Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.059689

ARTICLE

MATD3-Based End-Edge Collaborative Resource Optimization for NOMA-Assisted Industrial Wireless Networks

Ru Hao[1,2,3], Chi Xu[2,3,*] and Jing Liu[1]

[1]College of Information Engineering, Shenyang University of Chemical Technology, Shenyang, 110142, China

[2]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China

[3]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang, 110016, China

*Corresponding Author: Chi Xu. Email: xuchi@sia.cn

Received: 14 October 2024 Accepted: 05 December 2024 Published: 17 February 2025

ABSTRACT

Non-orthogonal multiple access (NOMA) technology has recently been widely integrated into multi-access edge computing (MEC) to support task offloading in industrial wireless networks (IWNs) with limited radio resources. This paper minimizes the system overhead regarding task processing delay and energy consumption for the IWN with hybrid NOMA and orthogonal multiple access (OMA) schemes. Specifically, we formulate the system overhead minimization (SOM) problem by considering the limited computation and communication resources and NOMA efficiency. To solve the complex mixed-integer nonconvex problem, we combine the multi-agent twin delayed deep deterministic policy gradient (MATD3) and convex optimization, namely MATD3-CO, for iterative optimization. Specifically, we first decouple SOM into two sub-problems, i.e., joint sub-channel allocation and task offloading sub-problem, and computation resource allocation sub-problem. Then, we propose MATD3 to optimize the sub-channel allocation and task offloading ratio, and employ the convex optimization to allocate the computation resource with a closed-form expression derived by the Karush-Kuhn-Tucker (KKT) conditions. The solution is obtained by iteratively solving these two sub-problems. The experimental results indicate that the MATD3-CO scheme, when compared to the benchmark schemes, significantly decreases system overhead with respect to both delay and energy consumption.

KEYWORDS

Industrial wireless networks (IWNs); multi-access edge computing (MEC); non-orthogonal multiple access (NOMA); task offloading; resource allocation

1 Introduction

The progression of wireless communication technology facilitates Industry 4.0, leveraging industrial wireless networks (IWNs) to boost the efficiency of conventional industries [1,2]. An increasing number of industrial end devices (iEDs) require instant processing for latency-sensitive and computational demanding tasks, driving the demand for advanced computation and communication resources. In traditional cloud computing, tasks are offloaded to cloud servers to compensate for the scarcity of

computing resources. However, the quantity of data produced by numerous iEDs, when uploaded to the cloud server for computing, imposes a considerable strain on the network load. Thus, multi-access edge computing (MEC) has become a viable solution and is widely adopted in IWNs [3,4]. The deployment of MEC servers within industrial base stations (iBSs) effectively mitigates device resource constraints and significantly alleviates excessive pressure on network load [5].

While MEC is an effective solution to the lack of computation resources in iEDs, task offloading to MEC servers introduces latency and energy concerns. Traditional orthogonal multiple access (OMA) techniques are constrained by limited orthogonal communication resources, which largely hampers the number of devices that can be accommodated for industrial mission offloading [6,7]. Besides, non-orthogonal multiple access (NOMA) technology significantly mitigates delay and energy consumption issues. By enabling simultaneous data transmission among devices within the same resource block and using successive interference cancellation (SIC) to separate signals, NOMA surpasses OMA in supporting more devices [8]. Nevertheless, considering the constrained computation and communication resources, it is challenging to efficiently utilize the resources to reduce task processing latency and energy consumption in NOMA-assisted MEC systems. Alternatively, most current research leans heavily on centralized scheduling strategies. In contrast, multi-agent deep reinforcement learning (MADRL) is a promising distributed approach applied in IWNs for enhanced decision-making [9].

Existing literature often adopts single OMA or NOMA for MEC systems, where optimization theory or single-agent deep reinforcement learning (DRL) is employed for resource allocation. In contrast, this paper combines OMA and NOMA technology and employs multi-agent DRL (MADRL) for joint task offloading and resource allocation. Specifically, this paper constructs an IWN model based on hybrid multiple access schemes with respect to OMA and NOMA, and formulates the system overhead minimization (SOM) problem. Accordingly, we combine the multi-agent twin delayed deep deterministic policy gradient (MATD3) and convex optimization, namely MATD3-CO for iterative optimization.

The key achievements in this paper can be summarized as follows:

- We study an end-edge collaborative computing scenario for IWNs with hybrid multiple access schemes with respect to OMA and NOMA, where iEDs covered by multiple iBSs share the total system bandwidth resources. Task offloading rate is improved by considering the factors affecting NOMA efficiency, i.e., sub-channel allocation, intra-edge interference, and inter-edge interference.
- With full consideration of resources and NOMA efficiency, the SOM problem is formulated in terms of sub-channel allocation, task offloading ratio and computation resource allocation. To tackle this mixed-integer non-convex problem, we divide it into two sub-problems, i.e., joint sub-channel allocation and task offloading sub-problem, and computation resource allocation sub-problem.
- To approximate the optimal solution, we propose the MATD3-CO scheme. Specifically, we employ MATD3 to optimize the joint sub-channel allocation and task offloading sub-problem due to its non-convexity. Furthermore, we employ convex optimization to solve the computation resource allocation sub-problem, and derive the closed-form expression by the Karush-Kuhn-Tucker (KKT) condition. The solution is obtained by iteratively solving these two sub-problems.

The remaining work of this paper is as follows. Section 2 presents the related works. Section 3 describes the NOMA-assisted system model. Section 4 establishes the SOM problem. Section 5

proposes the MATD3-CO scheme. Section 6 evaluates the experimental outcomes, and Section 7 summarizes this work.

2 Related Work

In recent years, numerous studies offered solutions to the technical hurdles faced by NOMA-assisted MEC systems. For instance, Ding et al. introduced a generic hybrid NOMA-MEC offloading strategy. They considered two special OMA and pure NOMA offloading scenarios to minimize both delay and energy consumption in MEC offloading process [10]. Albogamy et al. devised an efficient conjugate gradient-based approach for optimizing downlink power allocation in NOMA systems, maximizing the maximum weighted sum rate across users [11]. Besides, Muhammed et al. explored the intricacies of allocating resources in downlink NOMA networks. They significantly improved the energy efficiency of the system based on strict power constraints and quality of service (QoS) criteria [12]. Huang et al. assessed the effectiveness of three MEC offloading approaches: pure OMA, pure NOMA, and hybrid NOMA. Their findings highlighted the potential of hybrid NOMA in decreasing the energy consumption associated with MEC offloading [13]. Fang et al. developed a binary search algorithm to minimize delay in multi-user NOMA-assisted MEC systems through power allocation of data transmission [14]. Wu et al. optimized task allocation and resource scheduling in NOMA-MEC networks to minimize energy consumption, considering task delays and server capabilities [15]. Ding et al. optimized both power and time allocation simultaneously to lower the energy expenditure for computation offloading. Closed-form solutions guided the choice between OMA, pure NOMA, or hybrid NOMA [16]. Indeed, Wu et al. devised a distributed and efficient algorithm to reduce overall computation task latency by optimizing both offloaded workloads and transmission time concurrently in NOMA networks [17]. Pham et al. introduced collaborative game theory to resource optimization in NOMA MEC networks, minimizing both computational overhead through cooperative and competitive approaches [18]. Xu et al. developed a DRL-based multi-priority offloading strategy to minimize delay for MEC-enhanced IWNs with high-concurrent heterogeneous industrial tasks [19].

