i+1} + (2^{r_i} - 1) T^b_E N_0, i = 1, ..., Q, \tag{29}$$

where $A_{Q+1} = 0$ *and* $r_i = (S_i v_u)/BT^b_E$.

Then we have

$$\alpha_i = \frac{A_i - A_{i-1}}{h_i^2 \xi P T_W}. \tag{30}$$

Proof. (9) can be transformed to be

$$\alpha_i h_i^2 \xi P T_W = (2^{r_i} - 1) \left(\sum_{j=i+1}^{Q} \alpha_j h_j^2 \xi P T_W + T^b_E N_0 \right). \tag{31}$$

Then, (31) can be further transformed as

$$\sum_{j=i}^{Q} \alpha_j h_j^2 \xi P T_W = 2^{r_i} \sum_{j=i+1}^{Q} \alpha_j h_j^2 \xi P T_W + (2^{r_i} - 1) T^b_E N_0. \tag{32}$$

Define $A_i = \sum_{j=i}^{Q} \alpha_j h_j^2 \xi P T_W$. Then the recursive expression (29) is obtained from (32).

Clearly, (30) is obtained from $A_i = \sum_{j=i}^{Q} \alpha_j h_j^2 \xi P T_W$. □

When τ, $\mathbf{x}$ and T_E^b are given, after obtaining $\boldsymbol{\alpha}$, if it does not satisfy the constraint $(19e)$, no feasible $\boldsymbol{\alpha}$ exists.

The last problem is that after we obtain the ACTM-SOOD $\mathbf{x}_m$, we need to minimized T_E^b and obtain its corresponding $\boldsymbol{\alpha}$, which is expressed as follows.

$$\begin{aligned}(\text{Pb-S2}) : T_b(\mathbf{h}, \mathbf{S}, \tau, \mathbf{x}_m, \mathbf{T}_L^b) = \min_{\boldsymbol{\alpha}, T_E^b} \quad & T_W + \max\{T_E^b, T_{L,m-1}^{b*}\} \\ \text{s.t.} \quad & (19c) - (19h).\end{aligned} \tag{33}$$

Clearly a too small T_E^b cause that no infeasible $\boldsymbol{\alpha}$ exists. We need to find the minimal T_E^b with feasible $\boldsymbol{\alpha}$. Theorem 3 shows that T_E^{b*} is smaller than $T_{L,m}^{b*}$. Since the Lemma 1 confirms that $S_{E,m}^b$ is strictly monotonically increasing with T_E^b, the bisection search method is applied for finding the minimal T_E^b with feasible $\boldsymbol{\alpha}$ in $[0, T_{L,m}^{b*}]$.

Algorithm 2 summarizes the sub-problem solving algorithm based on discrete binary search method for binary offloading model. It is worth noting that in binary offloading mode, WD that performs edge computing completely offloads all computing tasks to IAP for auxiliary computing, and then the computation time of IAP becomes a fixed value $T_{E,m}^b$. Therefore, if we take into account the computation time of IAP, we only need to modify $T_E^b = T_{L,m}^{b*}$ in lines 6 to $T_E^b = T_{L,m}^{b*} - T_{E,m}^b$ and modify $T_E^b = T_m$ in lines 17 to $T_E^b = T_m - T_{E,m}^b$ of Algorithm 2.

Algorithm 2: The sub-problem solving algorithm for binary offloading model

input : For the t-th time block, the wireless channel gain $\mathbf{h}_t$, the amount of computation task $\mathbf{S}_t$ and the WPT duration ratio τ;
output: For the t-th time block, $\mathbf{x}^*$, $\boldsymbol{\alpha}^*$, $\mathbf{T}_L^{b*}$ and T_E^{b*};
1 Order N WDs from 1 to N in the ascending order of LCT, i.e., $T_{L,1}^{b*} \le T_{L,2}^{b*} \le \cdots \le T_{L,N}^{b*}$;
2 Initialize $N+1$ possible SOODs by (28);
3 Initialize $l = 0$, $r = N$ and the tolerance error $\beta > 0$;
4 **while** $l \neq r$ **do**
5 Let $m = (l + r)/2$;
6 For the given $\mathbf{x}_m$ and $T_E^b = T_{L,m}^{b*}$ to get $\boldsymbol{\alpha}$ according to (30);
7 **if** $\boldsymbol{\alpha}$ *does not satisfies constraint* $(19e)$ **then**
8 Update the search space by setting $l = m$;
9 **else**
10 Update the search space by setting $r = m$;
11 **end**
12 **end**
13 Obtain the ACTM-SOOD $\mathbf{x}^* = \mathbf{x}_m$;
14 Initialize $T_l = 0$ and $T_u = T_{L,m}^{b*}$;
15 **while** $T_u - T_l < \beta$ **do**
16 Let $T_m = (T_l + T_u)/2$;
17 For the given $\mathbf{x}^*$ and $T_E^b = T_m$ to get $\boldsymbol{\alpha}$ according to (30);
18 **if** $\boldsymbol{\alpha}$ *satisfies constraint* $(19e)$ **then**
19 Update the search space by setting $T_u = T_m$
20 **else**
21 Update the search space by setting $T_l = T_m$;
22 **end**
23 **end**
24 Obtain the minimum ECT $T_E^{b*} = T_m$.

C. Sub-problem : Partial Offloading Mode

When the WPT duration ratio τ is given, the sub-problem of the partial offloading can be formulated as follows:

$$\begin{aligned}(\text{Pp-S}) : \ T_p(\mathbf{h}, \mathbf{S}, \tau) = \min_{\boldsymbol{\alpha}, \mathbf{T}_L^b, T_E} \max \quad & \{T_i^p : i \in \mathcal{N}\} \\ \text{s.t.} \quad & (21c) - (21f).\end{aligned} \tag{34}$$

While DRL is effective at determining the duration of WPT, finding a solution to the (Pp-S) problem remains a challenge due to the need of balancing the fairness among WDs and the coupling of optimization variables.

