Annals of Emerging Technologies in Computing (AETiC)
Vol. 10, No. 2, 2026

Research Article

Heterogeneous IoT User Association and Channel Resource Joint Scheduling Method Based on MADDQN and DMADDPG

Yankai Xie
Fujian Polytechnic of Water Conservancy and Electric Power, China
*Correspondence: YankaiXie@outlook.com

Received: 04 February 2026; Accepted: 27 March 2026; Published: 01 April 2026

Abstract: The large-scale access of multiple types of terminals in the heterogeneous IoT makes it difficult to balance system performance and scalability due to the strong coupling relationship between user association and channel resource scheduling. Existing deep reinforcement learning methods still have shortcomings in multi-agent collaborative decision-making, hybrid discrete-continuous action modeling, and dynamic environment adaptability. Therefore, this study proposes a joint scheduling method of user association and channel resources based on MADDQN and DMADDPG. This method decouples discrete user scheduling and continuous resource allocation, and achieves multi-agent collaborative optimization under a centralized training and distributed execution framework. The results showed that the total throughput of the research method reached 468.52 Mbps, which was 116.42 Mbps higher than the weighted minimum mean square error of 352.10 Mbps. When the user scale was expanded to 200, the non-compliance rate of the research method was only 16.82%, which was 32.30% lower than the DMADDPG algorithm, and still maintained an average rate of 2.48 Mbps. In summary, the proposed method has good robustness and scalability while improving system performance, and provides an effective solution for large-scale heterogeneous IoT resource scheduling.

Keywords: ***Channel resource allocation; Double deep Q-network; Heterogeneous Internet of Things; Multi-agent reinforcement learning; User association***

1. Introduction

With the advancement of Internet of Things (IoT) technology, heterogeneous IoT targeting large-scale terminal access, differentiated business requirements, and diverse network forms has gradually become an important part of future wireless communication systems [1]. At the same time, it also makes wireless resource scheduling face more complex coupling relationships [2]. User association decision-making and Channel Resource Allocation (CRA) interact and restrict each other. Realizing joint scheduling of user association and channel resources under limited spectrum and power resource constraints has become the key to improving the performance of heterogeneous IoT systems [3-4]. Therefore, in recent years, the academic community has conducted a large amount of research on user scheduling and resource optimization in diverse network scenarios. Van *et al.* [5] proposed a closed-form analysis method and resource optimization framework for uplink traversal throughput based on maximum ratio merging for the IoT multi-access scenario of satellite and ground collaborative services. This framework realized dynamic scheduling of users and allocation of power through alternating optimization or graph neural network, thereby significantly improving the total system throughput under limited resources [5]. Lei *et al.* [6] proposed a collaborative design scheme for joint trajectory optimization, user scheduling and power allocation to address the physical layer security issues of UAV-assisted aerial cognitive IoT systems. This

Yankai Xie, "Heterogeneous IoT User Association and Channel Resource Joint Scheduling Method Based on MADDQN and DMADDPG", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, 1 April 2026, Vol. 10, No. 2, pp. 89-108, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2026.02.005, Available: http://aetic.theiaer.org/archive/v10/v10n2/p5.html.

scheme maximized the average user confidentiality rate through iterative block coordinate descent and successive convex approximation algorithms, effectively suppressing the threat of eavesdroppers with uncertain locations [6]. Li *et al.* [7] proposed a Deep Reinforcement Learning (DRL) algorithm based on policy gradient to address differentiated service quality and energy efficiency requirements in multi-service heterogeneous networks. This algorithm jointly optimized user association and power control in a continuous action space, thereby increasing the transmission rate and optimizing energy efficiency under delay constraints [7].

In addition, Lin *et al.* [8] proposed a distributed DRL method based on recursive neural network state prediction to address the problem of dynamic resource management in edge computing IIoT networks based on Non-Orthogonal Multiple Access (NOMA) and multi-server collaboration. This algorithm realized the joint optimization of task offloading decision-making and sub-channel allocation, thereby effectively improving the task satisfaction rate in partially observable environments [8]. De Souza *et al.* [9] designed a joint optimization scheme for user scheduling and power allocation based on group search to address the Quality of Service (QoS) guarantee problem of ultra-large-scale Multiple-Input Multiple-Output (MIMO) systems in high-density user scenarios. This solution maximized the number of serviceable users that meet QoS requirements under limited power and resource blocks [9]. Cheng *et al.* [10] proposed a multi-agent hybrid DRL algorithm to address the interference suppression problem under restricted backhaul in multi-connection UAV networks. This algorithm realized joint dynamic optimization of UAV-user association, channel allocation, and power control, thereby maximizing the long-term utility in the system in a hybrid discrete-continuous action space [10]. Salim *et al.* [11] proposed a comprehensive review method based on the three major segments to address the increasingly severe network attack threats in satellite communication systems. This method includes a teaching overview of the architecture, a detailed classification of existing network attacks corresponding to the STRIDE threat model, a systematic organization of general and specific defense strategies and technologies for each part, key insights extraction, open challenges, and identification of future research directions, as well as the production of the first comprehensive attack defense situation map and research roadmap covering all three parts, providing reference for satellite communication security research and practice [11]. Jiang *et al.* [12] proposed a comprehensive review method for terahertz (THz) band communication and sensing to address the challenges of supporting 6G requirements for higher capacity, better performance, and enhanced network perception capabilities. This review covers advantages and applications, propagation characteristics and channel modeling, measurement experiments, antennas and transceivers, beamforming, network architecture, communication-sensing synergy, and experimental platforms, grasping existing technologies and summarizing key challenges as well as technical approaches to compensate for high propagation losses [12]. To address the reduced endurance and network performance in UAV-assisted communication due to onboard energy constraints, Pandey *et al.* [13] proposed a comprehensive review and challenge analysis of energy harvesting technologies, receiver architectures, channel modeling, optimization methods, and integration with existing wireless infrastructure. The study systematically synthesized current technologies, key challenges, performance insights, and recommendations for future research directions and practical implementation [13].

In summary, existing research has conducted in-depth exploration of user scheduling and resource optimization issues in different wireless network scenarios, and has achieved certain performance improvements in specific system models. However, there are still several controversies and limitations in existing research. Firstly, there is a clear trade-off in the effectiveness of methodology. Multiple studies have shown that in deterministic environments with complete channel state information and static or quasi-static business models, centralized algorithms based on convex optimization or alternating iterations often converge to theoretical local optimal solutions, and their performance stability and optimality guarantees are usually superior to data-driven DRL methods. This contradiction highlights the performance boundaries of traditional optimization and deep reinforcement learning methods under different environmental assumptions: the former excels in static, precisely modelable scenarios with guaranteed performance and optimality, while the latter specializes in handling dynamism and uncertainty. In the large-scale heterogeneous IoT scenario that this article focuses on, the rapidly changing channel conditions, random arrival of heterogeneous traffic, and device mobility make it extremely

difficult to construct accurate, real-time solvable mathematical models. Therefore, although convex optimization performs better under ideal conditions, the data-driven and online adaptive learning capabilities of DRL make it a more feasible choice for addressing such practical dynamic environments. This provides key justification for adopting deep reinforcement learning methods in dynamic heterogeneous IoT. The deterministic method based on convex optimization is suitable for static or quasi-static environments, requiring accurate system models and complete channel state information, and can provide theoretical optimal solutions under certain conditions. The DRL method is more suitable for dynamic and random environments, and can learn optimization strategies without the need for precise models through data-driven methods, with online adaptive capabilities. In the actual deployment of heterogeneous IoT, due to factors such as user mobility, business randomness, and channel variability, the system often exhibits strong dynamic characteristics and uncertainties, making it extremely difficult to construct accurate mathematical models and solve them in real-time. At this point, although deterministic methods may have theoretical advantages under ideal conditions, the practicality and robustness of DRL methods in practical dynamic environments make them a more suitable choice. Secondly, although there have been attempts to use multi-agent DRL to solve coupled optimization problems, its scalability is often limited by the "curse of dimensionality" or inadequate communication design between agents. For example, in the standard MADDPG framework, centralized Critic needs to process the joint action and state information of all agents, and its input dimension increases exponentially with the number of agents, which directly leads to the bottleneck of unstable training and computational explosion in large-scale scenarios with more than dozens of agents. The centralized critic in the standard MADDPG framework does need to handle the joint state and action information of all agents, which poses a curse of dimensionality. The MADDQN-DMADDPG framework effectively alleviates this problem through the following innovative designs: firstly, decoupling the mixed action space into discrete user association decisions and continuous resource allocation decisions, respectively reducing the decision dimensions that the critic network needs to handle in each sub-problem. Secondly, a hierarchical centralized training distributed execution architecture is adopted, which utilizes global information to learn and coordinate strategies during the training phase, while each agent relies only on local observations for independent decision-making during the execution phase, avoiding the dependence on high-dimensional global information in actual deployment. In addition, by designing efficient state representation and attention mechanisms, the information dimensions that the critic network needs to process are further compressed. Similarly, the value decomposition network structure of methods such as Q-value Mixing Network (QMIX) may have limited function approximation ability when dealing with highly heterogeneous and tightly coupled mixed action spaces, making it difficult to coordinate complex interference competition relationships among large-scale users. Therefore, the precise gap in this field lies in the lack of a collaborative decision-making architecture that can efficiently handle hundreds of intelligent agents (i.e. the number of concurrent access users or small base stations in large-scale heterogeneous IoT, with a typical user scale of 100-500), a mixed discrete continuous action space (user association is a discrete decision, power and channel allocation is a continuous decision), and strong signal coupling interference (high coupling of same frequency interference between adjacent users and NOMA interlayer interference). For example, reference [7] adopts a single agent strategy gradient for joint optimization in a continuous action space, which simplifies the design but does not consider discrete user association decisions, and the scalability of its centralized architecture is questionable. Reference [10] introduced a mixed action space, but its model mainly targets small-scale clusters of unmanned aerial vehicle networks and has not been validated for generalization ability in dense ground heterogeneous IoT. Recurrent neural networks to handle partial observability in reference [8] is a highlight, but the balance between distributed execution efficiency and global performance has not been thoroughly explored. The lack of this critical perspective has resulted in existing research failing to clearly define the effective boundaries and core flaws of its solutions when facing the comprehensive challenge of large-scale, strongly coupled, and dynamically heterogeneous IoT.