Moreover, MADRL [20] performed well in fully cooperative, competitive and mixed relationship scenarios. Yu et al. proposed a multi-agent deep deterministic policy gradient (MADDPG) framework to minimize the cost of task processing latency and device energy consumption by optimizing UAV routes and IoT device offloading decisions [21]. Luo et al. suggested a framework for mobile crowd computing in networks, utilizing physical layer security and MATD3 algorithm to optimize task offloading and assignment while minimizing computing costs [22]. Xu et al. proposed a digital twin-driven collaborative optimization scheme based on MADRL, which minimizes task processing time by optimizing task offloading ratio, power allocation, and resource allocation [23]. On this basis, Xu et al. further considered cache storage and blockchain consensus for digital twin-assisted MEC networks [24]. Cai et al. introduced a computation offloading method based on MADRL, aimed at meeting the diverse needs of various tasks in heterogeneous systems [25]. Li et al. proposed a MADRL framework aimed at minimizing weighted energy consumption in MEC networks, addressing communication and computation uncertainties by optimizing UAV trajectory, task partition, and resource allocation [26]. Xiao et al. developed a collaborative algorithm based MADRL aimed at optimizing resource allocation in MEC networks, surpassing existing mainstream methods in multiple aspects [27]. Cao et al. proposed a new MADRL scheme to support multi-channel communication and task computation in Industry 4.0 of MEC, significantly reducing task processing time and improving channel utilization [28]. Zhou et al. proposed a MADRL framework for collaborative optimization in

MEC, which improves system performance, learning efficiency, and reliability by 50% through joint optimization of beam forming and offloading strategies [29].

3 System Model

3.1 Network Model

Fig. 1 depicts the IWN with NOMA and OMA to support process monitoring and industrial control. There are M iEDs and N iBSs, where each iBS is enhanced with an MEC server. The iBSs maintain wired connections to an industrial gateway, facilitating centralized scheduling and comprehensive management. All iEDs follow the coordination of iBSs to offload tasks on the total frequency band, which is partitioned into K equal-bandwidth orthogonal sub-channels. The iEDs select the appropriate sub-channels to transmit the signals, which travel through them and interference to reach the iBSs.

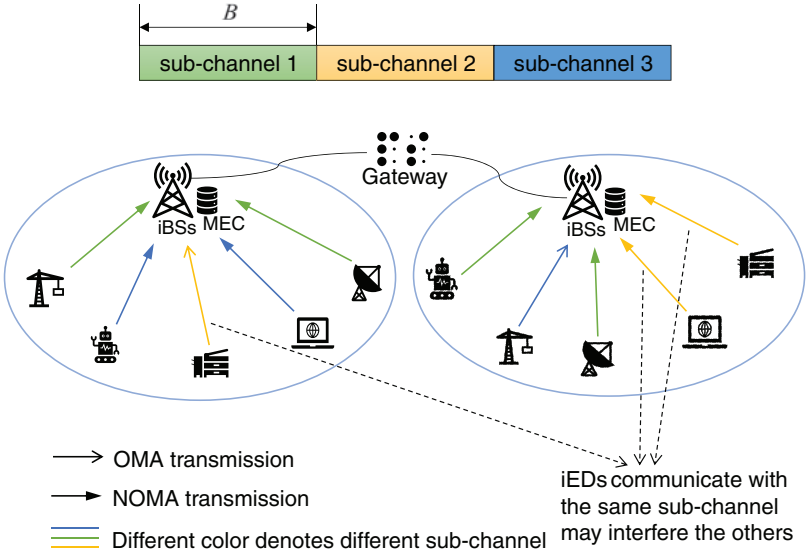


Figure 1: System model

Let time be slotted, denoted by t, with the set of slot indices represented by $\mathcal{T} = \{1, \ldots, t, \ldots, T\}$. At the beginning moment of each episode $\mathcal{T}$, all iEDs are equally assigned to iBSs to cover. The collection of iBSs is referred to $\mathcal{N} = \{1, \ldots, n, \ldots, N\}$, and iBS_n $(n \in \mathcal{N})$ is characterized by a two-tuple $n = (F_n, U_n)$, where F_n and U_n denote the maximum computation resources and communication range, respectively. The collection of iEDs is denoted as $\mathcal{M} = \{1, \ldots, m, \ldots, M\}$, and iED_m $(m \in \mathcal{M})$ generates a task k_m at each time t. The task is characterized by a two-tuple $k_m = (D_m, C_m)$, where D_m and C_m denote the task size and the CPU cycles for processing one byte of data, respectively. All iEDs divide their tasks in local and edge computing. The set of iEDs covered by the iBS_n radio is denoted as $\mathcal{M}_n = \left\{m \middle| d_{m,n} \leq U_n, \forall m \in \mathcal{M}\right\}, \mathcal{M}_n \in \mathcal{M}$, where $d_{m,n}$ denotes the distance between iED_m and iBS_n. Each iED can only be connected to an iBS whose radio covers it. The set of orthogonal sub-channels is referred to $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$, and the bandwidth of the sub-channels is B. All iBSs can be switched between OMA or NOMA for each allocated sub-channel. Each iED selects an orthogonal sub-channel to offload some of its tasks. Using the power-domain NOMA technique, a sub-channel can accommodate multiple iEDs with simultaneous offloading tasks.

3.2 Communication Model

The communication model is constructed based on the NOMA principle, where the sub-channel allocation decision is denoted as

$$o_{m,n}^{k} \in \{0,1\}, \tag{1}$$

where $o_{m,n}^{k} = 1$ indicates iED_m ($m \in \mathcal{M}_n$) selects sub-channel k to offload part of the task to iBS_n. Each iED can select only one sub-channel for data transmission, and the number of iEDs accessing the same sub-channel is limited to $M_{\max}$ to control SIC complexity [30]. Thus, we have

$$\sum_{\forall k \in \mathcal{K}} o_{m,n}^{k} = 1, \tag{2}$$

$$\sum_{\forall m \in \mathcal{M}_n} o_{m,n}^{k} \leq M_{\max}. \tag{3}$$

Let the transmission power of iED_m at time t denoted as p_m with $0 < p_m < P_{\max}$ be predefined. Then, the channel gain between iED_m and iBS_n at t is denoted as

$$h_{m,n}^{k} = \frac{\eta_{m,n}^{k}}{d_{m,n}^{\varphi}}, \tag{4}$$

where $\eta_{m,n}^{k}$ is the Rayleigh distributed small-scale fading, i.e., $\eta_{m,n}^{k} \sim \mathcal{CN}(0,1)$, and φ is the large-scale path loss index. The set of iEDs with instantaneous channel conditions worse than iED_m ($m \in \mathcal{M}_n$) is designated as

$$\mathcal{M}_{m,n,k} = \left\{ m' \left| \left| h_{m',n}^{k} \right|^2 < \left| h_{m,n}^{k} \right|^2, \forall m' \in \mathcal{M}_n \right. \right\}. \tag{5}$$

According to the NOMA principle, iBS_n employs SIC to cancel the signals from iEDs that possess superior channel conditions relative to iED_m.

Let $q_{n,k}$ be the number of iEDs associated with iBS_n accessing the same sub-channel k. If OMA is selected, then $q_{n,k} = 1$ and these iEDs suffer from inter-edge interference. If NOMA is selected, then $q_{n,k} > 1$ and these iEDs suffer from inter-edge and intra-edge interference. When iED_m offloads its task to iBS_n, the received signal-to-interference-plus-noise ratio (SINR) is

$$\phi_{m,n} = \sum_{\forall k \in \mathcal{K}} o_{m,n}^{k} \frac{p_m \left| h_{m,n}^{k} \right|^2}{I_{m,n,k}^{intra} + I_{m,n,k}^{intre} + N_0}, \tag{6}$$

where N_0 is the noise power, $I_{m,n,k}^{intra} = \sum_{\forall m' \in \mathcal{M}_{m,n,k}} o_{m',n}^{k} p_{m'} \left| h_{m',n}^{k} \right|^2$ is the intra-edge interference, and $I_{m,n,k}^{intre} = \sum_{\forall n' \in \mathcal{N} \setminus \{n\}} \sum_{\forall m' \in \mathcal{M}_{n'}} o_{m',n'}^{k} p_{m'} \left| h_{m',n}^{k} \right|^2$ denotes the inter-edge interference, which equals the sum of the product of the power allocated by the other iBSs to other iEDs using sub-channel k and the channel gain between other iEDs and iBS_n. Thus, the uplink data rate of iED_m transmitting to iBS_n can be described as

$$r_{m,n} = B \log_2 \left(1 + \phi_{m,n} \right). \tag{7}$$

3.3 Computing Model

Due to the constrained computation resources of iEDs, some tasks are offloaded to iBSs for edge computing. The task offloading ratio from iED_m ($m \in \mathcal{M}_n$) to iBS_n is denoted as

$$0 \leq v_m \leq 1, \tag{8}$$

where $v_m = 0$, $v_m \in (0, 1)$, and $v_m = 1$ indicate none offloading, partial offloading, and total offloading, respectively.