Let T_c denote the system computation time, i.e., $T_c = T_p - T_W$, and S_i^p denote the computation bits of D_i, i.e., $S_i^p = S_{L,i}^p + S_{E,i}^p$. It is clear that for a given T_c, the maximum LCT $T_{L,i}^p$ and offloading time T_E^p for D_i are both T_c. Therefore, we can assess the feasibility of a given T_c by verifying if there exists feasible $\boldsymbol{\alpha}$ that satisfies $S_i^p \ge S_i$ for each D_i under the condition that $T_{L,i}^p = T_E^p = T_c$. Note that, if there does not exist feasible $\boldsymbol{\alpha}$, there does not exist feasible $\boldsymbol{\alpha}$ for any $T_{L,i}^p$ and T_E^p which are smaller than T_c.

Below we introduce how to determine whether there exists α_i that satisfies $S_i^p \ge S_i$ under the condition that $T_{L,i}^p = T_E^p = T_c$.

Lemma 2. *When τ, $T_{L,i}^p$, T_E^p are given, the computation bits S_i^p of D_i is a concave function on α_i, as illustrated in Fig.* 3.

Proof. The second derivative of $S_{L,i}^p$ with respect to α_i can be expressed as

$$S_{L,i}^{p\prime\prime} = -\frac{2}{9\psi}(1-\alpha_i)^{-\frac{5}{3}} \left(\frac{{T_{L,i}^p}^2 \xi P h_i T_W}{\gamma_e} \right)^{\frac{1}{3}} \le 0. \tag{35}$$

So $S_{L,i}^p$ is a concave function about the variable α_i.

For $S_{E,i}^p$, due to the properties of the log function, $S_{E,i}^p$ is also a concave function about the variable α_i. Since the sum of concave functions is also a concave function, S_i^p is a concave function about the variable α_i. This completes the proof. □

Without loss of generality, assume $h_1 \ge h_2 \ge ... \ge h_N$. In the uplink NOMA communication, the signal of D_i is the interference for any $D_j (j = 1, ..., i-1)$ whose signals are decoded before D_i. Therefore, for each WD, we should minimize α_i while guaranteeing $S_i^p \ge S_i$. In our algorithm, from $i = N$ to $i = 1$ (from D_N to D_1), one by one we determine whether feasible α_i exists when $T_{L,i}^p = T_E^p = T_c$. If feasible α_i does not exist, no feasible $\boldsymbol{\alpha}$ exist (i.e., the given T_c is infeasible). If feasible α_i exists, we need to find the minimal feasible α_i to minimize the interference to previously decoded WDs and then continue to determine whether feasible α_{i-1} exists.

For any D_i, based on Lemma 2, we can obtain the computation bits S_i^l when $\alpha_i = 0$, and obtain its maximum computa-

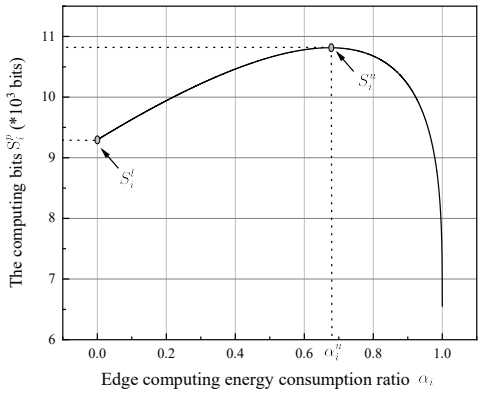


Fig. 3. The approximate graph of S_i^p with respect to α_i.

tion bits S_i^u (by the golden section search) and corresponding α_i^u. Then we have the following cases:

Case 1: $S_i^u < S_i$. This means that T_c is not feasible.

Case 2: $S_i^l \geq S_i$. This means that the minimal feasible α_i is 0.

Case 3: $S_i^l < S_i \leq S_i^u$. The minimal feasible α_i can be found by using the bisection search in the range $(0, \alpha_i^u]$ which satisfying $S_i^p = S_i$.

Knowing how to determine whether a given T_c is feasible, we can use the bisection search method to find the minimal feasible T_c in the range $[0, T_c^{max}]$ where $T_c^{max} = \max\{T_{L,1}^b, ..., T_{L,N}^b\}$ denote the maximum LCT among all WDs when $\boldsymbol{\alpha} = 0$ (i.e., the LCT $T_{L,i}^p = T_{L,i}^b$ of D_i when $\alpha_i = 0$). This is because a too small T_c is obviously not feasible. Additionally, if a specific T_c is feasible, a larger T_c is definitely feasible because even the same $\boldsymbol{\alpha}$ causes larger $S_{L,i}^p$ and $S_{E,i}^p$ (refer to (12) and (15)).

Algorithm 3 summarizes the sub-problem solving algorithm based on the binary search method for partial offloading mode. In partial offloading mode, WD can offload part of the computing tasks to IAP for auxiliary computing, and then the computing time of IAP becomes a variable related to edge computing energy consumption, which will affect the design of the algorithm.