To this end, this paper constructs a joint optimization method based on the collaborative combination of centralized single agent and multi-agent to deal with the joint scheduling issue of user association and channel resources in heterogeneous IoT. This method decouples discrete user scheduling and continuous

CRA, and introduces a Multi-Agent Double Deep Q-Network and Double Multi-Agent Deep Deterministic Policy Gradient (MADDQN-DMADDPG) collaborative optimization mechanism for large-scale systems under a centralized training and Distributed Execution (DE) framework. This mechanism enables joint decision-making by multiple agents in a hybrid discrete-continuous action space, aiming to ensure system scalability and DE capabilities while improving overall scheduling performance. The main innovation lies in constructing a unified DRL framework to jointly optimize user association and channel resources, and effectively alleviating non-stationary problems in multi-agent environments through centralized training and DE mechanisms. The core improvement of the research work on the standard MADDPG and DQN frameworks lies in designing a multi-objective reward function that integrates real-time throughput, fairness penalty, and switching overhead, and introduces a channel interference attention module in the front end of the Actor network to address coupling issues. These specific designs are key to addressing resource competition and interference coordination issues in heterogeneous IoT. This paper provides an efficient and scalable solution for resource scheduling in complex heterogeneous IoT scenarios.

2. Methodologies

2.1. Joint Resource Scheduling Method for Centralized Single-Agent DRL

A downlink heterogeneous IoT network consisting of one Macro Base Station (MBS) and K Small Base Stations (SBSs) is considered, collectively serving N User Equipment (UEs). The total system bandwidth is B, divided into M orthogonal sub-channels. The user association variable is defined as $x_{n,k} \in \{0,1\}$, indicating whether user n is associated with base station k. The CRA variable is $p_{n,k}^{m}$, representing the power allocated by base station k to user n on subchannel m. Under the total power constraint $\sum_{k,n,m} p_{n,k,m} \le P_{\max}$ and the user unique association constraint $\sum_{k} x_{n,k} = 1, \forall n$, the long-term discounted total system throughput is maximized through joint optimization of the user association matrix.

In heterogeneous IoT, user-related decision-making and CRA are highly coupled, and traditional optimization methods are difficult to adapt to the dynamic network environment. Given this, this study first constructs a centralized single-agent DRL joint resource scheduling method based on Deep Q-Network and Double Deep Deterministic Policy Gradient (DQN-DDDPG) as a baseline solution for subsequent multi-agent collaborative optimization methods. This method adopts a hierarchical structure of sequential execution: the DQN module first generates discrete user association decisions, which are then input as part of the system state to the DDDPG module for fine-grained continuous resource allocation under given association relationships. Two modules undergo end-to-end joint training through a jointly designed final reward. The scheduling method based on DQN-DDDPG is displayed in Fig.1.

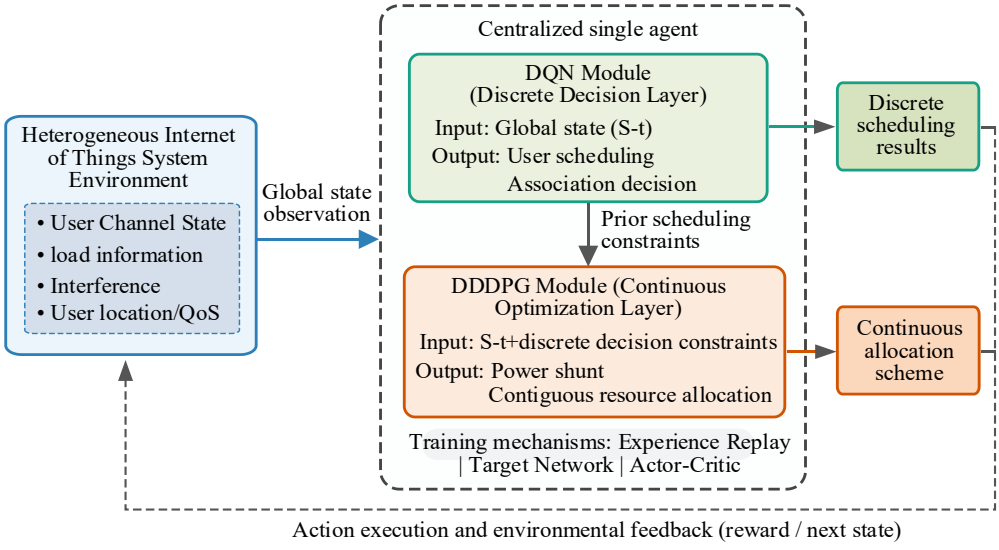


Figure 1. DQN-DDDPG method

In Fig.1, the network control node is modeled as a unique agent, which autonomously learns the joint decision-making strategy of user scheduling and CRA by sensing global network status information. To effectively handle the mixed discrete-continuous action space, this study adopts the hierarchical optimization structure of DQN-DDDPG. DQN is responsible for discrete user scheduling and association decision-making to determine the set of users participating in communication. DDDPG further optimizes CRA and power offloading strategies in continuous action space.

2.1.1. User Scheduling and Discrete Resource Decision Modeling Based On DQN

In the centralized single-agent DRL joint resource scheduling framework, user scheduling and discrete resource decision-making constitute a key link in the joint optimization process. Therefore, this study builds a centralized decision-making model based on DQN for user scheduling and discrete resource decision-making problems in heterogeneous IoT environments, as shown in Fig.2.

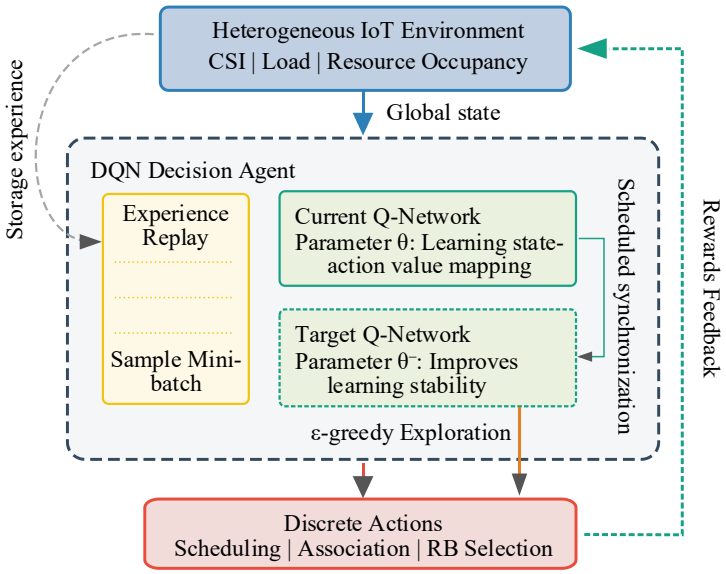


Figure 2. Centralized decision-making model based on DQN

In Fig.2, the centralized agent perceives and models the global network status through DQN, learns the status-action mapping relationship, and realizes adaptive optimization of user scheduling and discrete resource allocation. The state space describes the operating characteristics of the system in the current time slot, including user channel quality, network load and resource occupancy, etc. The action space consists of a set of discrete decisions describing choices such as user scheduling, user association, and discrete resource block allocation. State Space S_d: The global state observed by the agent at time slot t, include: The channel gain from all users to each base station, the current load of each base station, the status of the user service request queue, and the user association relationship from the previous time slot. Action Space A_d: The agent's discrete action is a multidimensional vector, defining the user-base station association decisions and rough sub-channel block allocations for users in the current time slot. The reward function is designed as $r_t = \alpha \cdot T_t - \beta \cdot F_t - \eta \cdot H_t \cdot$ T_t represents the system's current total throughput, F_t is the penalty term based on the Jain fairness index, and H_t is the user association switching overhead. α, β, and η are weighting coefficients. To gradually approach the optimal scheduling strategy in a complex discrete decision space, DQN adopts the $\grave{o}-greedy$ strategy to trade-off between exploration and utilization, and introduces experience replay and target Q-network mechanisms during the training process to reduce sample correlation and improve learning stability. The strategy $\grave{o}-greedy$ is shown in formula (1) [14]:

$$a_t = \begin{cases} \text{random action from A}, & \text{with probability } \grave{o} \\ \arg\max_{a \in A} Q(s_t, a; \theta), & \text{with probability } 1-\grave{o} \end{cases} \quad (1)$$

In formula (1), a_t is the discrete scheduling action finally selected by the agent. A is a discrete action space (user association and resource block set). ò is the exploration probability. Equation (1) ensures that the agent has a probability of ò to try random scheduling and a probability of $1-$ò to select the current optimal strategy to ensure that the model will not prematurely converge to the local optimal solution in a complex discrete space. The exploration probability adopts a linear decay strategy, gradually decaying from an initial value of 0.9 to 0.05, with the decay process spanning 80% of the total training steps.

Secondly, the target Q value is shown in formula (2) [15-16]:

$$y_t = r_t + \gamma \max_{a' \in A} \hat{Q}(s_{t+1}, a'; \theta^-) \quad (2)$$

In formula (2), y_t is the target Q value (the learned target label) of the t-th time slot. r_t is the immediate reward of environmental feedback (such as system throughput or energy efficiency). $\gamma \in [0,1]$ is the discount factor. s_{t+1} is the next state after executing the action. a' is the optional discrete action in the next state (ie, the next user scheduling decision). $\hat{Q}(\cdot)$ and θ^- are the output function and parameter of the target network. Formula (2) employs the Bellman equation and uses an independent set of parameters θ^- to predict the maximum value in the future. This avoids shocks caused by simultaneous updates and predictions on the same network and improves the stability of discrete resource scheduling strategies. The termination condition for model training is set as follows: within the sliding window (such as the last 100 training rounds), the average cumulative reward increase is less than the threshold (0.01), or the total training steps reach the preset upper limit.

In summary, centralized agents can effectively complete discrete scheduling decisions in a dynamic heterogeneous IoT environment, and lay a reliable decision-making basis for subsequent CRA and power optimization.

2.1.2. Joint Optimization of CRA And Power Offloading Based on DDDPG

On the basis of completing user scheduling and discrete resource decision-making based on DQN, the system still needs to further refine the continuous channel resources and power split parameters to fully unleash the performance potential of the heterogeneous IoT system. However, traditional reinforcement learning methods based on discrete action spaces are difficult to directly apply. Therefore, this study constructs a continuous resource joint optimization model based on DDDPG to achieve collaborative optimize CRA and power offloading under a centralized single-agent framework, as shown in Fig.3.