3.3.1 Local Computing

Let f_m^l be the computing power of iED_m. The time required for partially completing a task at iED_m is calculated as

$$T_m^l = (1 - v_m) \frac{C_m D_m}{f_m^l}. \tag{9}$$

Then, the energy consumption while computing the task partially at iED_m is [31]

$$E_m^l = (1 - v_m) D_m \kappa_m \left(f_m^l\right)^2, \tag{10}$$

where κ_m is the energy coefficient of iED_m.

3.3.2 Computation Offloading

In the computation offloading case, the primary components of the time needed to finish the computation task of iED_m are the transmission delay T_m^{up} and computation delay T_m^{com}. Let $f_{m,n}$ be the computation resources allocated by iBS_n to iED_m, and the overall computation resources allocated by iBS_n must not exceed its maximum computation capacity. Thus, we have

$$f_{m,n} > 0, \tag{11}$$

$$\sum\nolimits_{\forall m \in \mathcal{M}_n} f_{m,n} \leq F_n. \tag{12}$$

The completion time for iED_m during computation offloading can be calculated as

$$T_m^e = T_m^{up} + T_m^{com} = \frac{v_m D_m}{r_{m,n}} + \frac{v_m C_m D_m}{f_{m,n}}. \tag{13}$$

The energy consumption E_m^e of iED_m for transmitting its computation task to iBS_n is [31]

$$E_m^e = \frac{p_m}{\zeta_m} T_m^{up} = \frac{p_m}{\zeta_m} \frac{v_m D_m}{r_{m,n}}, \tag{14}$$

where ζ_m is the power amplifier efficiency of iED_m.

4 Problem Formulation and Transformation

4.1 Problem Formulation

With the above system model, we further formulate the SOM problem. Firstly, we designate the system overhead as the weight sum of latency and energy consumption as

$$C_m = \alpha \max\left(T_m^l, T_m^e\right) + \beta \left(E_m^l + E_m^e\right) = \alpha T_m + \beta E_m, \tag{15}$$

where α and β represent the weight factors for delay and energy consumption, respectively.

Then, the objective function is formulated by

$$\text{SOM}: \min_{\mathcal{O},\mathcal{V},\mathcal{F}} f = \sum_{\forall t\in\mathcal{T}} \sum_{\forall n\in\mathcal{N}} \sum_{\forall m\in\mathcal{M}_n} C_m,$$

$$\text{s.t. } (1), (2), (3), (8), (11), (12). \tag{16}$$

Herein, $\mathcal{O}$ is set of the sub-channel allocation, $\mathcal{V}$ is set of the task offloading ratio, and $\mathcal{F}$ is set of the computation resource allocation. The result of end-edge resource allocation is denoted as $(\mathcal{O},\mathcal{V},\mathcal{F})$, which is represented by

$$\begin{cases} \mathcal{O} = \left\{o_{m,n}^k, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}_n, \forall n \in \mathcal{N}\right\}, \\ \mathcal{V} = \left\{v_m, \forall m \in \mathcal{M}\right\}, \\ \mathcal{F} = \left\{f_{m,n}, \forall m \in \mathcal{M}_n, \forall n \in \mathcal{N}\right\}, \end{cases} \tag{17}$$

where $o_{m,n}^k$ is an integer variable that iED can only select one sub-channel to offload its tasks to iESs, and the total number of iEDs sharing a single sub-channel does not exceed $M_{\max}$.

In the SOM problem, constraints (1)–(3) state that the sub-channel allocation, constraint (8) declares the proportion of tasks to be offloaded, and constraints (11) and (12) require that the overall allocation of computation resources cannot exceed the computation power of iESs.

4.2 Problem Transformation

Obviously, SOM is a mixed integer optimization problem that is NP-hard and typically requires exponential time complexity to find the optimal solution [32]. Moreover, since system dynamics over time generate a myriad of system states, it is challenging to implement one-time optimization strategies in practice. Thus, we reformulate the original problem as follows:

First, the SOM problem can be rewritten as

$$\min_{\mathcal{O},\mathcal{V},\mathcal{F}} f = \sum_{\forall t\in\mathcal{T}} \sum_{\forall n\in\mathcal{N}} \sum_{\forall m\in\mathcal{M}_n} \alpha \max\left((1-v_m^*)\frac{C_m D_m}{f_m^l}, o_{m,n}^{k\,*} v_m^* \left(\frac{D_m}{r_{m,n}} + \frac{C_m D_m}{f_{m,n}}\right)\right) + \beta E_m^*, \tag{18}$$

where $E_m^* = (1-v_m^*) D_m \kappa_m \left(f_m^l\right)^2 + o_{m,n}^{k\,*} v_m^* p_m D_m / \zeta_m r_{m,n}$ is the energy consumption affected by the sub-channel allocation decision $o_{m,n}^{k\,*}$ and the task offloading ratio v_m^*.

By analyzing (18), we find that the system overhead depends on the computation resource allocation $f_{m,n}$ when the sub-channel allocation decision $o_{m,n}^{k\,*}$ and the task offloading ratio v_m^* are fixed. Further, constraints (11) and (12) limit the computation resource allocation variables and are separable from the other constraints. Thus, we can decompose SOM into two separated sub-problems, where the complexity of each sub-problem can be significantly reduced. We conduct independent analysis and validation of each sub-problem, utilizing the advantages of different optimization methods to obtain the final solution. Thus, the SOM is decoupled into two sub-problems at each time slot, i.e., $\mathcal{P}1$ and $\mathcal{P}2$.

4.2.1 Joint Sub-Channel Allocation and Task Offloading Sub-Problem

$\mathcal{P}1$ involves sub-channel allocation decisions and task offloading ratios for all iEDs. It is defined as

$$\mathcal{P}1: \min_{\mathcal{O}^t,\mathcal{V}^t} g_1 = \sum_{\forall n\in\mathcal{N}} \sum_{\forall m\in\mathcal{M}_n} C_m,$$

$$\text{s.t. } (1), (2), (3), (8). \tag{19}$$

4.2.2 Computation Resource Allocation Sub-Problem

$\mathcal{P}2$ involves computation resource allocation for all iEDs. Given the results obtained from $\mathcal{P}1$, $\mathcal{P}2$ can be expressed as

$$\mathcal{P}2: \min_{\mathcal{F}^t} g_2 = \sum_{\forall n\in\mathcal{N}} \sum_{\forall m\in\mathcal{M}_n} v_m \frac{\alpha C_m D_m}{f_{m,n}},$$

$$\text{s.t. } (11), (12). \tag{20}$$

5 The Proposed MATD3-CO Scheme

With the decoupled sub-problems, we develop MATD3-CO to approach the optimal solution. Fig. 2 depicts the framework for addressing the SOM problem.