Algorithm 3: The sub-problem solving algorithm for partial offloading mode

input : For the t-th time block, the wireless channel gain $\mathbf{h}_t$, the amount of computation task $\mathbf{S}_t$ and the WPT duration ratio τ;
output: For the t-th time block, $\boldsymbol{\alpha}^*$, T_c^*, $\mathbf{T}_L^{b*}$ and T_E^{b*};
1 Initialize the set $\mathcal{N}$ of WDs with $h_1 \geq h_2 \geq ... \geq h_N$;
2 Initialize $T_l = 0$ and $T_u = T_c^{max} = \max\{T_{L,1}^{b*}, ..., T_{L,N}^{b*}\}$;
3 Initialize the tolerance error $\beta > 0$;
4 **while** $T_u - T_l < \beta$ **do**
5 Let $T_m = (T_l + T_u)/2$;
6 **for** $i = N, ..., 1$ **do**
7 Let $T_{L,i}^p = T_E^p = T_m$, obtain $S_i^l(\alpha_i = 0)$, obtain S_i^u and α_i^u by the golden section search;
8 **if** $S_i^u < S_i$ **then**
9 T_m is not feasible and $T_l = T_m$;
10 Break;
11 **if** $S_i^l \geq S_i$ **then**
12 The minimal feasible $\alpha_i^* = 0$;
13 **else**
14 Initialize $\alpha_l = 0$ and $\alpha_u = \alpha_i^u$;
15 **while** $\alpha_u - \alpha_l < \beta$ **do**
16 Let $\alpha_m = (\alpha_u + \alpha_l)/2$;
17 **if** α_i *satisfying* $S_i^p > S_i$ **then**
18 $\alpha_u = \alpha_i$
19 **else**
20 $\alpha_l = \alpha_i$;
21 **end**
22 **end**
23 The minimal feasible $\alpha_i^* = \alpha_i$;
24 **end**
25 **end**
26 **if** $T_l \neq T_m$ **then**
27 Update the search space by setting $T_u = T_m$
28 **end**
29 **end**
30 $T_c^* = T_m$.

D. Algorithm complexity analysis

In the actor model of Algorithm 1, the system state $\mathcal{X}$ (N×2 dimensional) is input into the CNN model to generate the WPT duration τ (1×1 dimensional). The CNN model comprises M layers, with the number of neurons in the m-th layer denoted as U_m. Therefore, the computational complexity of the CNN model is $O_1 = O((N+2)U_1 + \sum_{m=2}^{M-1}(U_{m-1}U_m + U_mU_{m+1}) + U_M)$, which can be simplified to $O_1 = O(N)$ [42]. For Algorithm 2, it involves two binary search algorithms (lines 4 to 12 and lines 15 to 23). Therefore, the complexity of Algorithm 2 is $O_2 = O(\log_2(N+1) + \log_2(T_u/\beta))$ where β represents search accuracy and T_u represents the search upper bound, which can be simplified to $O_2 = O(\log_2(N))$ [16]. Algorithm 3 encompasses two binary searches (lines 4 to 29 and lines 15 to 22) and an N-times loop nesting (lines 6 to 25). Therefore, the complexity of Algorithm 3 is $O_3 = O(\log_2(T_u/\beta)N\log_2(\alpha_u/\beta))$ where T_u and α_u represent the upper bounds of the two binary searches respectively, which can be simplified to $O_3 = O(N)$ [16].

VI. Performance Evaluation

This section evaluates the performance of the proposed CDRO algorithm. The values of parameters in the simulations are listed in Table I unless otherwise stated. The time-varying channel model is considered, which follows the Rayleigh fading distribution. Specifically, the distance d_i from D_i to the IAP in the MEC network is randomly distributed within [10, 15] meters. $\bar{h}_i$ represents the average channel gain at D_i, which follows the free-space path loss model $\bar{h}_i = A_d\left(\frac{10^8}{4\pi f_c d_i}\right)^{d_e}$ where A_d = 4.11 is the antenna gain, f_c = 780 MHz is the carrier frequency and d_e = 2.5 is path loss exponent. In the t-th time block, the channel gain h_i^t of D_i is generated according to the Rayleigh fading model as $h_i^t = \bar{h}_i\delta_i^t$, where δ_i^t represents the independent random channel fading factor following an exponential distribution with a unit mean. At the beginning of each time block, each

TABLE I
SIMULATION PARAMETERS

Parameters	Notation	Value
The maximum wireless energy supply duration	T_0	5 s
The energy harvesting efficiency	ξ	0.7
RF signal transmission power of IAP	P	3 W
The number of cycles needed to process one bit of task data	ψ	100
The CPU effective capacitance coefficient	γ_e	10^{-25}
The proportion of communication overhead	υ_u	1.1
The communication bandwidth	B	3 MHz
The received noise power	N_0	10^{-11}
The exploration step length	Δ	0.001
The memory pool size	M	1500
The training interval	ϖ	10
The batch size of training samples	$\|\Gamma\|$	500

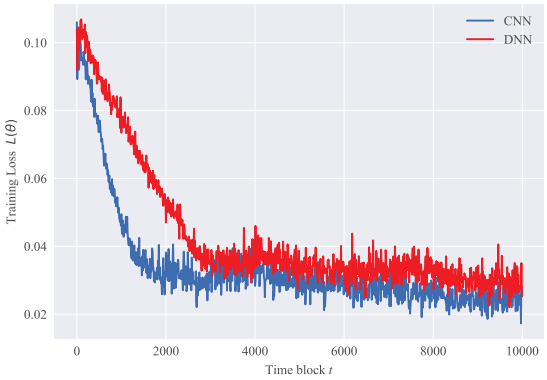


Fig. 4. Convergence performance comparison of using DNN and CNN in our algorithm under $N = 6$ and $K = 17$.

WD has a new computation task with the task size S_i of a random value in [1, 10] kB. In the CDRO algorithm, the CNN network consists of an input layer, a convolutional layer (the size of the convolutional kernel is 2×2), three hidden layers (there are 1200, 820, and 120 neurons, respectively), and an output layer. Additionally, the activation functions of hidden layers' neurons and output layer' neuron are ReLU function and sigmoid function, respectively. The simulations were conducted on a computer featuring a 4.9 GHz Intel Processor and 16 GB RAM. Additionally, the CDRO algorithm was implemented using PyTorch 1.7 in Python 3.7.

We compare our CRDO algorithm with the following algorithms.