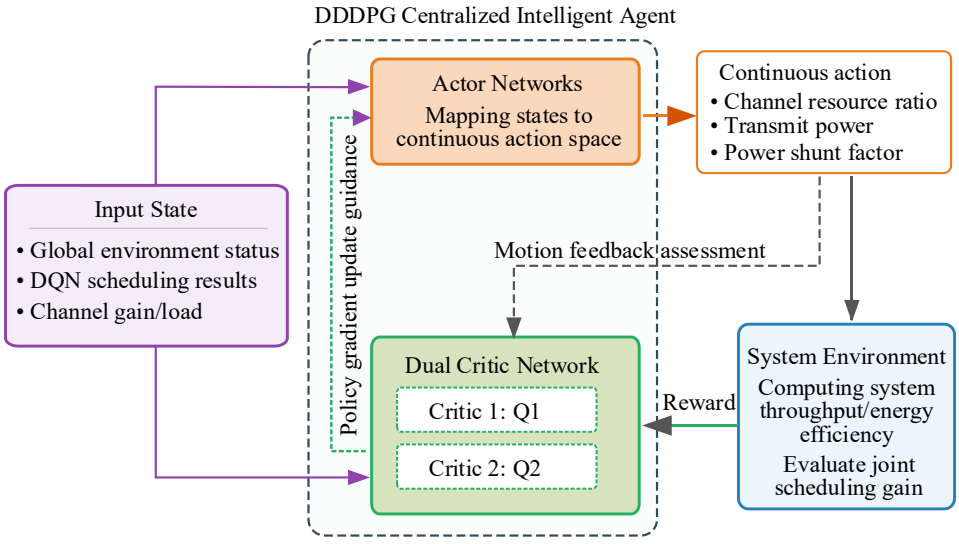


Figure 3. Continuous resource joint optimization model based on DDDPG

In Fig.3, the centralized agent uses the Actor-Critic architecture to model the continuous action space. The Actor outputs CRA and power distribution strategies based on the current system status. The state

space S_c of the formalized DDDPG component: On the basis of S_d, the current user association X_t generated by DQN decisions is additionally incorporated as a condition. The action space A_c of the DDDPG component: The continuous action of the agent is a vector, specifically the precise power allocation ratio for each associated link on the assigned sub-channel. The reward function of the DDDPG component inherits and refines r_d, incorporating considerations for energy efficiency and interference: $r_t^c = r_t + \omega \cdot (\sum_n R_n / P_{total}) - \mu \cdot I_{\text{inter}}$, where P_{total} represents the total power consumption, and I_{inter} estimates the inter-cell interference level. ω represents the weight coefficient of the energy efficiency term. μ represents the weight coefficient of the interference penalty term. The Critic evaluates the value of state-action pairs to guide policy updates, as shown in formula (3) [17]:

$$\nabla_{\theta^\mu} J \approx \mathrm{E}_{s_t \sim \mathrm{D}} \left[\nabla_a Q_1(s, a \mid \theta^{q_1}) \big|_{a=\mu(s|\theta^\mu)} \cdot \nabla_{\theta^\mu} \mu(s \mid \theta^\mu) \right] \tag{3}$$

In formula (3), $\nabla_{\theta^\mu} J$ is the policy gradient, which is utilized to update the Actor parameter θ^μ. Q_1 is the first to evaluate the Critic network. $\mu(s \mid \theta^\mu)$ is the current action generated by the Actor network. Formula (4) calculates the Critic's evaluation gradient of the current action (that is, "in which direction the action changes, the Q value will be greater"), and then transmits it to the parameters of the Actor network. In this way, the Actor can continuously "fine-tune" toward a higher performance resource allocation scheme within the continuous action space.

The state space further integrates user scheduling results on the basis of inheriting the system state description in the discrete scheduling stage, thereby ensuring that the continuous optimization process is conducted under scheduling constraints. The action space consists of continuous variables such as CRA ratio, transmit power and power shunt factor, and is used to finely characterize the system resource allocation plan. To improve training stability and convergence performance, DDDPG introduces a double critic structure and combines experience replay and target network mechanisms to alleviate the problems of Q-value overestimation and policy oscillation, as shown in formula (4) [18]:

$$y_t = r_t + \gamma \min_{i=1,2} \hat{Q}_i \left(s_{t+1}, \mu_{targ}(s_{t+1} \mid \theta^{\mu^-}) \mid \theta^{q_i^-} \right) \tag{4}$$

In formula (4), y_t means the target Q-value (used to update the Critic network). r_t is the immediate reward obtained after performing consecutive actions (power splitting, channel ratio). μ_{targ} is the target Actor. $\hat{Q}_i$ is the i-th target Critic ($i = 1, 2$). θ^{μ^-} and $\theta^{q_i^-}$ are the parameters of the target Actor and Critic. Formula (4) takes the smaller of the two target Critic evaluation values as the learning target, which effectively prevents the system from biasing the update direction caused by overly optimistic estimates of the benefits of power or resource allocation schemes.

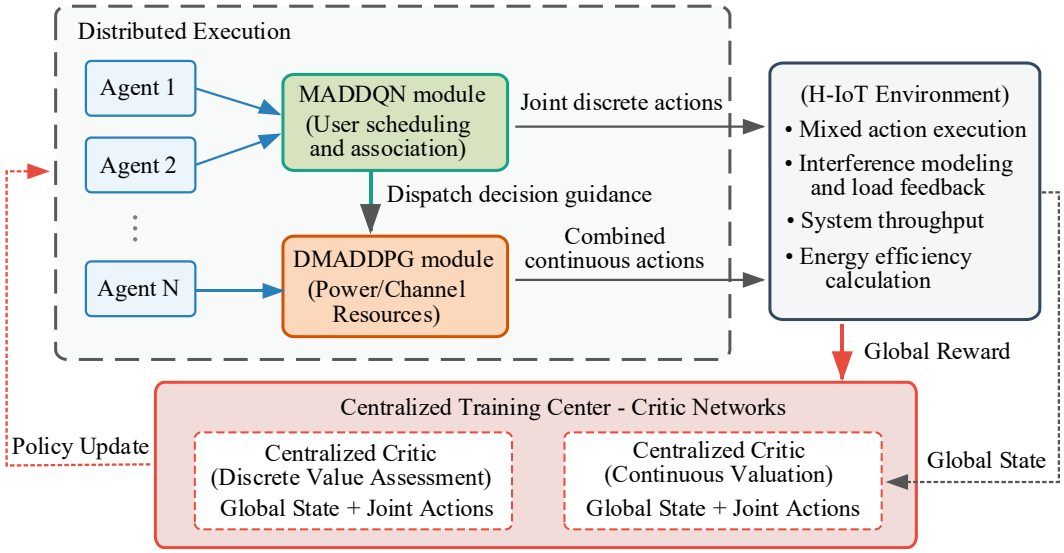
In summary, the DDDPG module can jointly optimize continuous resources and power distribution in a dynamic heterogeneous IoT environment, thereby further improving the overall effectiveness of the system.

2.2. Multi-Agent Collaborative DRL Joint Optimization Method for Large-Scale Systems

Although the centralized single-agent DRL method can effectively achieve joint optimization of user scheduling and CRA, it still faces problems such as high computational complexity and limited scalability in large-scale heterogeneous IoT scenarios. This study further introduces the multi-agent collaborative DRL framework and proposes the MADDQN-DMADDPG method, which improves the overall performance of the system while ensuring DE capabilities, as shown in Fig.4.

In Fig.4, the users or access nodes of the system are modeled as multiple collaborative agents, and the joint resource scheduling issue is characterized as a multi-agent Markov game under the centralized training and DE framework. For the hybrid discrete-continuous action space, this study adopts a hierarchical collaborative optimization structure. MADDQN is responsible for multi-agent user scheduling and discrete resource decision-making. DMADDPG co-optimizes continuous CRA and power splitting. In the training phase, the centralized critic uses joint status and joint actions to guide each agent's strategy update, while in the execution phase, each agent only depends on local observations to

make independent decisions. Training phase (centralized): The central critic collects local observations and actions of all agents to form joint information, which is used to calculate the global Q-value and update the Actor network of each agent. Intelligent agents collaborate through shared experience replay pools and central parameter servers for collaborative learning. Execution phase (distributed): Each agent generates actions using its independent Actor network based solely on its own local observations, without the need for real-time communication or information exchange with other agents.



MADDQN-DMADDPG Architecture for Large-scale H-IoT Systems

Figure 4. Optimization method based on MADDQN-DMADDPG

2.2.1. Multi-Agent User Scheduling and Discrete Collaborative Decision-Making Based on MADDQN

In the multi-agent collaborative DRL joint optimization framework, user scheduling and discrete resource decision-making are the key decision-making layers to achieve multi-agent collaboration. This study aims at multi-user concurrent access and highly coupled scheduling decisions in heterogeneous IoT, and proposes a user scheduling and discrete collaborative decision-making method based on MADDQN is proposed, as shown in Fig.5.

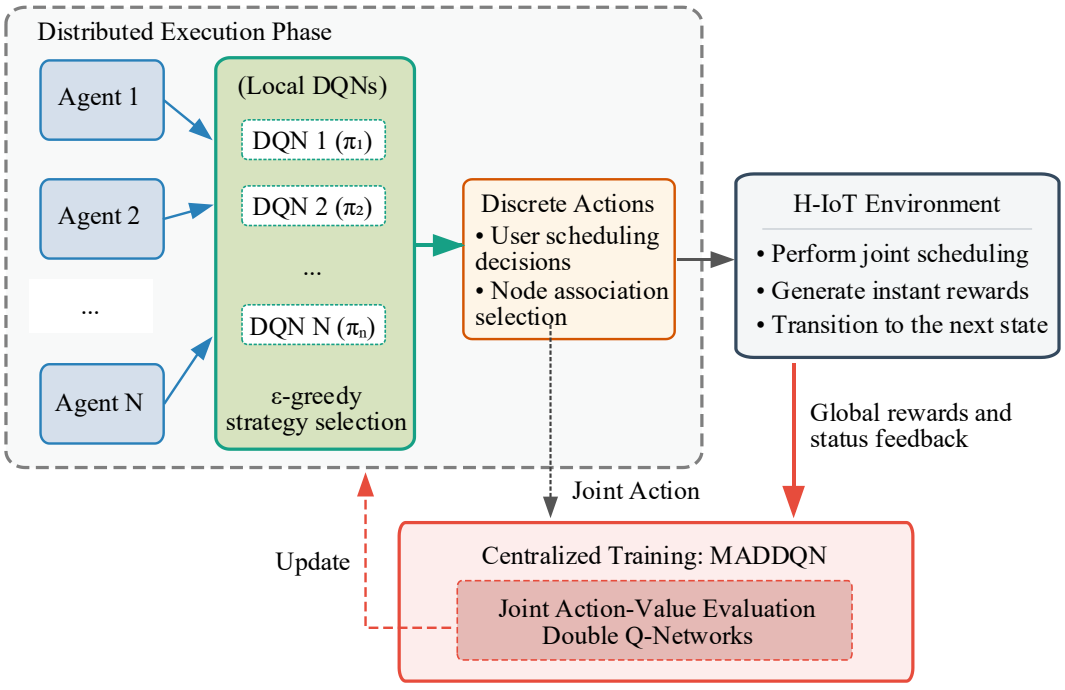


Figure 5. Decision-making method based on MADDQN

In Fig.5, each agent independently generates discrete scheduling and associated actions based on local observations, and achieves collaborative behavior through shared global rewards. In the training

phase, MADDQN uses joint states and joint actions to construct a joint action value function, which effectively alleviates non-stationary problems in multi-agent environments, as shown in formula (5) [19]:

$$Q_{tot}(s,a_1,a_2,\ldots,a_n;\theta)=f(Q_1(o_1,a_1;\theta_1),\ldots,Q_n(o_n,a_n;\theta_n)) \quad (5)$$

In formula (5), Q_{tot} is the joint action value function, which represents the global performance evaluation. s is the global state of the system (containing information about all agents). o_i and a_i are the local observations and discrete scheduling actions of the i-th agent. $f(\cdot)$ is an aggregation function used to transform local values into global values. This function takes the concatenated vector of local observations and actions of all agents as input, and outputs a scalar as an estimate of the joint action value. Based on formula (5), the system can quantify "the contribution of each agent's local scheduling decision to global network performance (such as total throughput)", thereby achieving collaboration.