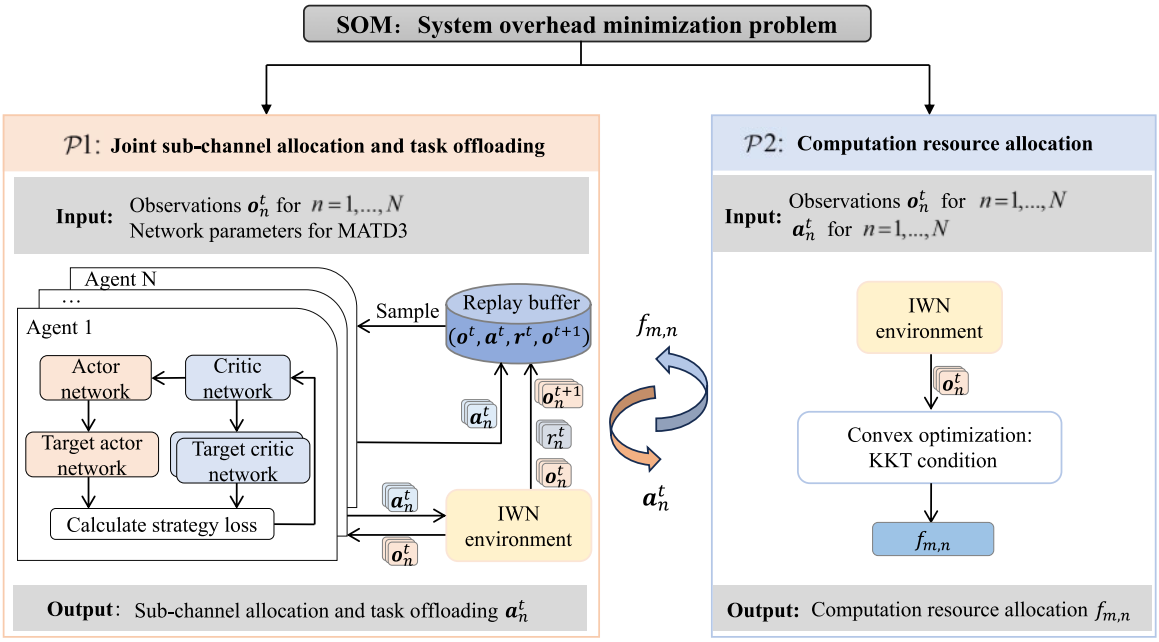


Figure 2: The framework to solve the SOM problem

For $\mathcal{P}1$, we employ multi-agent MDP to transform the SOM problem for the MADRL solution, and propose MATD3 to solve the sub-channel allocation and task offloading ratio. For $\mathcal{P}2$, we use convex optimization to obtain the computation resource allocation, where the closed-form solution is obtained by the KKT condition. The two algorithms are iteratively solved to approximate the optimal solution.

5.1 Joint Sub-Channel Allocation and Task Offloading Based on MATD3

5.1.1 Multi-Agent MDP Modeling

$\mathcal{P}1$ is a mixed integer non-linear programming problem, which is NP-hard [33]. Thus, we utilize multi-agent MDP to reformulate the sub-problem for a MADRL-based scheme.

In the IWN, we treat each iBS as an agent for its decision-making in order to obtain the minimum system overhead. The detailed three key elements of agent n are constructed as follows:

1) Observation: At each time slot t, since the agents can only observe part of the whole environment, the observations of agent n can be described as

$$\boldsymbol{o}_n^t = \left\{\boldsymbol{d}_{\mathcal{M}_n}, \mathrm{D}_{\mathcal{K}_n}, \mathrm{F}_{\mathcal{K}_n}, \mathrm{T}_{\mathcal{K}_n}\right\}, \tag{21}$$

where $\boldsymbol{d}_{\mathcal{M}_n}$ represents the distance between iBS_n and iED_m ($m \in \mathcal{M}_n$), $\mathrm{D}_{\mathcal{K}_n}$, $\mathrm{F}_{\mathcal{K}_n}$ and $\mathrm{T}_{\mathcal{K}_n}$ are the data size, CPU cycles required to process one bit of data and the final task duration of task k_m ($m \in \mathcal{M}_n$), respectively. Furthermore, the state space of N agents at the time slot t is $\boldsymbol{o}^t = \left\{\boldsymbol{o}_1^t, \ldots, \boldsymbol{o}_n^t, \ldots, \boldsymbol{o}_N^t\right\}$.

2) Action: According to the current observations, each agent decides its sub-channel and offloading ratio strategy. The actions of agent n at each time slot t is

$$\boldsymbol{a}_n^t = \left\{o_{m,n}^k, v_m \,|\forall m \in \mathcal{M}_n, \forall k \in \mathcal{K}\right\}. \tag{22}$$

The set of actions for N agents is denoted as $\boldsymbol{a}^t = \left\{\boldsymbol{a}_1^t, \ldots, \boldsymbol{a}_n^t, \ldots, \boldsymbol{a}_N^t\right\}$.

3) Reward Function: In the multi-agent MDP scenario, N agents interact with the environment, cooperating based on the state and policy to obtain an individual reward r_n^t. SOM aims to minimize the system overhead within resources constraints. Thus, the reward of agent n is the negative of the sum of the computation overhead of its overlay devices at time slot t.

$$r_n^t = -\sum_{\forall m \in \mathcal{M}_n} C_m. \tag{23}$$

The set of rewards of N agents is $\boldsymbol{r}^t = \left\{r_1^t, \ldots, r_n^t, \ldots, r_N^t\right\}$.

In MATD3, each agent aims to maximize its expected reward, represented by $R_n^t = \sum_{i \geq 0} \gamma^i r_n^{t+i}$, where γ is the discount, $0 \leq \gamma \leq 1$.

5.1.2 Algorithmic Structure

To track the MDP, we employ DRL algorithm. First, TD3 is an improved version of deep deterministic policy gradient (DDPG), with enhanced stability and convergence compared to DDPG. Unlike DDPG, which only uses one Q-value network, the key of TD3 is the introduction of two independent critics trained to minimize the squared error between two different Q-values. Meanwhile, TD3 also incorporates techniques such as delay policy updates and target network smoothing, further enhancing the stability of the algorithm. By extending the TD3 to multi-agent environments, namely MATD3, we can facilitate the mean overestimation and error accumulation problems in the MADDPG. For this purpose, we use centralized training and distributed execution method, where centralized training enhances collaboration between agents by obtaining global data, while trained distributed execution agents make independent decisions based on local data.

Specifically, each iBS is considered as an agent that maintains its networks and learns individual policies based on local observations. Each agent comprises the following DNNs: a DNN implementing

the actor π_ϕ for selecting actions, two DNNs implementing critics Q_{θ_1} and Q_{θ_2} for estimating the Q-values of the chosen action, a target actor $\pi_{\phi'}$, and two target critics $Q_{\theta_1'}$ and $Q_{\theta_2'}$. Here, ϕ, θ_1, θ_2, ϕ', θ_1' and θ_2' represent their corresponding network weights.

5.1.3 Algorithm Training

MATD3 is centralizedly trained by the industrial gateway, as summarized in Algorithm 1. Specifically, within this framework, critics of each agent are managed by the industrial gateway, which enables them to access global state and action information while ensuring full visibility of this information for all agents. All iBSs act as agents to perform actions through a decentralized observation mode, collect experience in interacting with the environment, and upload this experience to the industrial gateway for centralized training. The industrial gateway processes this data to calculate gradients, and then feeds back the updated model parameters to iBSs.