- *One-dimensional search (ODS)*: In the value range [0,1] of WPT duration ratio τ, exhaustively enumerate all possible τ with a step size of 0.0001. Then, solve the corresponding sub-problems for each τ to obtain the minimal SCCT.
- *DDPG*: Unlike typical reinforcement learning methods, DDPG can deal with continuous action spaces, which can be utilized in the studied scenario. Based on the AC framework, DDPG combines deterministic policy gradient with DNN. In particular, the policy-based actor network generates continuous WPT duration ratio τ, while the value-based critic network is used to evaluate the currently generated τ [25], [43].
- *Pure DRL (PDRL)*: Use a DRL model to learn all optimization variables in (Pb) and (Pp). If the solution generated by the DRL model does not comply with the system model constraints, a penalty term is introduced for the reward to penalize the infeasible solution that do not meet the specified constraints.
- *Pure local computation (PLC)*: All computation tasks are computed locally.
- *Pure edge computation (PEC)*: All computation tasks are offloaded to IAP for computing.

We define the normalized SCCT $\hat{T}_b$ of binary offloading as

$$\hat{T}_b = \frac{T_b^*(\mathbf{h}, \mathbf{S})}{T_b'(\mathbf{h}, \mathbf{S})}, \tag{36}$$

and the normalized SCCT $\hat{T}_p$ of partial offloading as

$$\hat{T}_p = \frac{T_p^*(\mathbf{h}, \mathbf{S})}{T_p'(\mathbf{h}, \mathbf{S})}. \tag{37}$$

T_b^* and T_p^* are the SCCTs obtained by solving problems (Pb) and (Pp) through our CDRO algorithm, respectively. Correspondingly, T_b' and T_p' are the minimal SCCTs obtained through the ODS algorithm, respectively.

A. Convergence Performance

Since the training process of DRL models inherently involves uncertainty the convergence process can be influenced by different actor models and exploration strategies. Therefore, similar to most related references [25], [26], [28], [30], [32], we use experimental simulations to demonstrate the convergence performance of the CDRL algorithm proposed in this paper. This subsection evaluates the convergence performance of the proposed algorithm. In Fig. 4, we compare the convergence performance of different actor modules in the CDRO algorithm under the binary offloading mode. The figure plots the training loss $L(\theta)$ using the CNN model and the DNN model as the actor module, respectively, under 10000 time blocks. Specifically, the DNN model replaces the convolutional layer in the CNN model with a fully connected layer, and the system state $\mathcal{X}_t$ is flattened as the input of the DNN. As expected, as the exploration module of CDRO continuously improves the quality of samples, the training loss gradually decreases. The $L(\theta)$ of the DNN model tends to converge and stabilize at 0.035 after 3000 time blocks, while the $L(\theta)$ of the CNN model tends to converge and stabilize at 0.025 after 2000 time blocks. This is because the convolutional layer of the CNN model can effectively extract system state features and also reduce the number of neural network parameters. Therefore, the CNN model can effectively improve the convergence speed of the CDRO algorithm as compared to DNN.

Fig. 5 and Fig. 6 show the training loss $L(\theta)$ of the CNN model and normalized SCCT of the CDRO algorithm in the binary and partial offloading modes, respectively. Specifically, the blue shading at each time block represents the normalized SCCT under the current time block, and the

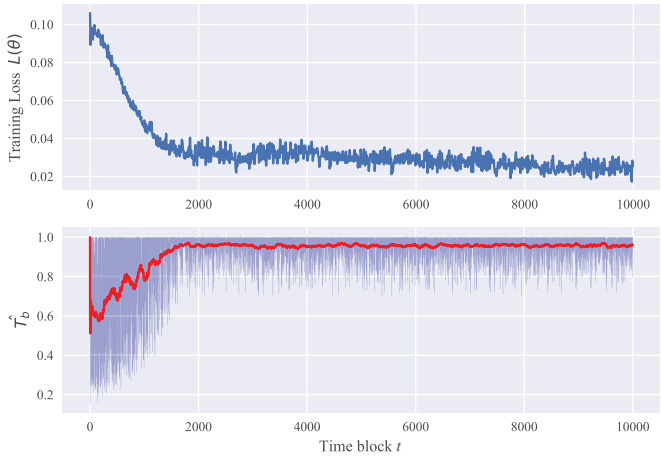


Fig. 5. Training losses $L(\theta)$ for CNN and normalized SCCT $\hat{T}_b$ under binary offloading mode when N=6 and K=17.

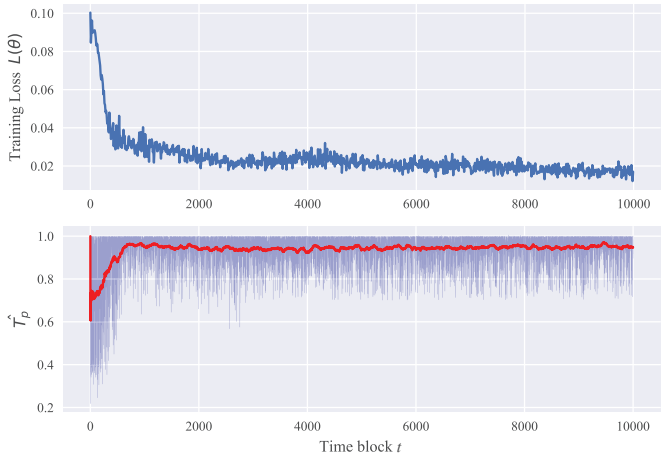


Fig. 6. Training losses $L(\theta)$ for CNN and normalized SCCT $\hat{T}_p$ under partial offloading mode when N=6 and K=17.

red curve represents the average normalized SCCT over the last fifty time blocks. The parameters of the CNN model are initialized randomly, and the samples at the early time blocks are of low quality. However, as the exploration module continuously improves the quality of training samples, $L(\theta)$ gradually decreases until convergence, and the performance of the CRDO algorithm tends to be near-optimal. After 2000 time blocks, the CDRO algorithm in the binary offloading mode tends to converge, with $L(\theta)$ oscillating at 0.025. Meanwhile, the CDRO algorithm in the partial offloading mode tends to converge after 1500 time blocks, with $L(\theta)$ oscillating at 0.02. When the CDRO algorithm converges, the average $\hat{T}_b$ in the binary offloading mode is 0.961, while the average $\hat{T}_p$ in the partial offloading mode is 0.956. Furthermore, among the last 2000 time blocks in Fig. 5, the average normalized SCCT for 972 time blocks exceeds 98%. Among the last 2000 time blocks in Fig. 6, 893 time blocks exceed the average normalized SCCT by 98%. Therefore, the performance of the proposed CDRO algorithm is very close to the ODS algorithm which obtains the optimal solutions.