In the execution phase, each agent only relies on local information to complete scheduling decisions, thereby significantly reducing communication and computing overhead, as shown in formula (6) [20]:

$$a_i^*=\arg\max_{a_i\in \mathrm{A}_i} Q_i(o_i,a_i;\theta_i) \quad (6)$$

In formula (6), a_i^* is the discrete scheduling plan finally executed by the i-th agent. A_i is the optional scheduling and associated action space of the agent. o_i is the agent's local observation (not dependent on the global state s). In formula (6), although joint information is used during training, in the execution stage, each agent only needs to reason about the local observation o_i through its own local network Q_i.

To improve the stability and convergence performance of discrete collaborative decision-making, MADDQN introduces a double-Q network structure to reduce the risk of Q value overestimation, as shown in formula (7) [21]:

$$y=r+\gamma\hat{Q}_{tot}(s',a_1',\ldots,a_n';\theta^-) \quad (7)$$

In formula (7), the action selection at the next moment follows $a_i'=\arg\max_{a_i} Q_i(o_i',a_i;\theta_i)$. y is the learning target (label) of multi-agent collaboration. r is the global shared reward. $\hat{Q}_{tot}$ is the joint target network. θ and θ^- are the evaluation network parameters and target network parameters. To alleviate overestimation, formula (8) adopts the "decoupling" idea, using the current evaluation network Q_i to select the optimal scheduling action a_i', and using the $\hat{Q}_{tot}$ to calculate the action value. This ensures that the update of scheduling decisions is more robust under the frequent fluctuations of large-scale heterogeneous IoT.

In summary, multi-agents can achieve efficient and stable discrete collaborative scheduling in a dynamic heterogeneous IoT environment, laying a reliable scheduling foundation for subsequent continuous CRA and power distribution optimization.

2.2.2. CRA And Multi-Agent Collaborative Optimization Based On DMADDPG

Since the optimization variables such as CRA ratio, transmit power and power shunt factor have continuous value characteristics, and the decision-making between multiple agents is coupled with each other, the traditional single-agent continuous control method is difficult to effectively apply. To this end, this paper proposes a CRA and collaborative optimization method based on DMADDPG, as shown in Fig.6.

In Fig.6, unlike centralized single-agent continuous control, DMADDPG configures an independent Actor network for each agent to autonomously output CRA and power offload decisions based on local observations. The multi-agent distributed policy gradient is shown in formula (8) [22]:

$$\nabla_{\theta^{\mu_i}} J(\theta^{\mu_i})=\mathrm{E}_{s,a\sim \mathrm{D}}\left[\nabla_{a_i} Q_i^{\mu}(s,a_1,\ldots,a_n)\cdot\nabla_{\theta^{\mu_i}}\mu_i(o_i\mid\theta^{\mu_i})\right] \quad (8)$$

In formula (8), o_i is the local observation information of the i-th agent. $\mu_i(o_i\mid\theta^{\mu_i})$ is the distributed Actor network of the i-th agent, which outputs a CRA strategy. $\nabla_{a_i} Q_i^{\mu}(\cdot)$ is the evaluation gradient provided by the centralized critic. Formula (8) expresses that although the joint action gradient

($\nabla_{a_i} Q_i$) provided by the global critic is needed when updating parameters, the final optimized Actor network (μ_i) only relies on the local observation o_i.

A centralized Critic network is introduced to conduct a unified value evaluation of the joint status and joint actions of multiple agents, thereby depicting the mutual influence and collaborative relationship between agents during the training phase. This centralized training and DE mechanism effectively alleviates the non-stationary problem in the continuous decision-making environment of multi-agent, allowing each agent to achieve collaborative behavior without global information during the execution phase. The centralized joint action value function is shown in formula (9) [23]:

$$Q_i^{\mu}(s, a_1, a_2, \ldots, a_n; \theta^{q_i}) \tag{9}$$

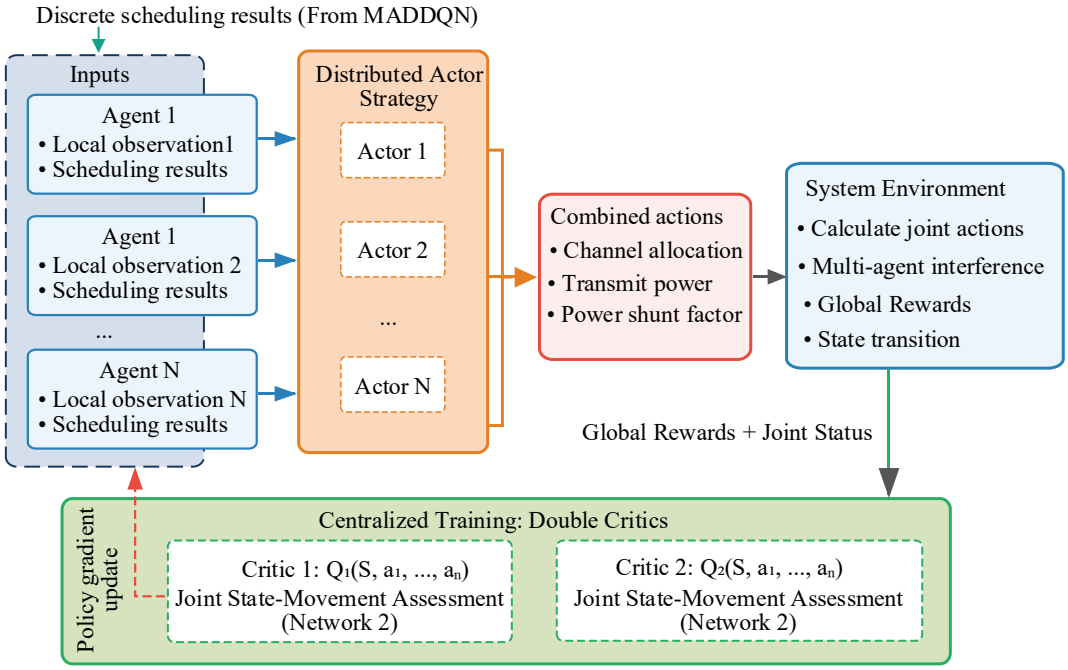


Figure 6. DMADDPG-based method

In formula (9), s is the global network status, including the joint channel status, interference environment and other information of all access nodes and users. $a_1, a_2, \ldots, a_n$ is the joint continuous action vector of n collaborative agents (such as the power split ratio, channel allocation ratio, etc. of each agent). θ^{q_i} is the parameter of the Critic to which the i-th agent belongs. Unlike a single agent $Q(s,a)$ that only observes its own actions, formula (9) takes the actions of all agents as input. In this way, Critic can observe the global interference and gain caused by each agent's resource adjustment, thereby accurately depicting the collaboration and coupling relationships between agents during the training phase.

To cope with the highly coupled and dynamic changes in channel state and user interference in heterogeneous IoT, the research method integrates a lightweight channel interference attention module at the front end of each actor network. The core function of this module is to adaptively extract the interference relationship information most relevant to the current agent decision from the original joint observations. For intelligent agents, their local observations contain Channel State Information (CSI) from all users to various access points within their service range. The attention module first maps these CSI information and the user's historical rate information into a series of feature vectors. The characteristics of the user currently served by the intelligent agent itself are used as the query vector. Subsequently, the scaled dot product attention mechanism is used to calculate the correlation weights between the query vector and all key vectors. A weighted environment representation vector focusing on key interference sources and channel changes is obtained. This vector is concatenated with other local state information of the agent, such as queue length, and used as input for the subsequent fully connected layers of the Actor network to generate continuous actions such as CRA and power splitting. Through this design, intelligent

agents can learn to "focus" on the dynamic factors that have the greatest impact on their decisions, significantly enhancing their ability to optimize collaborative resources in complex coupled environments.

2.2.3. Extension of the Model to NOMA Heterogeneous Networks

The MADDQN-DMADDPG framework has generality and can be extended to heterogeneous IoT scenarios based on Non-Orthogonal Multiple Access (NOMA). NOMA can improve spectral efficiency and access capacity by superimposing multiple user signals on the same time-frequency resource block and utilizing continuous interference cancellation (SIC) technology, but it also introduces more complex power domain coupling and user to user interference relationships.

In terms of expanding the state space, the global state not only includes the original channel state information (CSI), load, etc., but also needs to integrate channel gain ranking information and SIC decoding order of users within the NOMA user cluster. For each sub channel, the agent needs to perceive the relative channel conditions of users in the user cluster scheduled on it to evaluate inter layer interference.

In terms of adjusting and coupling the action space, the action definition needs to be expanded from "user subchannel association" to "user sub-channel power layer association". The intelligent agent not only needs to decide which sub channel to allocate to the user, but also needs to determine its power layer (such as high power layer or low power layer) in the corresponding channel NOMA user cluster. This is essentially a joint discrete decision of user grouping (pairing) and resource block allocation. Continuous action requires precise output of power allocation coefficients for each user within the same NOMA user cluster, based on the original power allocation. Intelligent agents must collaborate to optimize the power between clusters (different sub-channels) and within clusters (different users on the same sub-channel) to achieve a balance between improving spectral efficiency and ensuring successful decoding of SIC.

During the centralized training phase, the central Critic network is able to observe the joint interference situation inside and outside all user clusters, thereby learning to coordinate various agents (such as access points) for cross cluster interference management and user pairing within clusters.

To guide intelligent agents to learn scheduling strategies that balance system efficiency, fairness, and stability in dynamic and complex heterogeneous IoT environments, this study designs a comprehensive reward function that integrates multi-dimensional performance indicators. This function calculates t at each training time slot as an immediate evaluation of the joint actions of the agents. The reward r_t is composed of three weighted parts: the total throughput reward $R_{sum}(t)$, the fairness penalty $P_{\text{fair}}(t)$, and the switching cost penalty $P_{\text{handover}}(t)$. Its mathematical expression is shown in equation (10):

$$r_t = \omega_1 \cdot R_{\text{sum}}(t) + \omega_2 \cdot P_{\text{fair}}(t) + \omega_3 \cdot P_{\text{handover}}(t) \tag{10}$$

In equation (10), ω_1, ω_2, and ω_3 are the weight coefficients of each component used to balance the importance of different optimization objectives. Through the targeted design of states, actions, and rewards mentioned above, the proposed MADDQN-DMADDPG framework can learn joint strategies for user grouping, subchannel allocation, and precise power control in NOMA heterogeneous networks without changing its core architecture of "discrete continuous decoupling" and "centralized training distributed execution", thus addressing higher dimensional and stronger coupling resource scheduling challenges.