Algorithm 1: MATD3 training for $\mathcal{P}1$

Input: observation $\boldsymbol{o}_n^t$ of agent n, $\forall n \in \mathcal{N}$;
Output: ϕ_n' of agent n, $\forall n \in \mathcal{N}$;

```
Initialize the AC networks with the parameters φ_n, θ_{n,1}, θ_{n,2}, φ'_n, θ'_{n,1}, θ'_{n,2} for ∀n ∈ N;
Initialize the replay buffer B_n;
for episode = 1 : E do
    Initialize the observations of all agents;
    for t = 1 : T do
        for each agent n do
            Agent n selects action with exploration noise a_n^t = π_φ(o_n^t) + ε;
            Execute a_n^t, compute reward r_n^t and obtain o_n^{t+1};
            Store (o^t, a^t, r_n^t, o^{t+1}) as an experience in B_n;
            Sample a random mini-batch of U samples from B_n;
            According to (25), update critics;
            if t mod t_act then
                According to (26), update the actor network;
            end if
            if t mod t_tar then
                According to (27), update the three target networks;
            end if
        end for
    end for
end for
```

The training process of the neural network alternates with the interaction process. Each agent outputs an action $\boldsymbol{a}_n^t = \pi_\phi\left(\boldsymbol{o}_n^t\right) + \varepsilon$ based on local observations of the environment, where $\varepsilon \sim \mathcal{N}\left(0, \delta^2\right)$ is a noise with 0 mean and a standard deviation δ. By performing the above actions, each agent obtains the next observation $\boldsymbol{o}_n^{t+1}$ and the immediate reward r_n^t and stores the experience $\left(\boldsymbol{o}^t, \boldsymbol{a}^t, r_n^t, \boldsymbol{o}^{t+1}\right)$ into the replay buffer B_n. Once the buffer accumulates a sufficient quantity of experiences, each agent randomly draws a mini-batch of U quaternions from B_n, and $\left(\boldsymbol{o}^u, \boldsymbol{a}^u, \boldsymbol{r}^u, \boldsymbol{o}^{u+1}\right)$ is recorded as the u-th quaternion. The target value Y_n^u for agent n can be expressed as

$$Y_n^u = r_n^u + \gamma \min_{w=1,2} Q_{\theta'_{n,w}}\left(\boldsymbol{o}_n^{u+1}, \boldsymbol{a}^{u+1}\right), \tag{24}$$

where $\boldsymbol{a}^{u+1} = \left\{\boldsymbol{a}_1^{u+1}, \ldots, \boldsymbol{a}_n^{u+1}, \ldots, \boldsymbol{a}_N^{u+1}\right\}$, and $\boldsymbol{a}_n^{u+1}$ is acquired through the target actor network, i.e., $\boldsymbol{a}_n^{u+1} = \pi_{\phi'}\left(o_n^{u+1}\right) + \varepsilon, \varepsilon \sim \text{clip}\left(\mathcal{N}\left(0, \sigma\right), -c, c\right)$. MATD3 comprises two sets of target critic networks with the same network structure, who calculate the Q-values of the subsequent state-action pairs and then select the smaller value to calculate Y_n^u. This strategy effectively solves the over-estimation caused by maximization. The loss function of the critic network is denoted by

$$L\left(\theta_{n,w}\right) = \frac{1}{U}\sum\nolimits_u \left(Y_n^u - Q_{\theta_{n,w}}\left(\boldsymbol{o}_n^u, \boldsymbol{a}^u\right)\right)^2, w = 1, 2. \tag{25}$$

The actor network is updated in a delayed manner, i.e., updating the critic network many times before updating it, with an update cycle of t_{act}. For any Q_θ (e.g., Q_{θ_1}) the actor's parameters are updated via the policy gradient.

$$\nabla_{\phi_n} J\left(\phi_n\right) = \frac{1}{U}\sum\nolimits_u \nabla_{\boldsymbol{a}_n^u} Q_{\theta_{n,1}}\left(\boldsymbol{o}_n^u, \boldsymbol{a}^u\right) \nabla_{\phi_n} \pi_{\phi_n}\left(\boldsymbol{o}_n^u\right). \tag{26}$$

The target network is soft updated with an update cycle of t_{tar}.

$$\begin{aligned} &\theta'_{n,w} = \tau\theta_{n,w} + \left(1-\tau\right)\theta'_{n,w}, w = 1, 2, \\ &\phi'_n = \tau\phi_n + \left(1-\tau\right)\phi'_n, \end{aligned} \tag{27}$$

where $\tau \ll 1$.

5.2 Computation Resource Allocation Based on Convex Optimization

The variables associated with all iBSs are independent. Therefore, we can further divide $\mathcal{P}2$ into multiple problems, each related only to iBS_n, denoted as

$$\begin{aligned} &\mathcal{P}3\text{:} \min_{\mathcal{F}_n^t} g_2^n = \sum_{\forall m \in \mathcal{M}_n} v_m \frac{\alpha C_m D_m}{f_{m,n}}, \\ &\text{s.t. (11), (12),} \end{aligned} \tag{28}$$

where $\mathcal{F}_n^t$ denotes the variable in $\mathcal{F}^t$ associated with iBS_n.

Theorem 1. $\mathcal{P}3$ is a convex optimization problem.

Proof of Theorem 1. The partial derivatives of the objective function g_2^n are obtained using the Hessian matrix. That is

$$\frac{\partial^2 g_2^n}{\partial f_{i,n} \partial f_{j,n}} = \begin{cases} v_i 2\alpha C_i D_i / \left(f_{i,n}\right)^3, i = j, \\ 0, i \neq j, \end{cases} \tag{29}$$

where $v_i 2\alpha C_i D_i / \left(f_{i,n}\right)^3 > 0$. g_2^n corresponds to a positive definite Hessian matrix, then **Theorem 1** holds.

We introduce the Lagrange multipliers λ_n and formulate the Lagrangian as

$$L\left(f_{m,n}, \lambda_n\right) = \sum_{\forall m \in \mathcal{M}_n} v_m \frac{\alpha C_m D_m}{f_{m,n}} + \lambda_n \left(\sum_{\forall m \in \mathcal{M}_n} f_{m,n} - F_n\right). \tag{30}$$

Based on the KKT condition, the following equation is obtained:

$$\begin{cases} \nabla_{\mathcal{F}_n^t} \sum\limits_{\forall m \in \mathcal{M}_n} v_m \frac{\alpha C_m D_m}{f_{m,n}} + \lambda_n \nabla_{\mathcal{F}_n^t} \left(\sum\limits_{\forall m \in \mathcal{M}_n} f_{m,n} - F_n \right) = 0, \\ \sum\limits_{\forall m \in \mathcal{M}_n} f_{m,n} - F_n = 0, \\ \lambda_n \geq 0. \end{cases} \tag{31}$$

Solve the system of equations to get the optimal solution for task k_m computation resource allocation.

$$f_{m,n}{}^* = \frac{\sqrt{\alpha C_m D_m v_m} F_n}{\sum\limits_{\forall m \in \mathcal{M}_n} \sqrt{\alpha C_m D_m v_m}}, \forall m \in \mathcal{M}_n. \tag{32}$$

5.3 Algorithmic Complexity Analysis

Since the computation resource allocation problem has a closed-form solution, the complexity originates primarily from MATD3, which is predominantly influenced by the neural network architecture and the sheer quantity of the parameters. All networks utilize DNNs, and their computation complexity is formulated as

$$O(G) = O\left(\sum\nolimits_{l=1}^{L} d_l d_{l+1} \right), \tag{33}$$

where L is the number of layers, and d_l is the number of neurons in the l-th layer. Hence, the actors have a computation complexity of $O(G_a)$, while the critics have a complexity of $O(G_c)$.

During the intensive training phase, N agents with U experiences are trained for K iterations, with the computation complexity of the actors and critics being $O\left(G_a K U^N\right)$ and $O\left(G_c K U^N\right)$, respectively.

During the phase of distributed implementation, each agent makes independent decisions. The actors' computation complexity is $O(G_a)$.

6 Performance Evaluation

This section evaluates experimentally the performance of MATD3-CO.