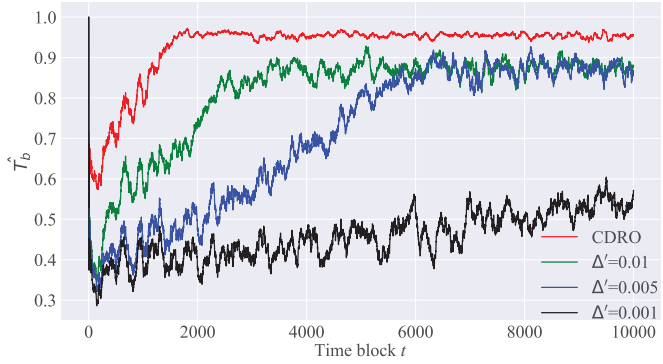


Fig. 7. Achieved normalized SCCT with different exploration policies under $N = 6$ and K=17.

B. The Influence of Different Exploration Policies

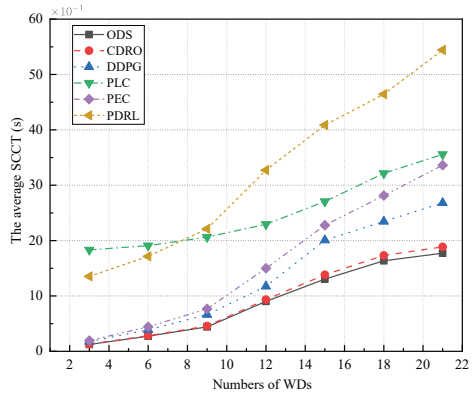
In Fig. 7, we compare the algorithm performance of our incremental exploration policy with the usually used fixed-step exploration policy under the binary offloading mode [33], [44]. The fixed-step exploration policy $\boldsymbol{\tau}'(\hat{\tau}_t, K)_t$ can be expressed as

$$\boldsymbol{\tau}'(\hat{\tau}_t, K)_t = \Big\{ \hat{\tau}_t + \frac{K-1}{2}\Delta', \ldots, \hat{\tau}_t + 2\Delta', \hat{\tau}_t + \Delta', \hat{\tau}_t, \hat{\tau}_t - \Delta', \hat{\tau}_t - 2\Delta', \ldots, \hat{\tau}_t - \frac{K-1}{2}\Delta' \Big\}, \tag{38}$$

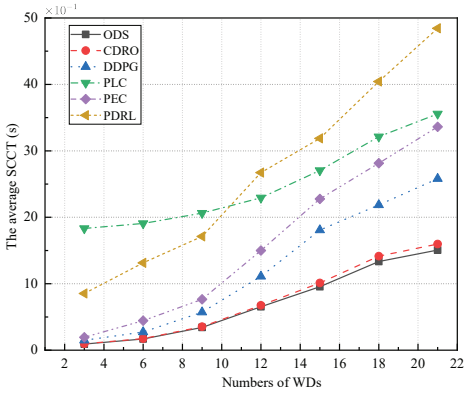
where Δ' denotes the exploration step length. As shown in Fig. 7, the value of Δ' will greatly affect the algorithm's convergence performance and accuracy. Whatever the value of Δ' is, however, the fixed-step exploration policy is worse than our incremental exploration policy. This is due to the fixed-step exploration policy uses a fixed step size. When Δ' is small, the accuracy of action exploration is high, but the corresponding exploration range is very limited. As a result, it becomes challenging to discover the optimal WPT duration, which leads to a long convergence time for the CNN model. Increasing Δ' expands the range of exploration but simultaneously reduces the accuracy of action exploration. This results in significant fluctuations in the convergence process of the algorithm. In contrast, our incremental exploration policy achieves a good balance between the exploration accuracy and exploration range, leading to a faster convergence speed and higher accuracy of the algorithm.

C. Evaluation of SCCT

Due to the black-box nature of CNN, it is challenging for us to theoretically analyze the gap between the solution output by CNN and the optimal solution. Therefore, similar to most related works [25], [26], [28], [30], [32], we demonstrate the effectiveness of the algorithm through simulations. In Fig. 8, we evaluate the average SCCT versus the number of WDs for different algorithms under the binary and partial offloading modes. Each data point on the graph represents the average SCCT of 3000 time blocks. Since the partial offloading mode can split computation tasks arbitrarily, the partial offloading mode can achieve a smaller SCCT than the binary offloading



(a) Binary offloading scheme.



(b) Partial offloading scheme.

Fig. 8. Comparison of the average SCCT performance for different offloading algorithms.

mode. It can be seen from the figure that the ODS algorithm can obtain the minimum SCCT, but due to its exhaustive search, its computational complexity is unacceptable in practical scenarios. Furthermore, due to the weak computation ability of WDs, the SCCT of the PLC algorithm is the largest. As the number of WDs increases, interference becomes more pronounced in the NOMA communication process, resulting in a larger gap between the SCCT of the PEC algorithm and those of other algorithms. The CDRO algorithm outperforms the PDRL algorithm in terms of the binary offloading mode and the partial offloading mode. This significant enhancement can be attributed to the reduced complexity in learning variables within the reinforcement learning framework. Compared with the DDPG algorithm, the SCCT obtained by the proposed CDRO algorithm is closer to the ODS algorithm. A notable improvement of the CDRO algorithm over the DDPG algorithm lies in its ability to evaluate the utilities of actions generated by the actor module through solving the sub-problem. This enhanced evaluation capability enables the CDRO algorithm to achieve a performance level close to the ODS algorithm. In general, the performance of the CDRO algorithm is greatly better than that of the PDRL, DDPG, PLC, and PEC algorithms in the SCCT, and even when the number of WDs is large, its SCCT is still close to that of the ODS algorithm.