In summary, multi-agent can achieve consistency and coordination of resource allocation strategies in continuous action space, improving the scalability and robustness of the method in large-scale heterogeneous IoT scenarios.

3. Results

3.1. Performance Improvement and Comprehensive Comparison Verification

To verify the superiority of the MADDQN-DMADDPG method, this study carries out simulation verification. The experimental parameters are shown in Table 1.

In Table 1, this study uses the open source wireless channel data set generated by the 3GPP TR 38.901 standard channel model and the synthetic heterogeneous IoT user distribution data set based on the Poisson point process and random walk model to conduct simulation experiments. User mobility adopts a

hybrid model that goes beyond simple random walks and includes both group movement and stationary users, to simulate the characteristics of fixed devices (such as sensors) and mobile devices (such as in vehicle terminals) in real heterogeneous IoT scenarios. The traffic model considers both continuous bitstreams (such as video surveillance) and burst data packets (such as environment aware) to cover multiple business service quality requirements. The interference calculation is based on the actual scheduling results accumulated by each sub-channel, and considers the serial interference cancellation process in non-orthogonal multiple access. The hyperparameter optimization adopts grid search and early stopping method, pre-training for 2,000 rounds with a scale of 100 users to determine the optimal combination.

Table 1. Environment and key parameters

Environment			**Configuration**	
Hardware	CPU	Intel i7 or AMD Ryzen 7 (≥8 cores)	Topology	1 macro BS + SSS small BSs; users randomly deployed
	GPU	NVIDIA RTX 3060 (12GB) 3090	Number of small base stations	3–15
	RAM	32GB	Number of users	20–200
	Storage	1TB SSD	Sub-channel	10–50
	Operating system	Ubuntu 22.04 LTS	Bandwidth BB	10–20 MHz (adjustable)
Software	Language	Python 3.10+	Noise PSD	−174 dBm/Hz
	DL Framework	PyTorch 2.x	Discount factor	0.90–0.99
	RL Interface	Gymnasium	Learning rate	Actor: 1e−41e{-4}1e−4; Critic: 1e−31e{-3}1e−3 (tunable)
	Numerical	SciPy	Replay Buffer	1e5 – 1e61e61e6
	Plotting	Matplotlib	Batch size	128–512
	Wireless Simulation	Custom PHY/link simulation in Python	Target network soft update	0.005–0.01

To comprehensively evaluate the performance of the proposed method, the following four representative algorithms are selected as comparison baselines. Weighted minimum mean square error combined with heuristic user association: Firstly, a heuristic rule based on reference signal reception quality is used for user base station association. That is, each user selects the base station with the highest received signal strength for access. When multiple users are associated with the same base station, they are sorted in descending order according to channel quality indicators, and the top K users with channel quality are prioritized for scheduling. Subsequently, under fixed correlation, the weighted minimum mean square error algorithm is used to iteratively optimize power allocation to maximize the system and rate.

QMIX: Based on the standard QMIX framework, the global Q-value is decomposed into monotonic combinations of local Q-values for each agent. Each intelligent agent performs discrete user scheduling and resource block allocation. This baseline uses default hyperparameters: learning rate 5e-4, discount factor 0.95, experience replay buffer size 1e5, batch size 128, and target network update interval of 200 rounds. The optimization objective is to maximize the global cumulative reward, and the reward function is consistent with the research method (including throughput and fairness penalties).

MADDPG: Adopting the original MADDPG framework, each agent outputs continuous actions (power allocation and channel ratio), and user associations are discretized through greedy rules. Hyperparameter settings: Actor learning rate 1e-4, Critic learning rate 1e-3, discount factor 0.95, experience replay buffer 1e6, batch size 256, and soft update rate τ=0.01. The exploration noise is OU process.

Centralized DQN-DDDPG: DQN part learning rate 1e-4, DDDPG part Actor learning rate 1e-4, Critic learning rate 3e-4, and other hyperparameters are consistent with the research method.

The experiment constructs various comparison scenarios under different user scales, channel resource constraints and network dynamic conditions, and evaluates the performance of the MADDQN-DMADDPG method and various mainstream comparison algorithms. All experiments are repeated no less than 3 times under the same initial conditions. Experimental results are expressed as

mean and standard deviation. The performance differences between different methods are analyzed for significance through paired samples t test, and the significance level is set at $p<0.05$.

First, a comprehensive performance verification of the research method is conducted. DQN-DDDPG, MADDPG, Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning (QMIX), and Weighted Minimum Mean Square Error with Heuristic User Association (WMMSE + Heuristic Association) are selected for comparisons, as shown in Fig.7.

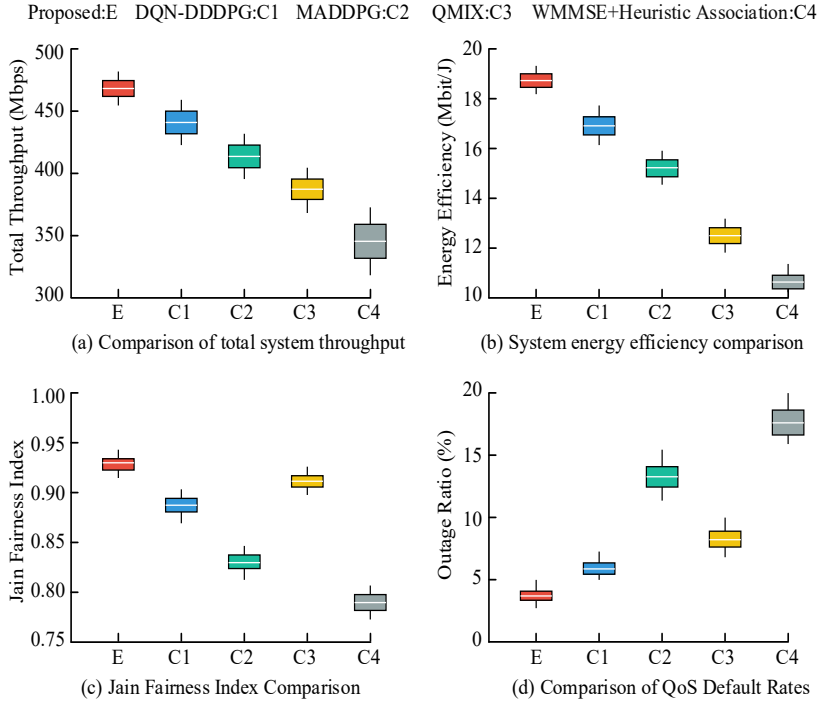


Figure 7. Comprehensive performance comparison analysis

In Fig.7, the total throughput of MADDQN-DMADDPG reaches 468.52 Mbps, which is 116.42Mbps higher than the 352.10 Mbps of WMMSE + Heuristic Association. In terms of energy efficiency performance, the research method achieves a score of 18.76 Mbit/J far exceeding QMIX's 12.56 Mbit/J. In terms of service reliability, the research method controls the QoS non-compliance rate at 3.85%, which is lower than that of MADDPG's 13.45% and WMMSE + Heuristic Association's 19.50%. In addition, the Jain fairness index of the research method reaches 0.93, which is better than the 0.89 of centralized DQN-DDDPG. The above data verify that the proposed method has higher system utility and scalability in large-scale heterogeneous scenarios through collaborative optimization of discrete and continuous actions. A series of ablation experiments were conducted to thoroughly analyze the effectiveness of the hierarchical architecture and double-network design in the proposed method. The study constructed three variants for comparison: Method 1 is the proposed complete method (MADDQN-DMADDPG); Method 2 is to ablate the Double-Q network (w/o Double-Q) by replacing MADDQN with standard multi-agent DQN at the discrete decision layer, using only a single Q-network; Method three is to ablate double critic (w/o Double Critic): in the continuous optimization layer, replace DMADDPG with standard MADDPG and only use a single critic network. In addition, the study used standard MADDPG as the baseline control. The experimental results are shown in Table 2.

Table 2. Analysis of ablation experiments

Performance metrics	MADDQN-D MADDPG	Ablation of Double-Q Network (w/o Double-Q)	Ablation of Double Critic (w/o Double Critic)	MADDPG
Total throughput (Mbps)	468.52±10.24	432.18±12.56*	445.73±11.78*	401.35±15.62**
Energy Efficiency (Mbit/J)	18.76±0.52	16.45±0.68*	17.22±0.61*	14.91±0.87**
QoS non-compliance rate (%)	3.85±0.41	5.92±0.58*	4.88±0.50*	9.14±0.92**
Final average return	1865±32	1720±48*	1792±41*	1622±54**
Return variance (× 10^2)	11.5±1.3	19.8±2.5*	15.2±1.9*	45.3±5.1**
Number of rounds required for convergence	1245±49	1580±71*	1420±65*	2380±112**

Note: Compared with MADDQN-DMADDPG, * represents p<0.05 and * * represents p<0.01.

The ablation experiment results in Table 2 showed that the total throughput of the system decreased from 468.52 Mbps to 435.18 Mbps, a decrease of 7.1%. Meanwhile, the service quality breach rate increased from 3.85% to 5.42%. This indicates that the module significantly optimizes resource allocation efficiency in strongly coupled environments by dynamically "focusing" the agent on the interfering users and channel states that have the greatest impact on its decisions. Its absence will lead to a decrease in the perception ability of intelligent agents in complex interference environments, resulting in sub-optimal scheduling decisions. When only using a single throughput reward, the system seriously sacrifices other performance in pursuit of capacity: The service quality breach rate skyrockets to 8.20%, an increase of over 113%. The Jain Fairness Index deteriorated from 0.93 to 0.84. More significantly, the frequency of user association switching surged from 0.12 to 0.31, indicating an increase in network volatility. This proves that the designed reward function that integrates fairness penalty and switching overhead penalty effectively guides the agent to learn a robust strategy that balances improving system throughput, maintaining rate fairness among users, and avoiding unnecessary switching. After removing the double-Q network, not only did the total throughput decrease, but the variance of the gains during training significantly increased (from 1150 to 1980). This indicates that in user scheduling and associated discrete action learning, the double-Q network effectively alleviates the overestimation of Q-values, making policy updates more stable and reliable, and avoiding decision oscillations caused by overly optimistic value estimates. In summary, the attention module mainly contributes to performance improvement by enhancing environmental awareness and directly optimizing spectral efficiency. The multi-objective reward function mainly contributes to behavior guidance, ensuring that the strategy achieves a balance between efficiency, fairness, and stability among multiple objectives. The double-Q network contributes to the stability of the learning process, providing a reliable foundation for the aforementioned performance improvement and optimization. These four components each have their own focus and complement each other, together forming the excellent performance of the MADDQN-DMADDPG method. Next, the experiment is conducted to verify the scalability performance that changes with the user scale, and DQN-DDDPG, MADDPG, and WMMSE + Heuristic Association are selected as controls, as shown in Fig.8.