6.1 Simulation Settings

6.1.1 Experimental Setup

Let a scalable IWN have different numbers of iBSs and iEDs, where the radio coverage radius of the iBSs is 100 m. All iEDs are evenly distributed over the coverage area and at least 5 m away from iBSs. During the experiment, we set the tasks to be randomly generated within a fixed range. The key parameters are set in Table 1. The configuration for the parameters of DNNs is given as follows. The actor network has 300 and 100 neurons in its first and second hidden layers, respectively. The size of the last layer is adjusted to match the action dimension of iBSs. Meanwhile, two critics adopt a consistent structure, each housing 300, 100 and 1 neurons across their three hidden layers. During the training phase, the learning rate of the actors and critics is $\eta_a = \eta_c = 10^{-3}$, the discount factor is $\gamma = 0.9$, and the batch size is 128. All experiments are operated on Intel i7-11700 CPU and NVIDIA GeForce RTX 3070 GPU with TensorFlow-GPU-1.14.0 and Python-3.7.

Table 1: Key parameters and values

Paraments	Values
Number of iBSs (N)	3∼4
Number of iEDs (M)	15∼75
Computation resource of iBSs (F_n)	100 GHz/s
Computation resource of iEDs (f_m^l)	5 GHz/s
Radio coverage of iBSs (U_n)	$r = 100$ m
Task size (D_m)	100∼5500 bytes
Computation cycles of iEDs (C_m)	0.25 MHz/byte
Transmission power of the iEDs (p_m)	100∼300 mW
Sub-channel bandwidth (B)	1∼5 MHz
Noise power (N_0)	10^{-11} mW
Path loss exponent (φ)	3
Energy coefficient (κ_m)	10^{-28} [34]
Latency trade-off factor (α)	0.7
Energy consumption trade-off factor (β)	0.3

6.1.2 Benchmark Schemes for Comparison

The following experiments involve several benchmark DRL schemes:

- MATD3-CO: the proposed scheme employing MATD3 and convex optimization to iteratively approach the optimal solution.
- MATD3: a scheme based on MATD3 algorithm.
- MADDPG: a scheme based on MADDPG algorithm.
- DDPG: a scheme based on DDPG algorithm.

6.2 Convergence Analysis

Reward measures the effectiveness of DRL-based algorithms. Fig. 3 illustrates the trend of reward convergence for all schemes, highlighting that as the number of iterations increases, the rewards of all schemes gradually increase and converge after a certain number of iterations. Moreover, as demonstrated in Fig. 3, we can see that the two TD3-based schemes converge more stable than the DDPG-based schemes, due to the introduction of two critic networks to address the overestimation problem, as well as the delayed updating of the actor networks. DDPG converges poorly in a multi-agent environment. Additionally, the three MADRL-based schemes perform better than the single-agent DDPG schemes in a multi-agent environment, obtaining higher reward. During the initial 400 episodes, the proposed scheme yields higher reward, primarily attributed to the optimal allocation of computation resources in making system decisions. The proposed MATD3-CO scheme yields the highest reward after convergence and a more stable reward curve.

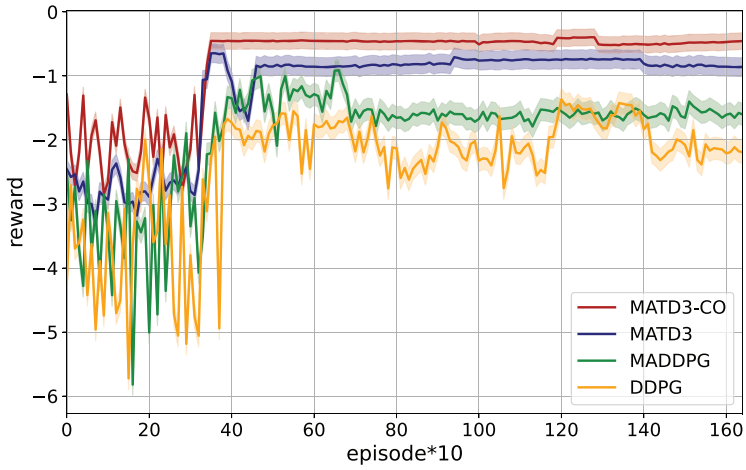
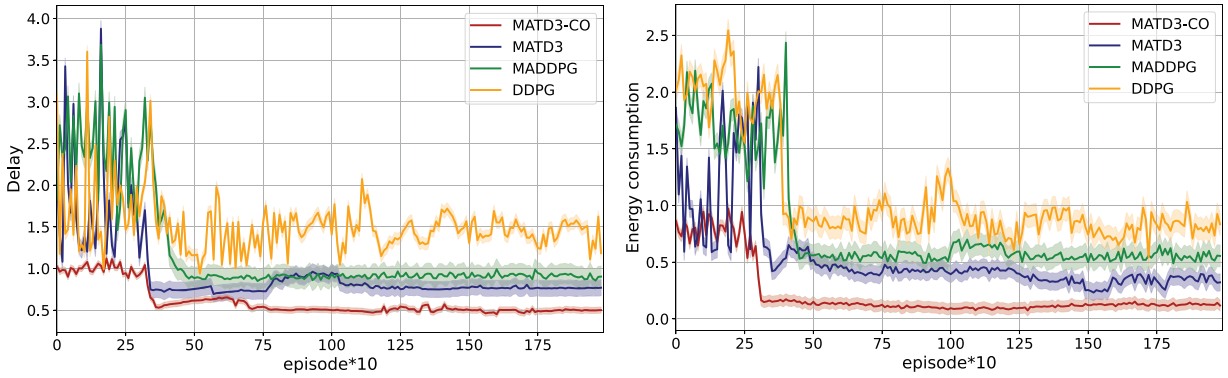


Figure 3: Reward *vs.* number of iterations for data training: $N = 3$, $M = 45$, $B = 5$ MHz, $F_n = 100$ GHz/s, $D_m \sim [150, 1500]$ bytes

Fig. 4 shows the comparison of delay and energy consumption performance with different schemes. The results demonstrate that as the iteration count rises, the delay and energy consumption of all schemes decrease, and converge after a certain number of iterations. However, our proposed solution demonstrates superior performance in both delay and energy consumption. Therefore, MATD3-CO is superior to the other schemes.



(a) Comparison of delay

(b) Comparison of energy consumption

Figure 4: (a) Comparison of delay; (b) Comparison of energy consumption

6.3 Performance Comparison

Fig. 5 depicts the effect of the number of iEDs on system overhead under different schemes. When the number of iEDs is small (e.g., $M = 15$), we can see that there are fewer tasks that need to be computed, so the system overhead of all schemes is almost the same and very small. This is because there are sufficient resources to handle the tasks of iEDs. As the number of iEDs increases, the system costs under all schemes also rise. The reasons for this situation are as follows. When more iEDs are connected to the system, iEDs compete for limited computation resources, resulting in increased computation delay and energy consumption. In addition, due to limited channel resources, more iEDs

sharing the same sub-channel increases the transmission delay and energy consumption for offloading tasks to iBSs. Overall, these factors lead to a regular increase in system overhead. As the number of iEDs increases further, the performance gap between different schemes also widens. Among all schemes, the system overhead of MATD3-CO consistently remains the smallest. This experimental result indicates that MATD3-CO is more suitable for more complex IWN environments.

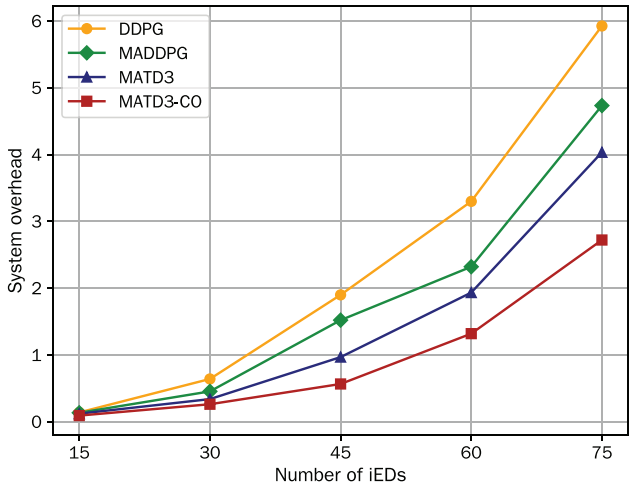


Figure 5: System overhead *vs.* number of iEDs: $F_n = 100$ GHz/s, $B = 5$ MHz, $D_m \sim [150, 1500]$ bytes

Fig. 6 presents how system overhead is affected by the number of iBSs, where the number of iBSs is set to 3 and 4. For a fixed number of iEDs, increasing the number of iESs leads to a slight decrease in system overhead. This trend is consistent across different schemes. The reason is that more iESs only reduce the edge computing delay of tasks, while the energy consumption and offloading delay to iESs remain unchanged, and they account for the percentage of total system overhead. Meanwhile, the system overhead of MATD3-CO is lower than that of MATD3.

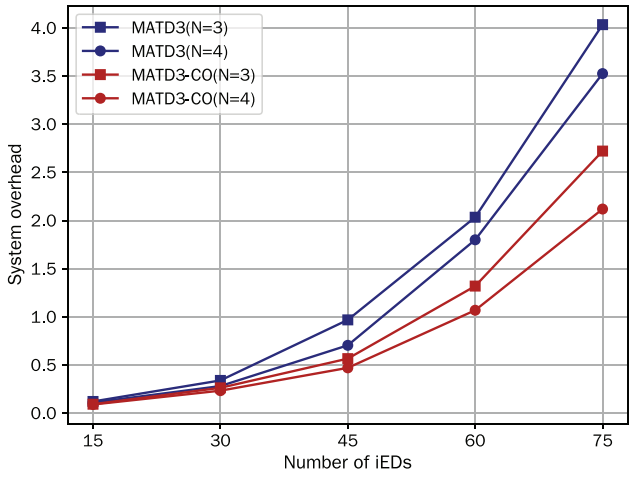


Figure 6: System overhead *vs.* number of iESs: $B = 5$ MHz, $D_m \sim [150, 1500]$ bytes

Fig. 7 illustrates the effect of sub-channel bandwidth on system overhead. In this evaluation, we increase the sub-channel bandwidth in the system from 4 to 20 MHz and set the task size to $D_m \sim [4500, 5500]$ bytes. From the results, we observe that as the total system bandwidth increases, system overhead decreases across all schemes. This is because increasing the bandwidth can effectively increase the transmission rate of the tasks, thereby reducing the transmission delay and energy consumption during offloading. These reductions affect the system overhead in the IWN. The proposed scheme outperforms MATD3, MADDPG, and DDPG across different bandwidth settings.

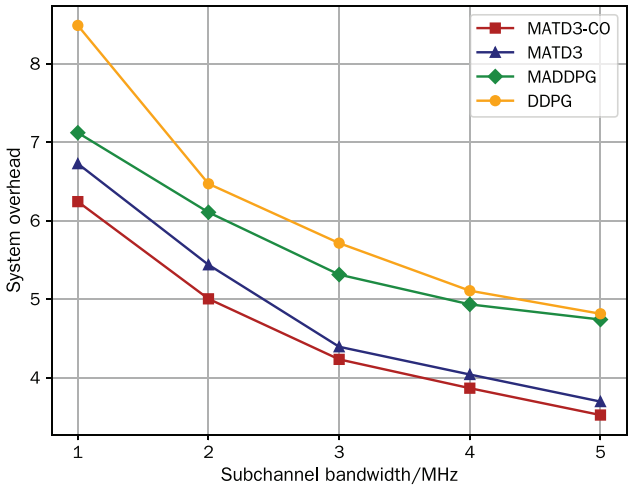


Figure 7: System overhead *vs*. sub-channel bandwidth: $M = 45$, $F_n = 100$ GHz/s, $D_m \sim [4500, 5500]$ bytes

Fig. 8 illustrates the impact of task size changes on system overhead. For this evaluation, the task size D_m follows a uniform distribution, increasing from $D_m \sim [150, 1500]$ bytes to $D_m \sim [4500, 5500]$ bytes. We observe from the results that the system overhead for all schemes increases as the task size grows. This is attributed to the expanded size of task data and the heightened resources demanded for computing tasks. The computation delay, transmission delay and transmission energy consumption increase somewhat. Overall, the system overhead of MATD3-CO is less than MATD3, MADDPG and DDPG.

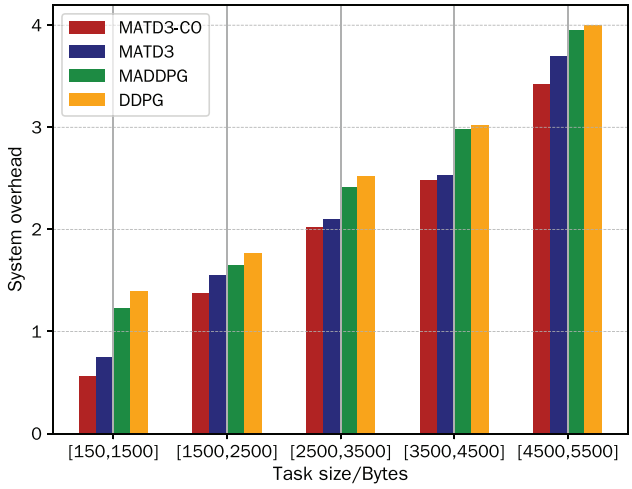


Figure 8: System overhead *vs*. task size: $M = 45$, $F_n = 100$ GHz/s, $B = 5$ MHz

7 Conclusion

In this paper, we investigated the problem of end-edge cooperative resource optimization for IWN with hybrid multiple access scheme. We established communication and computation models under this network model and fully considered the factors affecting the NOMA transmission rate under the constraint of limited computation and communication resources. We defined the weighted sum of task processing delay and energy consumption as the system overhead and formulated the system overhead minimization problem by jointly optimizing the sub-channel allocation, task offloading ratio, and computation resource allocation. To track this problem, we proposed MATD3-CO scheme to iteratively approach the optimal solution. The experimental results demonstrated that the proposed scheme converges well and effectively reduces the system overhead.

Future works will consider more constraints on the computation resource of iESs, the transmission power of iEDs, the cache of iEDs, and so on. Correspondingly, we will apply more algorithms for solutions, such as game-theoretic and other machine learning algorithms.

Acknowledgement: The authors would like to thank the editor and the reviewers for their time and effort in reviewing this paper.

Funding Statement: This work was supported by the National Natural Science Foundation of China under Grants 92267108, 62173322 and 61821005, and the Science and Technology Program of Liaoning Province under Grants 2023JH3/10200004 and 2022JH25/10100005.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Ru Hao, Chi Xu, Jing Liu; analysis and interpretation of results: Ru Hao, Chi Xu; draft manuscript preparation: Ru Hao, Chi Xu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data are contained within this paper.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] C. Xu, H. Yu, X. Jin, C. Xia, D. Li and P. Zeng, "Industrial internet for intelligent manufacturing: Past, present, and future," *Front Inf. Technol. Electron. Eng.*, vol. 25, no. 9, pp. 1173–1192, Sep. 2024. doi: 10.1631/FITEE.2300806.

[2] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014. doi: 10.1109/TII.2014.2300753.

[3] Y. Yu, "Mobile edge computing towards 5G: Vision, recent progress, and open challenges," *China Commun.*, vol. 13, no. S2, pp. 89–99, 2016. doi: 10.1109/CC.2016.7405725.

[4] X. Liu, C. Xu, H. Yu, and P. Zeng, "Multi agent deep reinforcement learning for end-edge orchestrated resource allocation in industrial wireless networks," *Front Inf. Technol. Electron. Eng.*, vol. 23, no. 1, pp. 47–60, Feb. 2022. doi: 10.1631/FITEE.2100331.

[5] P. Guan, X. Deng, Y. Liu, and H. Zhang, "Analysis of multiple clients behaviors in edge computing environment," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 9052–9055, Sep. 2018. doi: 10.1109/TVT.2018.2850917.

[6] J. Ding and J. Cai, "Two-side coalitional matching approach for joint MIMO-NOMA clustering and BS selection in multi-cell MIMO-NOMA systems," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 3, pp. 2006–2021, Mar. 2020. doi: 10.1109/TWC.2019.2961654.

[7] W. Feng *et al.*, "Hybrid beamforming design and resource allocation for UAV-aided wireless-powered mobile edge computing networks with NOMA," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3271–3286, Nov. 2021. doi: 10.1109/JSAC.2021.3091158.

[8] P. Wang, J. Xiao, and L. Ping, "Comparison of orthogonal and non-orthogonal approaches to future wireless cellular systems," *IEEE Veh. Technol. Mag.*, vol. 1, no. 3, pp. 4–11, Sep. 2006. doi: 10.1109/MVT.2006.307294.

[9] S. Hwang, H. Kim, H. Lee, and I. Lee, "Multi-agent deep reinforcement learning for distributed resource management in wirelessly powered communication networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14055–14060, Nov. 2020. doi: 10.1109/TVT.2020.3029609.

[10] Z. Ding, D. Xu, R. Schober, and H. V. Poor, "Hybrid NOMA offloading in multi-user MEC networks," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 7, pp. 5377–5391, Jul. 2022. doi: 10.1109/TWC.2021.3139932.

[11] F. R. Albogamy, M. A. Aiyashi, F. H. Hashim, I. Khan, and B. J. Choi, "Optimal resource allocation for NOMA wireless networks," *Comput. Mater. Contin.*, vol. 74, no. 2, pp. 3249–3261, 2023. doi: 10.32604/cmc.2023.031673.

[12] A. J. Muhammed, Z. Ma, Z. Zhang, P. Fan, and E. G. Larsson, "Energy-efficient resource allocation for NOMA based small cell networks with wireless backhauls," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3766–3781, Jun. 2020. doi: 10.1109/TCOMM.2020.2979971.

[13] W. Huang and Z. Ding, "New insight for multi-user hybrid NOMA offloading strategies in MEC networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2918–2923, Feb. 2024. doi: 10.1109/TVT.2023.3318151.

[14] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng and G. K. Karagiannidis, "Optimal resource allocation for delay minimization in NOMA-MEC networks," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7867–7881, Dec. 2020. doi: 10.1109/TCOMM.2020.3020068.

[15] Y. Wu, B. Shi, L. P. Qian, F. Hou, J. Cai and X. S. Shen, "Energy-efficient multi-task multi-access computation offloading via NOMA transmission for IoTs," *IEEE Trans. Ind. Inform.*, vol. 16, no. 7, pp. 4811–4822, Jul. 2020. doi: 10.1109/TII.2019.2944839.

[16] Z. Ding, J. Xu, O. A. Dobre, and H. V. Poor, "Joint power and time allocation for NOMA-MEC offloading," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6207–6211, Jun. 2019. doi: 10.1109/TVT.2019.2907253.

[17] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12244–12258, Dec. 2018. doi: 10.1109/TVT.2018.2875337.

[18] Q. V. Pham, H. T. Nguyen, Z. Han, and W. J. Hwang, "Coalitional games for computation offloading in NOMA-enabled multi-access edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1982–1993, Feb. 2020. doi: 10.1109/TVT.2019.2956224.

[19] C. Xu, P. Zhang, H. Yu, and Y. Li, "D3QN-based multi-priority computation offloading for time-sensitive and interference-limited industrial wireless networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13682–13693, Sep. 2024. doi: 10.1109/TVT.2024.3387567.

[20] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020. doi: 10.1109/TCYB.2020.2977374.

[21] K. Yu, Q. Cui, Z. Zhang, X. Huang, X. Zhang and X. Tao, "Efficient UAV/satellite-assisted IoT task offloading: A multi-agent reinforcement learning solution," in *2022 27th APCC*, Jeju Island, Republic of Korea, 2022, pp. 83–88.

[22] X. Luo, Y. Liu, H. H. Chen, and Q. Guo, "PHY security design for mobile crowd computing in ICV networks based on multi-agent reinforcement learning," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 10, pp. 6810–6825, Oct. 2023. doi: 10.1109/TWC.2023.3245637.

[23] C. Xu, Z. Tang, H. Yu, P. Zeng, and L. Kong, "Digital twin-driven collaborative scheduling for heterogeneous task and edge-end resource via multi-agent deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3056–3069, Oct. 2023. doi: 10.1109/JSAC.2023.3310066.

[24] C. Xu, P. Zhang, X. Xia, L. Kong, P. Zeng and H. Yu, "Digital twin-assisted intelligent secure task offloading and caching in blockchain-based vehicular edge computing networks," *IEEE Internet Things J.*, pp. 1–16, 2024. doi: 10.1109/JIOT.2024.3482870.

[25] J. Cai, H. Fu, and Y. Liu, "Multitask multi objective deep reinforcement learning-based computation offloading method for industrial internet of things," *IEEE Internet Things J.*, vol. 10, no. 2, pp. 1848–1859, Jan. 2023. doi: 10.1109/JIOT.2022.3209987.

[26] B. Li, R. Yang, L. Liu, J. Wang, N. Zhang and M. Dong, "Robust computation offloading and trajectory optimization for multi-UAV-assisted MEC: A multiagent DRL approach," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 4775–4786, Feb. 2024. doi: 10.1109/JIOT.2023.3300718.

[27] Y. Xiao, Y. Song, and J. Liu, "Collaborative multi-agent deep reinforcement learning for energy-efficient resource allocation in heterogeneous mobile edge computing networks," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 6, pp. 6653–6668, Jun. 2024. doi: 10.1109/TWC.2023.3335597.

[28] Z. Cao, P. Zhou, R. Li, S. Huang, and D. Wu, "Multiagent deep reinforcement learning for joint multichannel access and task offloading of mobile-edge computing in Industry 4.0," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6201–6213, Jul. 2020. doi: 10.1109/JIOT.2020.2968951.

[29] H. Zhou, Y. Long, S. Gong, K. Zhu, D. T. Hoang and D. Niyato, "Hierarchical multi-agent deep reinforcement learning for energy-efficient hybrid computation offloading," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 986–1001, Jan. 2023. doi: 10.1109/TVT.2022.3202525.

[30] Z. Song, Y. Liu, and X. Sun, "Joint radio and computation resource allocation for NOMA-based mobile edge computing in heterogeneous networks," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2559–2562, Dec. 2018. doi: 10.1109/LCOMM.2018.2875984.

[31] V. D. Tuong, T. P. Truong, T. V. Nguyen, W. Noh, and S. Cho, "Partial computation offloading in NOMA-assisted mobile-edge computing systems using deep reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13196–13208, Sep. 2021. doi: 10.1109/JIOT.2021.3064995.

[32] B. Yang, X. Cao, J. Bassey, X. Li, and L. Qian, "Computation offloading in multi-access edge computing: A multi-task learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 9, pp. 2745–2762, Sep. 2021. doi: 10.1109/TMC.2020.2990630.

[33] Y. Yang, L. Chen, W. Dong, and W. Wang, "Active base station set optimization for minimal energy consumption in green cellular networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 11, pp. 5340–5349, Nov. 2015. doi: 10.1109/TVT.2014.2385313.

[34] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018. doi: 10.1109/TWC.2017.2785305.