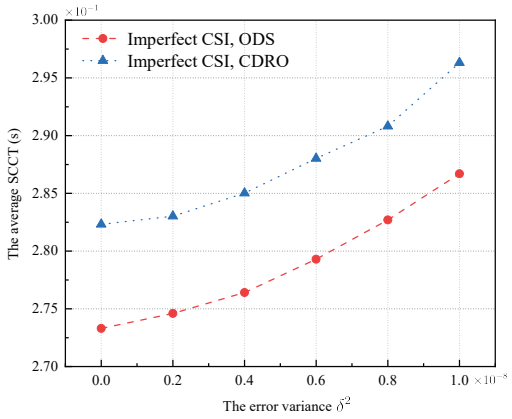


Fig. 9. The impact of imperfect CSI on the CDRO algorithm under $N = 6$ and K=17.

TABLE II
CPU EXECUTION DELAY WITH DIFFERENT NUMBERS OF WDS.

Numbers of WDs	ODS	CDRO	DDPG	PDRL
	Execution delay	Execution delay	Execution delay	Execution delay
3	2.7e-1s	4.2e-3s	1.3e-3s	0.8e-3s
6	4.9e-1s	1.1e-2s	3.1e-3s	1.7e-3s
9	8.4e-1s	1.8e-2s	6.8e-3s	3.1e-3s
12	1.4s	2.8e-2s	1.0e-2s	4.4e-3s
15	3.8s	3.6e-2s	1.9e-2s	5.8e-3s
18	6.2s	5.1e-2s	2.3e-2s	7.2e-3s
21	9.3s	6.8e-2s	3.7e-2s	8.4e-3s

D. Stability Analysis

To verify the stability of the CDRO algorithm, we assume that there is an error in the channel state information (CSI) obtained by the system due to the channel estimation and quantization [45]–[47]. Let $\hat{h}_i$ represent the estimated value of the channel gain h_i from the i-th WD to the IAP, the actual h_i takes the form as

$$h_i = \hat{h}_i + \varepsilon, \tag{39}$$

where ε is the error bound of channel estimation, which follows a Gaussian distribution with mean 0 and variance δ^2, represented by $\varepsilon \sim \mathcal{CN}(0, \delta^2)$. Note that δ^2 denotes the quality of the channel estimate. Fig. 9 shows the achieved SCCT with different error variances. It can be observe that the performance of the CDRO algorithm decreases with increasing the error variance δ^2 because the larger the channel estimation error, the stronger the interference. $\delta^2 = 0$ reduces to a scenario with perfect CSI without noise. Under imperfect CSI, although the performance of the CDRO algorithm will be affected, it can still achieve performance close to ODS.

E. The Execution Delay

Table II evaluates the execution delay of different algorithms. As the PLC and PEC algorithms rely entirely on local and edge computing, they are not included in the table. The execution delays specified in the table represent the average

execution delays for 3000 time blocks. The ODS algorithm has the highest execution delay, while the PDRL algorithm has the shortest execution time. The ODS algorithm requires the exhaustive enumeration of WPT durations and the subproblem solving for each WPT duration, which consumes considerable time. Since PDRL outputs all optimization variables simultaneously, it has the lowest computational complexity, but its SCCT performance cannot meet actual need. Additionally, the DDPG algorithm uses four neural networks, but the CDRO algorithm generates additional $K-1$ candidate WPT durations through its exploration module, needing to solve more subproblems per time block than the DDPG algorithm. Therefore, the execution time of the CDRO algorithm exceeds that of the DDPG algorithm. In summary, the CDRO algorithm achieves the near-minimal SCCT while requiring only a small execution delay.

VII. Conclusion

This paper investigated the problem of minimizing the system computation time in WP-MEC networks using the NOMA communication under two offloading modes: binary offloading and partial offloading. Due to the complexity of the optimization problem and the time-varying nature of the channel state, we decompose the original problem into two problems: the WPT duration optimization problem and the resource allocation problem, and then propose an online offloading algorithm called the CDRO algorithm. Simulation results demonstrate that our proposed CDRO algorithm can achieve the near-minimal computation time with a small execution delay, enabling online decision-making in time-varying channel environments.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[3] L. Zhu, H. Liang, H. Wang, B. Ning, and T. Tang, "Joint security and train control design in blockchain-empowered cbtc system," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8119–8129, 2022.

[4] J. Feng, L. Liu, X. Hou, Q. Pei, and C. Wu, "Qoe fairness resource allocation in digital twin-enabled wireless virtual reality systems," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3355–3368, 2023.

[5] P. Ramezani and A. Jamalipour, "Toward the evolution of wireless powered communication networks for the future Internet of Things," *IEEE Network*, vol. 31, no. 6, pp. 62–69, 2017.

[6] K. W. Choi, L. Ginting, A. A. Aziz, D. Setiawan, J. H. Park, S. I. Hwang, D. S. Kang, M. Y. Chung, and D. I. Kim, "Toward realization of long-range wireless-powered sensor networks," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 184–192, 2019.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[8] F. Jalali, S. Khodadustan, C. Gray, K. Hinton, and F. Suits, "Greening iot with fog: A survey," in *2017 IEEE international conference on edge computing (EDGE)*. IEEE, 2017, pp. 25–31.

[9] K. Jiang, C. Sun, H. Zhou, X. Li, M. Dong, and V. C. Leung, "Intelligence-empowered mobile edge computing: Framework, issues, implementation, and outlook," *IEEE Network*, vol. 35, no. 5, pp. 74–82, 2021.