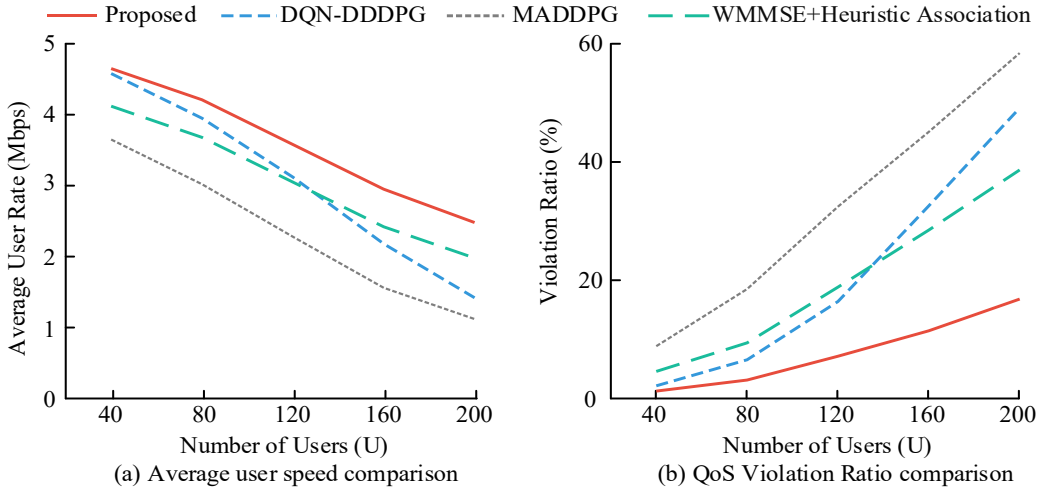
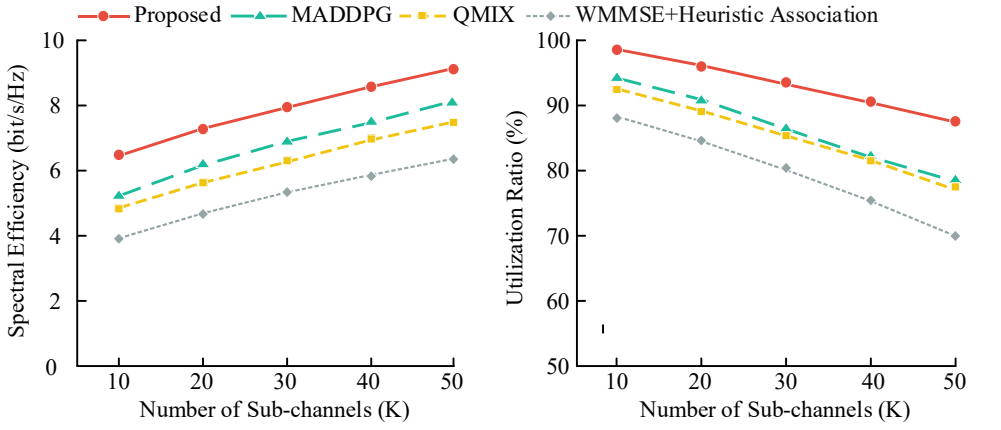


Figure 8. Scalable performance analysis as user size changes

In Fig.8(a), when the user scale increases to 200, the research method can still maintain a rate level of 2.48 Mbps, which is 1.06 Mbps higher than DQN-DDDPG and 1.36 Mbps higher than WMMSE + Heuristic Association. In Fig.8(b), when the user scale reaches 200, the non-compliance rate of the research method is only 16.82%, which is 32.30% lower than DQN-DDDPG and 41.60% lower than WMMSE + Heuristic Association. In summary, the research method can effectively alleviate the serious performance degradation caused by resource competition when the user scale expands, and shows good system robustness in large-scale access scenarios.

Next, the sensitivity verification of resource tension (number of sub-channels/bandwidth) is performed and compared with QMIX, MADDPG, and WMMSE + Heuristic Association, as shown in Fig.9.



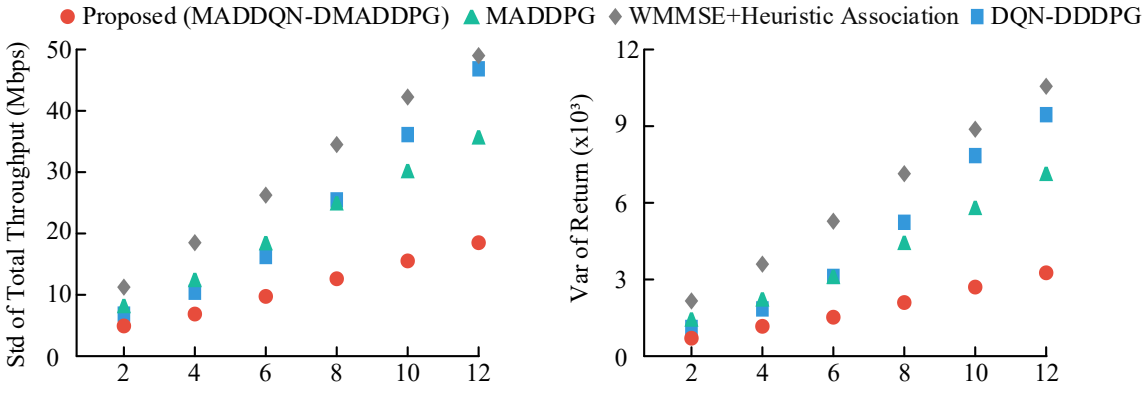
(a) Spectral efficiency comparison (b) Comparison of unit resource block utilization

Figure 9. Resource tension sensitivity analysis

In Fig.9(a), when the number of sub-channels is 10 in an extremely tight environment, the research method reaches 6.45 bit/s/Hz, an improvement of 1.23 bit/s/Hz compared to MADDPG that focuses on continuous optimization, and an improvement of 1.60 bit/s/Hz compared to QMIX that focuses on discrete collaboration. In Fig.9(b), when the number of sub-channels increases to 50, the research method can still maintain a utilization level of 87.56%, which is 17.38% higher than WMMSE + Heuristic Association. In an extremely restricted environment with 10 sub-channels, the utilization rate of the research method is as high as 98.52%. In summary, through the deep integration of "discrete decision-making + continuous optimization", the research method can more effectively tap the system potential and ensure a high level of resource utility when resources are scarce.

3.2. Robustness, Stability, and Training Efficiency Verification

After verifying the performance of MADDQN and DMADDPG, this study then conducts verification of robustness, stability and training efficiency. The experimental parameters and environment are the same as above. First, channel dynamics/disturbance robustness verification is performed. The control group is the same as Fig.8, and the results are shown in Fig.10.



(a) Comparison of system throughput fluctuations (b)Comparison of convergence stability indices

Figure 10. Channel dynamics/disturbance robustness analysis

In Fig.10(a), in terms of system performance fluctuations, when the channel fluctuation intensity increases to 12 dB, the throughput standard deviation of the research method only rises to 18.42 Mbps. This is a reduction of 28.43 Mbps compared to the centralized DQN-DDDPG, which is severely affected by environmental non-stationarity, and a reduction of 31.10 Mbps compared to the WMMSE + Heuristic Association, which has difficulty in iterative convergence. In Fig.10(b), in terms of convergence stability, in the face of a strong disturbance of 12 dB, the return variance of the research method is maintained at 3.24×10^3, which is significantly lower than the 9.45×10^3 of DQN-DDDPG and the 7.15×10^3 of MADDPG. In

summary, the research method can effectively suppress the policy oscillation caused by channel disturbance, and shows strong operational stability in dynamic heterogeneous scenarios.

Finally, the experiment is conducted to verify the training efficiency and convergence speed, and MADDPG, QMIX, DQN-DDDPG, and Multi-Agent Proximal Policy Optimization (MAPPO) are selected as controls. To quantify convergence speed, define the metric of "number of turns required to achieve optimal performance of 0.9". During the complete training process, each algorithm records the sliding average return for every 100 rounds. The average return at the end of training (last 500 rounds) is taken as the "final convergence return" of the algorithm. When the sliding average return reaches 90% of the final convergence return for the first time, the corresponding number of training epochs is the indicator value, as shown in Table 3.

Table 3. Performance of training efficiency and convergence stability

Method	MADDQN-DMADDPG	MADDPG	QMIX	DQN-DDDPG	MAPPO
Episodes to 0.9 Best (Episodes)	1245.32 ± 48.56*	2380.15 ± 112.45	1756.40 ± 82.34	3450.62 ± 215.10	4120.85 ± 250.60
Final Avg Return	1865.42 ± 32.15*	1622.18 ± 54.30	1445.75 ± 62.18	1782.55 ± 45.22	1588.60 ± 72.45
Variance of Return (×102)	11.52 ± 1.25*	45.32 ± 5.12	28.45 ± 3.20	52.18 ± 6.85	31.42 ± 4.15
Inference Time (ms/decision)	12.45 ± 0.85*	14.12 ± 1.15	11.56 ± 0.72	42.18 ± 3.45	15.65 ± 1.32
Comm. Overhead (KB/decision)	82.15 ± 5.42*	115.60 ± 8.12	78.42 ± 4.56	455.32 ± 28.50	128.45 ± 10.25

Note: "**" denotes a highly significant difference compared with the control group ($p<0.01$). "*" denotes a significant difference ($p<0.05$).

In Table 3, for convergence speed, the number of rounds required for the final performance of the research method is 1245.32, which is 2205.30 and 2875.53 rounds less than DQN-DDDPG and MAPPO. In terms of return quality and stability, the final average return of the research method increased to 1865.42, and the return variance after convergence was reduced by 33.80×10^2 compared with MADDPG. In terms of operating efficiency, the inference time of the research method is reduced by 29.73 ms compared with DQN-DDDPG, and the communication overhead per decision is also reduced by 373.17 KB. In summary, the proposed method not only improves the efficiency of training samples through decoupling optimization of discrete and continuous decision spaces, but also takes into account the low latency and low bandwidth occupancy characteristics of DE. To comprehensively evaluate the practicality of the proposed method, this section quantitatively analyzes the computational complexity and convergence time of MADDQN-DMADDPG and its comparative methods. Complexity analysis mainly focuses on the computational complexity required for each decision step, while convergence time evaluates the training resources required for the algorithm to achieve stable performance. The specific results are shown in Table 4.

Table 4. Comparative analysis of computational complexity and convergence time

Performance metrics	MADDQN-DMADDPG	DQN-DDDPG	MADDPG	QMIX	WMMSE+Heuristic
Execution phase FLOPs (×10^6)	4.2±0.3	15.8±1.1**	5.5±0.4*	3.8± 0.3	25.0±2.0**
Number of training epochs to achieve 90% optimal performance (epoch)	1245±49	3451±215**	2380±112**	1756±82**	/
Total training time (h)	8.7±0.4	21.5±1.2**	14.2±0.8**	11.8±0.6**	/
Average inference time (ms)	12.45±0.85	42.18±3.45***	14.12±1.15*	11.56±0.72	120.50±10.50**
Theoretical time complexity	$O(N \cdot (d_s+d_a)^2)$	$O((N \cdot d_s)^2$	$O(N^2 \cdot d_a^2)$	$O(N \cdot \log N \cdot d_s)$	$O(I \cdot N^3)$

Note: Compared with MADDQN-DMADDPG, * represents $p<0.05$, and ** represents $p<0.01$.

The results in Table 4 quantitatively reveal the comprehensive advantages of the proposed MADDQN-DMADDPG method in terms of computational efficiency. In the execution phase, this method only requires about 4.2 million floating-point operations, which is much lower in computational burden than the centralized DQN-DDDPG's 15.8 million operations and the traditional optimization method WMMSE's 25.0 million operations, demonstrating better computational lightweight. In terms of training efficiency, this method can achieve 90% optimal performance in only about 1245 training epochs and a total time of 8.7 hours, and its convergence speed is significantly faster than other compared deep

reinforcement learning methods. The average single decision inference time of this method is only 12.45ms, which is several times faster than DQN-DDDPG's 42.18 milliseconds and far behind WMMSE's 120.50ms, fully demonstrating its potential to meet the real-time scheduling needs of large-scale IoT systems. Overall, the proposed method performs excellently in terms of computational complexity, convergence speed, and real-time inference delay, laying a solid foundation for its transition from simulation verification to practical deployment. To enhance the comparison with cutting-edge research, the study further selected two representative and latest multi-agent deep reinforcement learning scheduling algorithms as controls. Including multi-agent hybrid DRL algorithm and policy gradient DRL algorithm. The study was compared under the same experimental environment, and the results are shown in Table 5.

Table 5. Comparison and Analysis of progressiveness of Different Algorithms

Performance metrics	MADDQN-DMADDPG	Multi agent hybrid DRL	Strategic gradient DRL
Total throughput (Mbps)	468.52 ± 10.24	423.68 ± 14.37*	395.42 ± 16.85**
Energy Efficiency (Mbit/J)	18.76 ± 0.52	15.93 ± 0.80*	14.20 ± 0.95**
QoS non-compliance rate (%)	3.85 ± 0.41	7.25 ± 0.78*	11.36 ± 1.12**
Jain Fairness Index	0.93 ± 0.02	0.88 ± 0.03*	0.85 ± 0.04*
Average user speed (Mbps)	2.48 ± 0.15	1.85 ± 0.22*	1.42 ± 0.25**
Large scale non-compliance rate (%)	16.82 ± 1.52	28.91 ± 2.35**	35.74 ± 3.01**
Number of rounds required for convergence	1245 ± 49	2010 ± 98*	3320 ± 185**
Average reasoning time (ms)	12.45 ± 0.85	18.33 ± 1.45*	38.75 ± 3.20**

Note: Compared with MADDQN-DMADDPG, * represents $p<0.05$, and ** represents $p<0.01$.

According to the comparison results in Table 5, the MADDQN-DMADDPG method proposed in this study significantly outperforms the two cutting-edge multi-agent deep reinforcement learning algorithms as controls in all key performance indicators. In terms of core communication performance, this method achieved a total throughput of 468.52 megabits per second and an energy efficiency of 18.76 megabits per joule, while controlling the service quality non-compliance rate at 3.85% and achieving a Jain fairness index of 0.93, surpassing the control algorithm comprehensively. Especially in large-scale network scalability testing, when the user scale is expanded to two hundred, this method can still maintain an average user rate of 2.48 megabits per second and control the large-scale non-compliance rate at 16.82%. Its performance degradation is much lower than that of the control algorithm. In addition, this method exhibits significant advantages in training and execution efficiency, requiring only 1245 rounds of training to converge, and the average inference time for a single decision is only 12.45 milliseconds. Its convergence speed and real-time response capability are significantly ahead. Overall, compared with existing advanced algorithms, the proposed method exhibits comprehensive and significant superiority in multiple dimensions such as system throughput, resource efficiency, service quality, fairness, large-scale scalability, and algorithm efficiency. The study horizontally compared the proposed methods and summarized their performance in key dimensions, as shown in Table 6.

Table 6. Horizontal comparison results of different methods

Method source	Van Chien *et al.* [5]	Lei *et al.* [6]	Li *et al.* [7]	Lin *et al.* [8]	De Souza *et al.* [9]	Cheng *et al.* [10]	Proposed Method
Maximum number of supported users	50	10	30	100	500	20	200
Number of epochs/iterations required for convergence	180	140	1520	1950	0	2980	1245
Total throughput (Mbps)	322	210	284	380	408	356	468.52
Energy Efficiency (Mbit/J)	15.2	11.5	12.7	13.8	16	14.3	18.76
Service quality breach rate (%)	8.5	14.2	11.2	9.8	7.2	12.5	3.85
Average user speed (Mbps)	6.44	21.0[2]	9.47	3.8	0.82	17.8	2.48

The horizontal comparison results in Table 6 indicate that the method proposed in this paper demonstrates significant advantages in multiple key performance indicators. Specifically, the proposed method achieved the highest total throughput (468.52 Mbps) and energy efficiency (18.76 Mbit/J), while keeping the service quality breach rate at the lowest level (3.85%), showing significant improvement compared to the comparative methods in references [5-10]. In terms of convergence efficiency, the

proposed method only required 1245 rounds to converge, significantly better than that of the deep reinforcement learning methods proposed by Li *et al.* [7], Lin *et al.* [8], and Cheng *et al.* [10] (1520-2980 rounds), and far lower than that of traditional heuristic methods [9] (no training required but suboptimal performance). Although the maximum number of users supported by De Souza *et al.* [9] (500) was higher than the 200 supported by the proposed method, its average user rate was only 0.82 Mbps, far lower than that of the 2.48 Mbps supported by the proposed method, indicating that the proposed method can still maintain higher single user service quality even when the user scale is large. Overall, this method achieves the best balance between system throughput, resource efficiency, convergence speed, and service reliability, verifying its comprehensive superiority in dynamic heterogeneous IoT scenarios.

4. Discussion And Conclusion

Aiming at the highly coupled problem of user association and CRA in heterogeneous IoT, where traditional methods struggle to balance system performance and scalability, this study proposes a joint user association and channel resource scheduling method based on MADDQN-DMADDPG. The method decouples discrete user scheduling from continuous CRA and achieves multi-agent collaborative optimization under a centralized training with decentralized execution (CTDE) framework. In experiments, the total system throughput of the proposed method reached 468.52 Mbps, which was 116.42 Mbps higher than that of WMMSE. The energy efficiency reached 18.76 Mbit/J, significantly outperforming QMIX (12.56 Mbit/J). When the user scale was expanded to 200, the proposed method could still maintain an average rate of 2.48 Mbps while keeping the non-compliance rate at 16.82%. In extremely resource-constrained scenarios, the proposed method achieved a spectrum efficiency of 6.45 bit/s/Hz and a resource utilization rate of up to 98.52%. Furthermore, under strong channel disturbance, the standard deviation of its throughput fluctuation was only 18.42 Mbps, the variance of the convergence return was maintained at 3.24×10^{3}, and the number of convergence rounds was reduced to 1245. The results show that the proposed method delivers high performance, strong robustness, and good training efficiency in large-scale heterogeneous IoT scenarios.

This research focuses on the complex coupling between user association and channel resource joint scheduling in large-scale heterogeneous IoT. Its main contributions are threefold. First, by integrating a multi-agent double deep Q-network with a double-multi-agent deep deterministic policy gradient, the MADDQN-DMADDPG model is constructed. This framework innovatively decouples and collaboratively optimizes discrete user association decisions and continuous CRA, effectively addressing the shortcomings of existing methods in mixed action space modeling and multi-agent collaboration. Second, dedicated algorithm components are designed for the coupled optimization problem. Beyond the standard deep reinforcement learning (DRL) algorithm, a multi-objective reward function integrating real-time throughput, fairness penalty, and switching overhead is devised to address the coupling between resource competition and interference coordination. Additionally, a channel interference attention module is introduced into the agent network. Third, high performance and high scalability are effectively unified through the CTDE architecture, which overcomes the non-stationarity of multi-agent environments via centralized training while ensuring low complexity in distributed deployment.

However, this study is mainly conducted in a simulation environment, which underestimates the impact of channel uncertainty and system implementation complexity on algorithm performance in real-world deployment. Moreover, in practical network deployment, synchronizing state information and coordinating policies among distributed agents will introduce inevitable signaling delays, potentially affecting the real-time performance and consistency of scheduling decisions in highly dynamic wireless environments. Future research should explore low-overhead synchronization mechanisms and delay-robust training algorithms to ensure the practical performance in engineering applications.

CRediT Author Contribution Statement

Yankai Xie: Conceptualization; Methodology; Software; Validation; Formal analysis; Investigation; Resources; Data Curation; Writing - Original Draft; Writing - Review & Editing; Visualization; Supervision.

References

[1] Yonghui Sun, Junjie Xu and Shuguang Cui, "User association and resource allocation for MEC-enabled IoT networks", *IEEE Transactions on Wireless Communications*, Print ISSN: 1536-1276, Online ISSN: 1558-2248, October 2022, Vol. 21, No. 10, pp. 8051–8062, Published by IEEE, DOI: 10.1109/TWC.2022.3163809, Available: https://ieeexplore.ieee.org/document/9737513.

[2] Nilanjan Biswas, Zijian Wang, Luc Vandendorpe and Hamed Mirghasemi, "On Joint Cooperative Relaying, Resource Allocation, and Scheduling for Mobile Edge Computing Networks", *IEEE Transactions on Communications*, Print ISSN: 0090-6778, Online ISSN: 1558-0857, September 2022, Vol. 70, No. 9, pp. 5882-5897, Published by IEEE, DOI: 10.1109/TCOMM.2022.3191681, Available: https://ieeexplore.ieee.org/document/9831783.

[3] Majd Shakhatreh, Hazim Shakhatreh and Ahmad Ababneh, "Efficient 3D Positioning of UAVs and User Association Based on Hybrid PSO-K-Means Clustering Algorithm in Future Wireless Networks", *Mobile Information Systems*, Print ISSN: 1574-017X, Online ISSN: 1875-905X, 2023, Vol. 2023, No. 1, pp. 6567897, Published by Hindawi, DOI: 10.1155/2023/6567897, Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2023/6567897.

[4] Hyun Jung Park, Hyeon Woong Kim and Sung Ho Chae, "Deep-Learning-Based Resource Allocation for Transmit Power Minimization in Uplink NOMA IoT Cellular Networks", *IEEE Transactions on Cognitive Communications and Networking*, Print ISSN: 2332-7731, Online ISSN: 2332-774X, September 2023, Vol. 9, No. 3, pp. 708-721, Published by IEEE, DOI: 10.1109/TCCN.2023.3254515, Available: https://ieeexplore.ieee.org/document/10064048.

[5] Trinh Van Chien, Ha An Le, Ta Hai Tung, Hien Quoc Ngo and Symeon Chatzinotas, "Joint Power Allocation and User Scheduling in Integrated Satellite–Terrestrial Cell-Free Massive MIMO IoT Systems", *IEEE Internet of Things Journal*, Print ISSN: 2327-4662, Online ISSN: 2372-2541, 2024, Vol. 11, No. 20, pp. 32883-32900, Published by IEEE, DOI: 10.1109/JIOT.2024.3423008, Available: https://ieeexplore.ieee.org/document/10586868.

[6] Hongjiang Lei, Haosi Yang, Ki-Hong Park, Imran Shafique Ansari, Jing Jiang *et al.*, "Joint Trajectory Design and User Scheduling for Secure Aerial Underlay IoT Systems", *IEEE Internet of Things Journal*, Print ISSN: 2327-4662, Online ISSN: 2372-2541, 2023, Vol. 10, No. 15, pp. 13637-13648, Published by IEEE, DOI: 10.1109/JIOT.2023.3262697, Available: https://ieeexplore.ieee.org/document/10083188.

[7] Sisi Li, Yong Zhang, Siyu Yuan and Tengteng Ma, "User Scheduling and Slicing Resource Allocation in Industrial Internet of Things", *China Communications*, Print ISSN: 1673-5447, Online ISSN: 1673-5447, June 2023, Vol. 20, No. 6, pp. 368-381, Published by China Institute of Communications, DOI: 10.23919/JCC.2023.00.017, Available: https://ieeexplore.ieee.org/document/10056792.

[8] Lixia Lin, Wen'an Zhou, Zhicheng Yang and Jianlong Liu, "Deep Reinforcement Learning-Based Task Scheduling and Resource Allocation for NOMA-MEC in Industrial Internet of Things", *Peer-to-Peer Networking and Applications*, Print ISSN: 1936-6450, Online ISSN: 1936-6450, January 2023, Vol. 16, No. 1, pp. 170-188, Published by Springer Nature, DOI: 10.1007/s12083-022-01348-x, Available: https://link.springer.com/article/10.1007/s12083-022-01348-x.

[9] João Henrique Inacio de Souza, José Carlos Marinello Filho, Abolfazl Amiri and Taufik Abrão, "QoS-Aware User Scheduling in Crowded XL-MIMO Systems Under Non-Stationary Multi-State LoS/NLoS Channels", *IEEE Transactions on Vehicular Technology*, Print ISSN: 0018-9545, Online ISSN: 1939-9359, June 2023, Vol. 72, No. 6, pp. 7639-7652, Published by IEEE, DOI: 10.1109/TVT.2023.3243488, Available: https://ieeexplore.ieee.org/document/10040750.

[10] Zhipeng Cheng, Minghui Liwang, Ning Chen, Lianfen Huang, Nadra Guizani *et al.*, "Learning-Based User Association and Dynamic Resource Allocation in Multi-Connectivity Enabled Unmanned Aerial Vehicle Networks", *Digital Communications and Networks*, Print ISSN: 2352-8648, Online ISSN: 2352-8648, January 2024, Vol. 10, No. 1, pp. 53-62, Published by Elsevier, DOI: 10.1016/j.dcan.2022.05.026, Available: https://www.sciencedirect.com/science/article/pii/S2352864822001195.

[11] Sara Salim, Nour Moustafa and Martin Reisslein, "Cybersecurity of Satellite Communications Systems: A Comprehensive Survey of the Space, Ground, and Links Segments", *IEEE Communications Surveys & Tutorials*, Print ISSN: 1553-877X, Online ISSN: 2373-745X, 2025, Vol. 27, No. 1, pp. 372-425, Published by IEEE, DOI: 10.1109/COMST.2024.3408277, Available: https://ieeexplore.ieee.org/document/10546924.

[12] Wei Jiang, Qiuheng Zhou, Jiguang He, Mohammad Asif Habibi, Sergiy Melnyk *et al.*, "Terahertz Communications and Sensing for 6G and Beyond: A Comprehensive Review", *IEEE Communications Surveys & Tutorials*, Print ISSN: 1553-877X, Online ISSN: 2373-745X, 2024, Vol. 26, No. 4, pp. 2326-2381, Published by IEEE, DOI: 10.1109/COMST.2024.3385908, Available: https://ieeexplore.ieee.org/document/10494372.

[13] Gaurav Kumar Pandey, Devendra Singh Gurjar, Suneel Yadav, Yuming Jiang and Chau Yuen, "UAV-Assisted Communications with RF Energy Harvesting: A Comprehensive Survey", *IEEE Communications Surveys & Tutorials*, Print ISSN: 1553-877X, Online ISSN: 2373-745X, 2025, Vol. 27, No. 2, pp. 782-838, Published by IEEE, DOI: 10.1109/COMST.2024.3425597, Available: https://ieeexplore.ieee.org/document/10589561.

[14] Mesfin Leranso Betalo, Supeng Leng, Hayla Nahom Abishu, Fayaz Ali Dharejo, Abegaz Mohammed Seid *et al.*, "Multi-Agent Deep Reinforcement Learning-Based Task Scheduling and Resource Sharing for O-RAN-Empowered Multi-UAV-Assisted Wireless Sensor Networks", *IEEE Transactions on Vehicular Technology*, Print ISSN: 0018-9545, Online ISSN: 1939-9359, July 2024, Vol. 73, No. 7, pp. 9247-9261, Published by IEEE, DOI: 10.1109/TVT.2023.3330661, Available: https://ieeexplore.ieee.org/document/10314006.

[15] Yudian Huang, Meng Li, F. Richard Yu, Pengbo Si, Haijun Zhang *et al.*, "Resources Scheduling for Ambient Backscatter Communication-Based Intelligent IIoT: A Collective Deep Reinforcement Learning Method", *IEEE Transactions on Cognitive Communications and Networking*, Print ISSN: 2332-7731, Online ISSN: 2332-774X, 2024, Vol. 10, No. 2, pp. 634-648, Published by IEEE, DOI: 10.1109/TCCN.2023.3330065, Available: https://ieeexplore.ieee.org/document/10308961.

[16] Tinghao Zhang, Kwok-Yan Lam and Jun Zhao, "Deep Reinforcement Learning Based Scheduling Strategy for Federated Learning in Sensor-Cloud Systems", *Future Generation Computer Systems*, Print ISSN: 0167-739X, Online ISSN: 1872-7115, 2023, Vol. 144, No. 1, pp. 219-229, Published by Elsevier, DOI: 10.1016/j.future.2023.03.009, Available: https://www.sciencedirect.com/science/article/pii/S0167739X23000870.

[17] Ahmed A. Ismail, Nour Eldeen Khalifa and Reda A. El-Khoribi, "A Survey on Resource Scheduling Approaches in Multi-Access Edge Computing Environment: A Deep Reinforcement Learning Study", *Cluster Computing*, Print ISSN: 1386-7857, Online ISSN: 1573-7543, January 2025, Vol. 28, No. 1, pp. 184, Published by Springer Nature, DOI: 10.1007/s10586-024-04893-7, Available: https://link.springer.com/article/10.1007/s10586-024-04893-7.

[18] Yuqing Cheng, Zhiying Cao, Xiuguo Zhang, Qilei Cao and Dezhen Zhang, "Multi Objective Dynamic Task Scheduling Optimization Algorithm Based on Deep Reinforcement Learning", *The Journal of Supercomputing*, Print ISSN: 0920-8542, Online ISSN: 1573-0484, March 2024, Vol. 80, No. 1, pp. 6917-6945, Published by Springer Nature, DOI: 10.1007/s11227-023-05714-1, Available: https://link.springer.com/article/10.1007/s11227-023-05714-1.

[19] Sudheer Mangalampalli, Ganesh Reddy Karri, Mohit Kumar, Osama Ibrahim Khalaf, Carlos Andres Tavera Romero *et al.*, "DRLBTSA: Deep Reinforcement Learning Based Task-Scheduling Algorithm in Cloud Computing", *Multimedia Tools and Applications*, Print ISSN: 1380-7501, Online ISSN: 1573-7720, June 2023, Vol. 83, No. 3, pp. 8359-8387, Published by Springer Nature (Kluwer Academic Publishers), DOI: 10.1007/s11042-023-16008-2, Available: https://link.springer.com/article/10.1007/s11042-023-16008-2.

[20] Harshala Shingne and R. Shriram, "Mutated Deep Reinforcement Learning Scheduling in Cloud for Resource-Intensive IoT Systems", *Wireless Personal Communications*, Print ISSN: 0929-6212, Online ISSN: 1572-834X, 2023, Vol. 132, No. 1, pp. 2143-2155, Published by Springer Nature, DOI: 10.1007/s11277-023-10709-5, Available: https://link.springer.com/article/10.1007/s11277-023-10709-5.

[21] Yifan Chen, Shaomiao Chen, Kuan-Ching Li, Wei Liang and Zhiyong Li, "DRJOA: Intelligent Resource Management Optimization Through Deep Reinforcement Learning Approach in Edge Computing", *Cluster Computing*, Print ISSN: 1386-7857, Online ISSN: 1573-7543, October 2023, Vol. 26, No. 1, pp. 2897-2911, Published by Springer Nature, DOI: 10.1007/s10586-022-03768-z, Available: https://link.springer.com/article/10.1007/s10586-022-03768-z.

[22] S. Nagarajan, P. Shobha Rani, M. S. Vinmathi, V. Subba Reddy, Angel Latha Mary Saleth *et al.*, "Multi Agent Deep Reinforcement Learning for Resource Allocation in Container-Based Clouds Environments," *Expert Systems*, Print ISSN: 0266-4720, Online ISSN: 1468-0394, 2025, Vol. 42, No. 1, pp. e13362, Published by Wiley, DOI: 10.1111/exsy.13362, Available: https://onlinelibrary.wiley.com/doi/10.1111/exsy.13362.

[23] Erdem Erdemir, Erkan Kaplanoglu, Cihan Uyanik and Gazi Akgun, "Motion Trajectory Estimation for Hand Grasping States Using a Deep Learning Approach", *Sensors and Wearable Technologies (SWT)*, Print ISSN: N/A, Online ISSN: N/A, August 2025, Vol. 1, No. A5, pp. A5, Published by Bon View Press, Available: https://ojs.bonviewpress.com/index.php/SWT/article/view/5659.