[10] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[11] Y. Yuan, S. Wang, Y. Wu, H. V. Poor, Z. Ding, X. You, and L. Hanzo, "NOMA for next-generation massive IoT: Performance potential and technology directions," *IEEE Communications Magazine*, vol. 59, no. 7, pp. 115–121, 2021.

[12] G. Li, M. Zeng, D. Mishra, L. Hao, Z. Ma, and O. A. Dobre, "Energy-efficient design for IRS-empowered uplink MIMO-NOMA systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9490–9500, 2022.

[13] S. Bi, C. K. Ho, and R. Zhang, "Wireless powered communication: Opportunities and challenges," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 117–125, 2015.

[14] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784–1797, 2017.

[15] K. Zheng, G. Jiang, X. Liu, K. Chi, X. Yao, and J. Liu, "DRL-based offloading for computation delay minimization in wireless-powered multi-access edge computing," *IEEE Transactions on Communications*, vol. 71, no. 3, pp. 1755–1770, 2023.

[16] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177–4190, 2018.

[17] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[18] Y. Yang, Y. Hu, and M. C. Gursoy, "Deep reinforcement learning and optimization based green mobile edge computing," in *2021 IEEE CCNC*. IEEE, 2021, pp. 1–2.

[19] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581–2593, 2019.

[20] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2018.

[21] J. Yan, S. Bi, Y. J. Zhang, and M. Tao, "Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 235–250, 2019.

[22] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1079–1092, 2018.

[23] Q. Cui, J. Zhang, X. Zhang, K.-C. Chen, X. Tao, and P. Zhang, "Online anticipatory proactive network association in mobile edge computing for IoT," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4519–4534, 2020.

[24] S. Wang, T. Lv, W. Ni, N. C. Beaulieu, and Y. J. Guo, "Joint resource management for MC-NOMA: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 5672–5688, 2021.

[25] X. Wang, Y. Zhang, R. Shen, Y. Xu, and F.-C. Zheng, "DRL-based energy-efficient resource allocation frameworks for uplink NOMA systems," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7279–7294, 2020.

[26] P. X. Nguyen, D.-H. Tran, O. Onireti, P. T. Tin, S. Q. Nguyen, S. Chatzinotas, and H. V. Poor, "Backscatter-assisted data offloading in OFDMA-based wireless-powered mobile edge computing for IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9233–9243, 2021.

[27] G. Li, M. Zeng, D. Mishra, L. Hao, Z. Ma, and O. A. Dobre, "Latency minimization for IRS-aided NOMA MEC systems with WPT-enabled IoT devices," *IEEE Internet of Things Journal*, 2023.

[28] H. Zhou, K. Jiang, X. Liu, X. Li, and V. C. Leung, "Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1517–1530, 2021.

[29] X. Chen, W. Dai, W. Ni, X. Wang, S. Zhang, S. Xu, and Y. Sun, "Augmented deep reinforcement learning for online energy minimization of wireless powered mobile edge computing," *IEEE Transactions on Communications*, 2023.

[30] C. Wang, W. Lu, S. Peng, Y. Qu, G. Wang, and S. Yu, "Modeling on energy-efficiency computation offloading using probabilistic action generating," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 681–20 692, 2022.

[31] A. Gao, S. Zhang, Y. Hu, W. Liang, and S. X. Ng, "Game-combined multi-agent DRL for tasks offloading in wireless powered MEC networks," *IEEE Transactions on Vehicular Technology*, 2023.

[32] S. Zhang, H. Gu, K. Chi, L. Huang, K. Yu, and S. Mumtaz, "DRL-based partial offloading for maximizing sum computation rate of wireless powered mobile edge computing network," *IEEE Transactions on Wireless Communications*, vol. 21, no. 12, pp. 10 934–10 948, 2022.

[33] W. Chen, G. Shen, K. Chi, S. Zhang, and X. Chen, "DRL based partial offloading for maximizing sum computation rate of FDMA-based wireless powered mobile edge computing," *Computer Networks*, vol. 214, p. 109158, 2022.

[34] X. Zhou, L. Huang, T. Ye, and W. Sun, "Computation bits maximization in UAV-assisted mec networks with fairness constraint," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 20 997–21 009, 2022.

[35] P. Chen, B. Lyu, S. Gong, H. Guo, J. Jiang, and Z. Yang, "Computational rate maximization for IRS-assisted full-duplex wireless-powered MEC systems," *IEEE Transactions on Vehicular Technology*, 2023.

[36] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[37] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems," *IEEE transactions on wireless communications*, vol. 15, no. 11, pp. 7244–7257, 2016.

[38] K. Chi, Y.-H. Zhu, Y. Li, L. Huang, and M. Xia, "Minimization of transmission completion time in wireless powered communication networks," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1671–1683, 2017.

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

[40] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680–692, 2017.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "Intelligent service deployment policy for next-generation industrial edge networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3057–3066, 2021.

[43] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[44] L. Qian, Y. Wu, F. Jiang, N. Yu, W. Lu, and B. Lin, "NOMA assisted multi-task multi-access mobile edge computing via deep reinforcement learning for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5688–5698, 2020.

[45] F. Fang, K. Wang, Z. Ding, and V. C. Leung, "Energy-efficient resource allocation for NOMA-MEC networks with imperfect CSI," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3436–3449, 2021.

[46] Y. Gao, B. Xia, Y. Liu, Y. Yao, K. Xiao, and G. Lu, "Analysis of the dynamic ordered decoding for uplink NOMA systems with imperfect CSI," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6647–6651, 2018.

[47] Z. Yang, Z. Ding, P. Fan, and G. K. Karagiannidis, "On the performance of non-orthogonal multiple access systems with partial channel information," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 654–667, 2015.

Wenchao Chen received the B.S. degrees from Jiujiang University, Jiujiang, China, in 2020. He currently pursues the Ph.D. degrees in the School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His current research focuses on Internet of Things (IoT), Deep Reinforcement Learning (DRL), and wireless powered sensor networks.

Xinchen Wei received the M.Sc. degree in software engineering from Zhejiang University of Technology, Zhejiang, China, in 2018, and the Ph.D. degree from the Department of Electronics Engineering, University of York, U.K, in 2022. She is currently a lecturer in the School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Her current research interests include non-orthogonal multiple access (NOMA), Internet of Things (IoT), convex optimization techniques, and resource allocation in wireless networks.

Kaikai Chi received the B.S. and M.S. degrees from Xidian University, Xi'an, China, in 2002 and 2005, respectively, and the Ph.D. degree from Tohoku University, Sendai, Japan, in 2009. He is currently a professor in the School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His current research focuses on wireless cellular network, wireless ad hoc network and wireless sensor network. He was the recipient of the Best Paper Award at the IEEE Wireless Communications and Networking Conference in 2008. He has published more than 50 referred technical papers in proceedings and journals like IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, IEEE Transactions on Parallel and Distributed Systems, etc.

Keping Yu (Member, IEEE) received the M.E. and Ph.D. degrees from the Graduate School of Global Information and Telecommunication Studies, Waseda University, Japan, in 2012 and 2016, respectively. He was a Research Associate, Junior Researcher, Researcher with the Global Information and Telecommunication Institute, Waseda University, from 2015 to 2019, 2019 to 2020, 2020 to 2022, respectively. He is currently an Associate Professor, the Vice Director of Institute of Integrated Science and Technology, and the Director of the Network Intelligence and Security Laboratory, Hosei University and a Visiting Scientist at the RIKEN Center for Advanced Intelligence Project, Japan. Dr. Yu has hosted and participated in more than ten projects, is involved in many standardization activities organized by ITU-T and ICNRG of IRTF, and has contributed to ITU-T Standards Y.3071 and Supplement 35. He received the IEEE Outstanding Leadership Award from IEEE BigDataSE 2021, the Best Paper Award from IEEE Consumer Electronics Magazine Award 2022 (1st Place Winner), IEEE ICFTIC 2021, ITU Kaleidoscope 2020, the Student Presentation Award from JSST 2014. He has authored more than 200 peer-review research papers and books, including over 80 IEEE/ACM Transactions papers. He is an Associate Editor of IEEE Open Journal of Vehicular Technology, Journal of Intelligent Manufacturing, Journal of Circuits, Systems and Computers, and IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. He has been a Guest Editor for more than 20 journals such as IEEE Transactions on Computational Social Systems, IEEE Journal of Biomedical and Health Informatics, and Renewable & Sustainable Energy Reviews. He served as general co-chair and publicity co-chair of the IEEE VTC2020-Spring 1st EBTSRA workshop, general co-chair of IEEE ICCC2020 2nd EBTSRA workshop, general co-chair of IEEE TrustCom2021 3nd EBTSRA workshop, sessio.

Amr Tolba received the M.Sc. and Ph.D. degrees from the Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Egypt, in 2002 and 2006, respectively. He is currently a Full Professor in computer science with King Saud University (KSU), Saudi Arabia. He has authored/coauthored over 180 scientific articles in top-ranked (ISI) international journals, such as IEEE INTERNET OF THINGS JOURNAL, ACM TOIT, IEEE SYSTEMS JOURNAL, etc. He served as a TPC Member at several conferences, such as DSIT 2022, CICA2022, EAI MobiHealth 2021, DSS 2021, etc. He has been included in the list of the top 2% of influential researchers globally (prepared by scientists from Stanford University, USA) during the calendar years 2020, 2021, 2022, and 2023, respectively. His main research interests include artificial intelligence (AI), the Internet of Things (IoT), data science, and cloud computing.

Shahid Mumtaz (Senior Member, IEEE) received the master's and Ph.D. degrees in electrical and electronic engineering from the Blekinge Institute of Technology, Karlskrona, Sweden, and University of Aveiro, Aveiro, Portugal, in 2006 and 2011, respectively. He has more than 12 years of wireless industry/academic experience. Since 2011, he has been with the Instituto de Telecomunicações, Aveiro, Portugal, where he currently holds the position of Auxiliary Researcher and adjunct positions with several universities across the Europe-Asian Region. He is currently also a Visiting Researcher with Nokia Bell Labs, Murray Hill, NJ, USA. He is the author of 4 technical books, 12 book chapters, and more than 150 technical papers in the area of mobile communications. Dr. Mumtaz is an ACM Distinguished Speaker, Editor-in-Chief for IET Journal of Quantum Communication, Vice Chair of Europe/Africa Region IEEE ComSoc: Green Communications and Computing society, and Vice Chair for IEEE standard on P1932.1, Standard for Licensed/Unlicensed Spectrum Interoperability in Wireless Mobile Networks.

Mohsen Guizani (Fellow, IEEE) received the BS (with distinction), MS and PhD degrees in Electrical and Computer engineering from Syracuse University, Syracuse, NY, USA in 1985, 1987 and 1990, respectively. He is currently a Professor of Machine Learning at the Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE. Previously, he worked in different institutions in the USA. His research interests include applied machine learning and artificial intelligence, smart city, Internet of Things (IoT), intelligent autonomous systems, and cybersecurity. He became an IEEE Fellow in 2009 and was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2019, 2020, 2021 and 2022. Dr. Guizani has won several research awards including the "2015 IEEE Communications Society Best Survey Paper Award", the Best ComSoc Journal Paper Award in 2021 as well 5 Best Paper Awards from ICC and Globecom Conferences. He is the author of 11 books, more than 1000 publications and several US patents. He is also the recipient of the 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, the 2018 AdHoc Technical Committee Recognition Award, and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award. He served as the Editor-in-Chief of IEEE Network and is currently serving on the Editorial Boards of many IEEE Transactions and Magazines. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer.