Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on Deep Reinforcement Learning-based Coordinated Control Strategy for Multi-Intelligent Bodies under Software System Architecture

Lin Li[1,✉], Zehui Yang[2], Gaiqiang Yang[3]

[1] *School of Information Science and Technology, Shanxi Finance and Taxation College, Taiyuan, Shanxi, 030024, China*

[2] *School of Big Data, Shanxi Finance and Taxation College, Taiyuan, Shanxi, 030024, China*

[3] *School of Environmental Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi, 030024, China*

ABSTRACT

In today's society, a single intelligent body does not meet the needs of complex tasks, and coordinated control of multiple intelligences becomes an important solution. In this regard, this paper carries out the research on the coordinated control strategy of multiple intelligences supported by deep reinforcement learning technology. Aiming at the problems of uneven task distribution and unsatisfactory decision consistency arising from the collaborative decision making of multiple intelligences under the software system architecture, a hierarchical multi-intelligence collaborative decision-making algorithm based on the AC framework is proposed to realize the information exchange and decision-making collaboration among intelligences, so as to improve the efficiency of coordinated control. However, with the increase of the number of multi-intelligents, the algorithm will have the problem of upper and lower level non-smoothness, in order to solve this problem, a multi-intelligents collaborative algorithm based on role parameter sharing is designed. Finally, the research scheme of this paper is evaluated and analyzed from multiple dimensions. When the number of intelligences increases by 5, the reward value of this paper's algorithm does not show a decreasing trend, which indicates that this paper's algorithm is able to handle the control coordination problem in the case of a small number of intelligences. When the number of intelligences increases by 15, the original method shows a decreasing trend, while in the multi-intelligence body collaboration algorithm based on the sharing of role parameters, the performance is very bright, which ensures the coordinated control effect of multi-intelligence bodies under the software system architecture.

Keywords: deep reinforcement learning, AC framework, multiple intelligences, coordinated control

1. Introduction

Since the world's first industrial robot was introduced in the 1950s, robotics has become a cutting-edge technology after more than half a century of development, and its application areas, in addition

✉ Corresponding author.
E-mail address: bhglxq2024@163.com (L. Li).
Received 20 March 2024; Revised 05 June 2024; Accepted 20 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-343
© 2025 The Author(s). Published by Combinatorial Press. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

to traditional industries, also involve all walks of life as well as all aspects of people's lives [1-3]. In recent years, due to the increased complexity and danger of the environment in which robots perform tasks, a single robot can no longer meet the requirements of the task. Therefore, multi-intelligent body systems have emerged.

Multi-intelligent body system does not mean that multiple intelligent individuals are simply added together, but refers to a collection of multiple interacting intelligent individuals, which have a certain degree of autonomy, and at the same time can coordinate their behaviors by exchanging information and cooperatively complete a task or solve a problem [4]. Multi-intelligent body system makes up for the lack of individual robot ability, and its advantage is mainly reflected in the fact that each intelligent body in the system has a certain degree of autonomy, and its functions are different, which can not only effectively reduce the complexity of individual intelligent bodies, but also improve the ability of the whole system to deal with problems [5-6]. Secondly, the design of multi-intelligent body system pursues the coordination consistency of the system, and the tasks are executed in parallel by the intelligences through mutual coordination, which greatly improves the efficiency of the system [7-8]. Again, multi-intelligent body system adopts different coordination algorithms as well as expert systems with different functions to collaborate in solving various difficulties in the system, which effectively improves the system's ability to deal with problems [9-10]. Finally, the multi-intelligent body system integrates modular information together, which not only improves the information utilization of the system, but also makes the system easy to expand and effectively reduces the total cost of the system [11-12]. Due to the characteristics of spatial distribution, temporal parallelism, fault tolerance, high utilization, and low cost, the multi-intelligent body system is thus widely used in agriculture, industry, multi-domain exploration, and national defense business [13-16]. Intelligent bodies in the multi-intelligent body system can utilize the sensors and communication devices carried by themselves as well as their flexible mobility to achieve the purpose of collaboration with each other, which is used to replace human beings to complete the tasks targeted in special environments, in which the control ability of multi-intelligent bodies plays a decisive role in the efficiency and effectiveness of the completion of these complex tasks [17-20]. For example, intelligent mobile robot swarms replace human activities in tasks such as search and rescue and exploration, unmanned tank swarms, unmanned fighter swarms, etc. replace the military in combat, robotic production line, reduce casualties, improve efficiency, reduce costs, etc. It can be seen that the application of multi-intelligent body system can accelerate the modernization process of industry, agriculture and national defense construction to a certain extent.

However, as the environment in which multi-intelligent body systems work is becoming more and more complex, many factors (noise in the environment or limitations of the intelligent body's own sensors, etc.) affect the ability of coordinated control of multi-intelligent bodies, and uncertainty and low model adaptability are the challenges faced by the current research on coordinated control of multi-intelligent bodies [21-23]. If these factors are not well handled, the system will face various bad conditions. Theoretically, intelligences can coordinate their behaviors through information exchange and sharing, but due to issues such as bandwidth and network structure, information cannot all be broadcasted to all intelligences of the system [24-25]. Moreover, in some real-world environments, many communication methods are unreliable, which further impacts the system. Therefore, there is a need for an in-depth study of coordinated control of multiple intelligences.

Multi-intelligent body system has been a research hotspot in the international academic community since it was proposed. Its main research contents are the system mechanism of the system, the communication mode, the coordination and control strategy as well as the dissipation of conflicts, etc., among which the coordinated control of multi-intelligent body has always been the key problem in the research of multi-intelligent body system [26-27]. Zhang et al [28] utilized a single-wheel control approach and multi-intelligentsia theory for coordinated control between an anti-lock braking system and an adaptive headlight system for electric vehicles, carried out by distributed model predictive

control. Yuan et al [29] developed a multi-intelligence coordinated control system using hierarchical control and blackboard modeling for a coordinated control strategy of active collision avoidance and hysteresis reduction for intelligent vehicles to assist in rule-based control decisions. Li et al [30] faced the problem of coordinated control of hydrogen and air supply systems in proton exchange membrane fuel cells and designed a multi-intelligent deep deterministic strategy gradient (DDSG) algorithm based on multiple reinforcement learning explorers and multiple control algorithm explorers. Wang et al [31] applied adaptive variational modal decomposition and multi-intelligent dual-delay DDSG to predict and optimize the power output of wind power generation, respectively, and coordinated the dynamics-induced noise in the algorithm by combining with the "stable" Lévy noise, so as to achieve the coordinated control of the wind turbine and the hybrid energy storage system. Zhang et al [32] introduced the attention mechanism into multi-intelligence deep reinforcement learning (DRL), and the multi-intelligence was trained centrally and decentralized to obtain energy management strategies and perform strategy execution, respectively, to achieve the coordinated control of multiple energy sources for natural gas, electricity, and fresh water. Jing et al [33] created a two-layer collaborative control strategy model for integrated energy sources, with a hybrid game decision-making model supported by a multi-intelligence DDSG algorithm in the upper layer, and an electricity-gas flow computation model in the lower layer, where the results of energy data computation in the lower layer are fed back to the upper layer to assist in decision-making. Wang et al [34] obtained a cooperative control strategy for power and heat by constructing a hierarchical distributed control model and a distributed control model with the support of a multi-intelligent body system, and designing an operating rule for following thermal loads and a hill-climbing search model under these two models, respectively. Rahman and Oo [35] jointly assembled a coordinated control strategy for a distributed energy microgrid with a graph-theory based multi-intelligentsia communication architecture and the Ziegler-Henness method. In addition, Wang et al [36] proposed a time-invariant transform-optimized distributed interval observer to refine an uncertain directed network multi-intelligentsia system as a way to achieve robust coordinated control of this system. Li and Zhong [37], in order to solve the uncertainty problem of delay and noise for coordinated group control in multi-intelligence systems, contributed 2 group control protocols which impose weak and strong consensus and constraints on the intelligences in different subgroups, respectively. Although the theoretical results of these researches are very mature, they still face great challenges, including the problems of scalability, optimization, coordination of algorithms and complexity of modeling specific behaviors.

DRL applied in the above research is a popular direction in the field of artificial intelligence, which synthesizes the advantages of deep learning and reinforcement learning, and is able to solve more complex problems to a certain extent. Its core idea is to place an intelligent body in a certain environment, and receive the state of the environment at each moment, choose the correct action according to the current state, and get the corresponding reward, learn an optimal strategy so that the intelligent body can get the maximum total reward [38]. The software system architecture involves all aspects of the system, including components, modules, data flow, and the performance of the system, etc. A reasonable software system architecture can improve the maintainability, scalability, and reliability of the system so as to satisfy the user's needs [39]. Combining DRL with software system architecture provides ideas for breaking through the technical difficulties in multi-intelligent body cooperative control strategy.

In this paper, we propose a hierarchical multi-intelligence coordination and control strategy based on the AC framework for the control coordination problem in multi-intelligence software system architectures, which divides the multi-intelligence coordination and control problem into two levels; the high-level strategy is learning for the whole team, with the goal of coordinating the decision-making among different intelligences to achieve the overall reward optimization. The low-level strategy is learning for each intelligent body, and the goal is to learn how to take optimal actions to realize the requirements of the high-level strategy. When the number of intelligences increases rapidly,

the algorithm then suffers from the upper and lower level unsteady problem, in response to which a multi-intelligence collaborative algorithm based on role parameter sharing is designed to further improve the control coordination in multi-intelligence software systems. Finally, a series of experimental preparations are identified to validate and analyze the research program specified in this paper.

2. Hierarchical Multi-Intelligent Body Coordinated Control Strategy Based on AC Framework

Aiming at the problems of irrational task allocation and poor decision consistency in multi-intelligence body collaborative decision-making under software system architecture, this paper proposes a hierarchical multi-intelligence body collaborative decision-making method based on AC framework. The method improves coordination and control efficiency by dividing the decision-making process into different levels and using the AC framework to realize information exchange and decision-making collaboration among intelligences. At the high level, the top-level intelligences make task decisions, decompose the total task and assign it to the bottom-level intelligences. At the low level, the bottom-level intelligences make action decisions based on subtasks and feedback the results to the high level. The details are as follows:

2.1. Deep Reinforcement Learning and Semi-Markov Decision Processes

2.1.1. Deep reinforcement learning. Traditional reinforcement learning is suitable for dealing with environments with small state and action spaces, and is unable to deal with complex environments or diverse input information when faced with them, such as pictures and videos. The development of deep learning has laid the foundation for further prosperity of reinforcement learning. Traditional reinforcement learning is mainly categorized into two directions: one is the value function-based method and the other is the policy-based algorithm. The value function-based method alone is not applicable to continuous action space or taking stochastic strategy, because its variation may lead to unstable strategy gradient, which affects the convergence of the model and its training speed. The strategy gradient-based algorithm alone also causes feedback delays and the intelligences may not be able to learn effective strategies. Moreover, since the cumulative rewards are available only after the end of a round of the game, it makes the data utilization low and the variance of the gradient too large, which can easily cause the intelligent body to fall into the local optimal solution. Thus, combining Actor network based on strategy method and Critic network based on value function method becomes very necessary [40]. The Actor-Critic framework combines the advantages of both, where the Actor network takes as input the value of the state in which the intelligence is in, and outputs the probability of executing a given action, and then either selects the action with the largest probability value as the next action to be executed by the intelligence or outputs directly the action to be executed. The Critic network, also known as a judgment network, takes as input the state and action of the intelligence, and outputs the expected benefit Q value to be used to assess the goodness of executing this strategy with the expected gain Q value. The Actor network relies on the gradient descent method for updating, while the Critic network amount relies on the TD difference method for updating. The Actor-Critic algorithm is a single-step updating algorithm, i.e., the model is updated once for one step of interaction between the intelligent body and the environment, which improves the training speed of the model. When the variance of the probabilistic strategy is close to 0, each step of the action can output a deterministic strategy through the Actor network. Unlike the DQN algorithm, this algorithm is applicable to both environments with discrete action spaces and continuous action spaces.

2.1.2. Semi-Markov decision processes. Comparison of MDP and SMDP states as shown in Fig. 1, traditional reinforcement learning models are based on the assumption that each action is executed in

only a single time step, which limits their ability to process actions that span multiple time steps. To overcome this limitation, a Semi-Markov Decision Process (SMDP) is formed by subdividing the traditional Markov Decision Process (MDP) over multiple time scales [41]. In SMDP, the time step between two decisions is set as T. Depending on whether T is an integer or a real number, it can be categorized into two types: discrete-time SMDP and continuous-time-discrete-event SMDP, the former of which is evaluated by an evaluator at the end of the time step, and this evaluation serves as a guide and feedback for the next layer of training in this strategy. In the latter case, the lower layer does not fix the number of time steps and needs to give the end condition of the lower layer as a judgment to return to the upper layer for decision making.

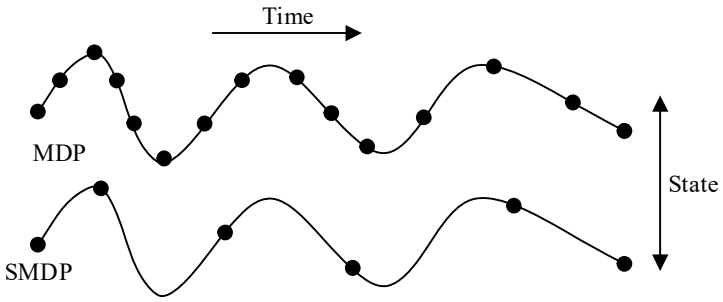


Fig. 1. Comparison of MDP and SMDP status

2.2. Hierarchical multi-intelligence body regulation scheme based on the AC framework

A hierarchical reinforcement learning approach based on the Actor-Critic framework is illustrated and applied to a multi-intelligent body coordinated control scenario. The previous section introduced the theoretical foundation of deep reinforcement learning, and this subsection describes in detail how to integrate this concept into the Actor-Critic model to form a hierarchical multi-intelligent body regulation and control scenario. Details are given below:

2.2.1. Multi-intelligent body modulation program design. The semi-Markov chain unfixed step decision process is shown in Fig. 2, which is used to model the problem in this paper. In the Conceptual Model of Mission Space (CMMS), an EATI template is proposed, and based on the EATI template, the coordinated control process of the intelligences is divided into two parts to be carried out at high level (HL) and low level (LL). The high-level strategies are responsible for goal assignment and the low-level strategies are responsible for basic operations. The high-level strategy learns stability and provides effective coordination guidance for the low-level strategy, which improves the performance of the multi-intelligent body coordination task.

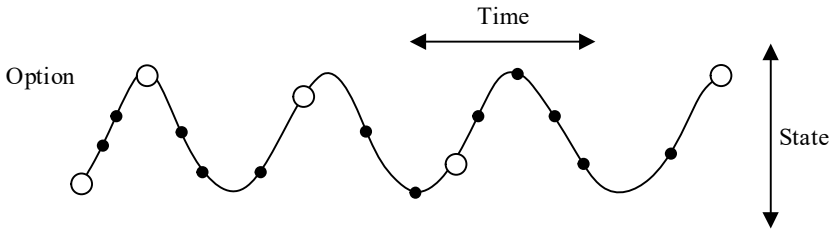


Fig. 2. Semi-markov chain unfixed step decision process

The framework of the hierarchical multi-intelligence body algorithm based on the AC framework is shown in Fig. 3. After receiving the policy A_h given by the HL-Actor network, the LL-Critic network calculates the current state and the Q value under the action according to the action given by the LL-Actor network and updates it by calculating the loss function according to the local reward generated

by the current action, and the LL-Actor network completes the updating based on the policy gradient. At the end of the training process at the bottom layer, a global reward is generated, the HL-Critic network at the top layer calculates a Q value based on A_h and updates it by combining it with the global reward, and the HL-Actor network iteratively updates it based on the policy gradient as well. After continuously going through this cycle, eventually, both networks reach a stable equilibrium state.

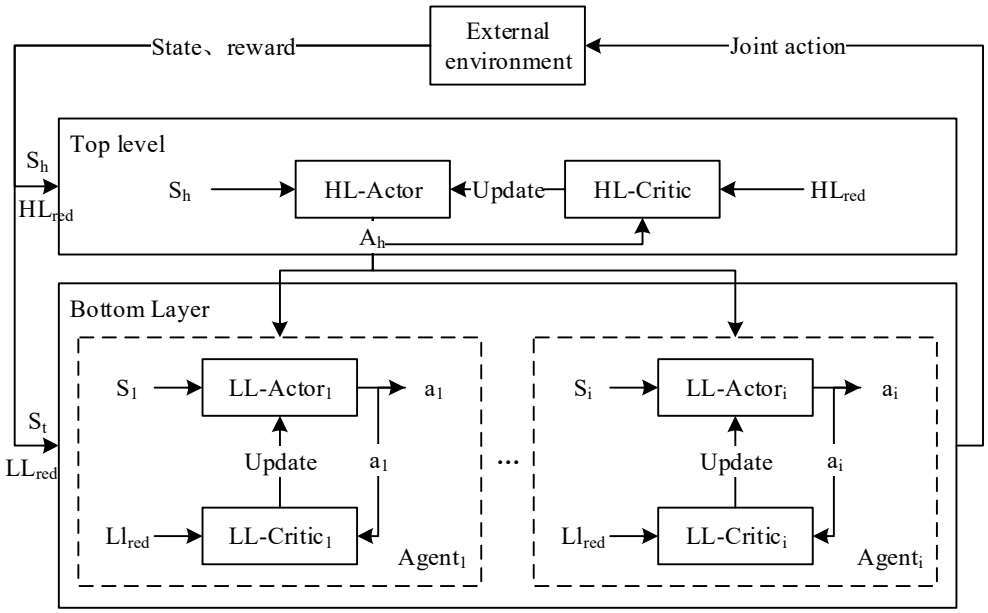


Fig. 3. Hierarchical multi-agent algorithm framework based on AC framework

2.2.2. Hierarchical Multi-Intelligents Based MDP Modeling. Based on the hierarchical structure, the state space and action space are divided. At the high-level decision-making level, macro instructions for the whole cluster are formulated based on comprehensive action information, and at the low-level decision-making level, task units that execute these macro instructions are specifically manipulated to perform specific operations.

1) High-level decision dimension decomposition. For a task scenario, given the total task M, M is first split into sub-task set $\{M_0, M_1, \cdots, M_i\}$ and its strategy π is split into sub-strategy set $\{\pi_0, \pi_1, \cdots, \pi_i\}$. where π_i denotes the strategy for M_i. For each subtask M_i is equivalent to the task planning based MDP model in Section 2.2.2.

2) Low-Level Decision Dimension Decomposition. In a multi-intelligence joint control scenario, the low-level intelligences control the actions of each unit, so the bottom-level intelligences use the MDP model based on action control.

3) Global reward function design. The global reward should be able to guide the intelligences to learn towards the optimal behavior. During the task, the upper layer intelligent body only needs to focus on whether the sub-task selection is appropriate or not, when the task is appropriate to give a positive reward reward, and vice versa is a negative reward reward, in this paper, according to whether the global task is completed to judge whether the sub-task selection is correct or not.

4) Local reward function design. For each intelligence in the lower level, after each simulation step a step reward is obtained based on the executed action, through which the intelligence is guided to complete the given task. Step reward r_{step} consists of multiple sub-rewards weighted as follows, and sub-reward r_k is defined as follows:

Target Distance Reward r_1:

$$r_1 = -k\left(d_t - d_{begin}\right) \tag{1}$$

Where: d_t is the relative distance between the current intelligent body and the target, and d_{begin} is the relative distance between the intelligent body and the target in the initial posture. In order to ensure that the pursuit intelligent body unit efficiently completes the task of surprise defense, the relative distance between the intelligent body and the target is calculated at each time step, and r_1 is set as a negative reward function, which is positively correlated with the reward of the pursuit distance r_1, so that when the intelligent body is far away from the target within the simulation step, r_1 is a negative value. When the smart body is close to the target within the simulation step, r_1 will be positive.

Intelligent body collision bonus r_2:

$$r_2 = -e^{d_{\min}} \tag{2}$$

A reward function of negative exponential form is established r_2 describing the risk of collision between the intelligences, and $d_{\min}$ denoting the current intelligence's closest distance to other intelligences.

To summarize, the step reward of each intelligent body is the weighted sum of the above two reward functions: $r_{step} = r_1 + r_2$.

Each sub-reward in the step reward r_{step} is set to a negative value, and the closer the synergistic situation formed between the intelligences is to the ideal state, the closer the value of r_{step} converges to 0, which can guide the intelligences to update to a better synergistic strategy. When the task is completed, all the intelligences will get positive rewards.

2.2.3. Algorithm flow. In this algorithm, both top-level and bottom-level strategies use independent neural networks and experience buffer pools, and the bottom-level strategy is trained using sample $(s_t, m_t, a_t, r_t, s_{t+1}, d_t)$, where $r_t = r(s_t, m_t, a_t, s_{t+1})$ is the intrinsic reward.

The TD-target is first calculated based on the sample:

$$y_i = r_i + \gamma Q_i'(s_{i+1}', a_1', \ldots, a_t')\big|_{a_k' = \mu_k'(o_k)} \tag{3}$$

Next construct the mean square Bayesian error function. Update the critic network by minimizing the loss function:

$$\mathrm{L}(\theta_i) = \frac{1}{B}\sum\left(y_i - Q_i(s_t, a_1, \ldots, a_N)\right)^2 \tag{4}$$

Finally, the target network is updated. After the training of the bottom layer policy is completed, the top layer policy is trained using sample $\left(s_t, m_t, \sum R_{t:t+c-1}, s_{t+c}\right)$, where $\sum R_{t:t+c-1}$ is the sum of the external environmental rewards over c time steps. The concept of entropy constraint is added to the top layer strategy, for a random variable x, the entropy is defined as $H_x = \mathrm{E}_{x\sim p(x)}\left[-\log p(x)\right]$. The entropy of the strategy represents the degree of mixing of different actions that may be output by the strategy $\pi(a|s)$ for a certain state s. The entropy of the strategy is used as a constraint to maximize the entropy of the strategy while learning to maximize the experience, and then the TD-target of the top layer strategy at this time becomes:

$$y_i = r_i + \gamma \min_{j=1,2} Q_{\omega_j^-}(s_{i+1}', a_1', \ldots, a_i') - \alpha \log \pi_\theta(a_1', \ldots, a_i' | s_{i+1}')\big|_{a_k' = \pi_k'(o_k)} \tag{5}$$

Where an estimate of the entropy can be obtained by sampling computation a' and computing $-\log \pi(a'|s')$, the estimate of the Q function at the next state is estimated using the smaller of the two objective networks. Updates are performed using the same loss function for both critic networks ω_1 and ω_2:

$$\begin{cases} L(\omega_1) = \mathrm{E}_{(s,a,r,s')\sim D}\left[\left(Q_{\omega_1}(s,a) - y_t\right)^2\right] \\ L(\omega_2) = \mathrm{E}_{(s,a,r,s')\sim D}\left[\left(Q_{\omega_2}(s,a) - y_t\right)^2\right] \end{cases} \tag{6}$$

Sample action $\tilde{a}_i$ using the reparameterization trick and then update the top actor network with the policy gradient:

$$L_\pi(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(\alpha \log \pi_\theta\left(\tilde{a}_i \mid s_i\right) - \min_{j=1,2} Q_{\omega_j}\left(s_i, \tilde{a}_i\right)\right) \tag{7}$$

In this way, the top-level strategy can learn how to set appropriate subgoals, while the bottom-level strategy will learn how to accomplish the subgoals output by the top-level strategy, and this hierarchical strategy structure can help the intelligent body to accomplish the reinforcement learning task in complex scenes.

The flow of HMaAC algorithm is shown in Figure 4. Firstly, initialization operation is carried out to initialize the actor network and critical network weights, the parameters required for reinforcement learning, and the experience buffer pool, and at the same time, the simulation platform is connected to obtain the posture information, and the state space and action space of the intelligent body are initialized. After initialization, the algorithm iteration starts, the top-level intelligent body (HL_agent) obtains the global situation information and selects the current optimal task strategy according to the top-level algorithm and assigns it to the bottom-level intelligent body (LL_agent). LL_agent generates the intelligent body action set according to the top-level task and the current situation, and then sends this action set to the simulation environment for training and updating the state and the return value. LL_agent passes the bottom layer parameters to HL_agent and trains the top layer intelligent body after judging the completion of the subtask.

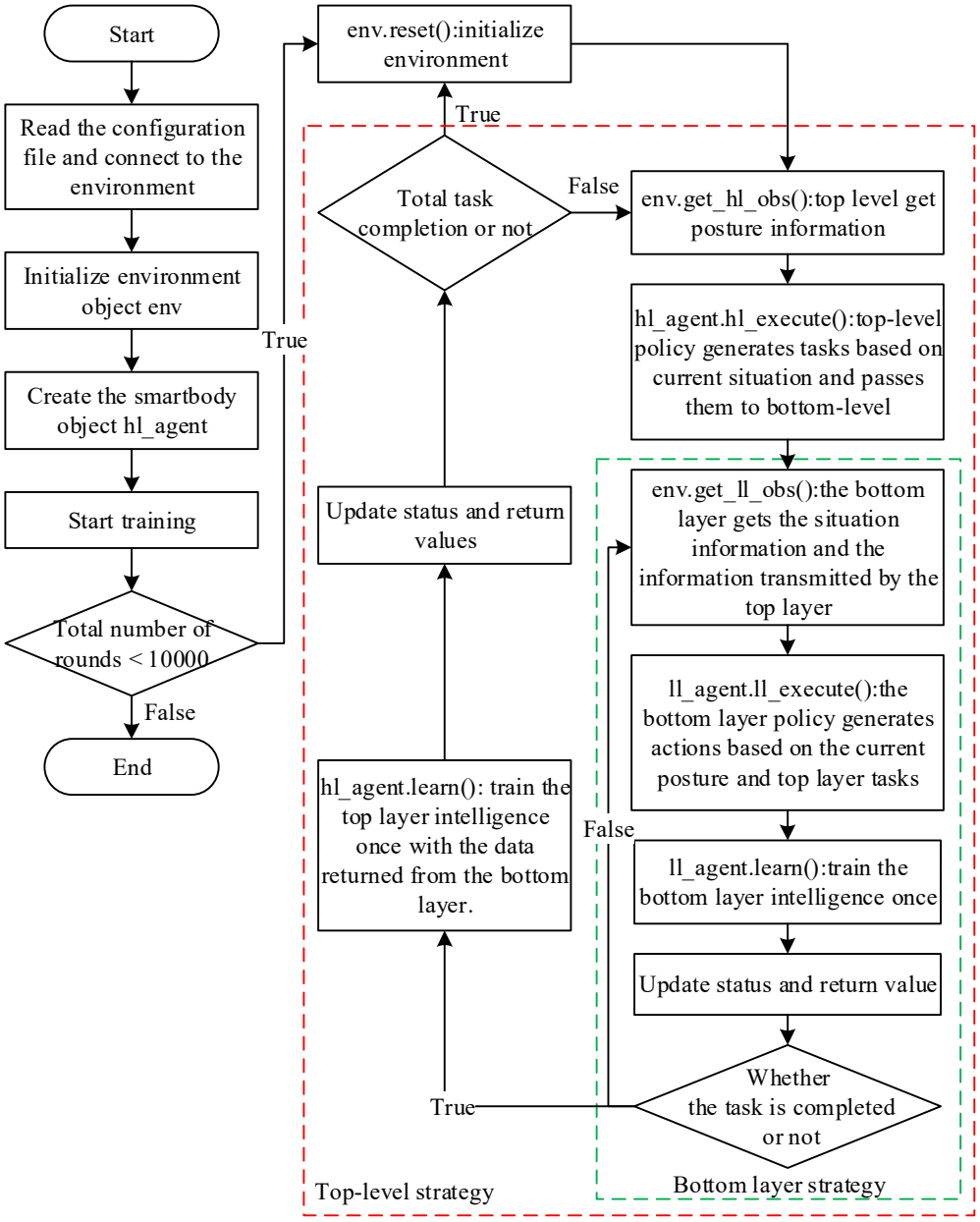


Fig. 4. HMaAC algorithm flow

3. Multi-intelligence collaborative algorithm based on role parameter sharing

For the multi-intelligent body global optimization problem, a hierarchical reinforcement learning coordination framework is given to coordinate the action selection among the intelligences in the multi-intelligent body system. However, as the complexity of the scene and the number of intelligences increase, the above method then has an upper and lower layer unsmoothness problem, for the upper and lower layer unsmoothness problem, the multi-intelligence role parameter sharing problem is further investigated on the basis of which a multi-intelligence reinforcement learning coordination algorithm based on role parameter sharing is proposed. Role division can make gains in partial parameter sharing, which forms a diversity of strategies through intrinsic rewards and reduces certain training time, yielding excellent performance and collaboration and making its multi-intelligence body coordinated control more stable.

3.1. Homogenization and intrinsic rewards

Another key challenge in multi-intelligent deep reinforcement learning is the upper and lower layer non-smoothness problem. The upper and lower layer unsmoothness problem is mainly related to the situation where the complexity and computational complexity of the software system increases as the number of intelligences increases. The increase in the number of intelligences leads to an exponential increase in the state space and action space of the software system as well, along with a dramatic increase in computational complexity.Exploration in deep reinforcement learning refers to the attempt of an intelligent body to try out a new action in an unknown environment in order to understand the environment. Exploitation refers to the intelligent body using known information to choose the optimal action to obtain the maximum reward.In deep reinforcement learning, the exploration-exploitation problem refers to how the agent makes a trade-off between exploration and exploitation to achieve the best learning effect. A common approach is to use intrinsic rewards as additional incentives to encourage the intelligences to engage in continuous exploration and generate diverse strategies. This section focuses on issues in homogenization and intrinsic reward in multi-intelligent deep reinforcement learning.

3.1.1. Homogenization. In order to apply multi-intelligent deep reinforcement learning algorithms to large-scale multi-intelligent and resource-constrained scenarios, it is necessary to deal with the upper and lower level unsteady problems, and most of the algorithms have high variance of gradient estimation and need to increase the computational cost and memory as the number of intelligences increases. Most algorithms generally take a parameter sharing approach in order to solve this problem. Parameter sharing uses only one parameterized function for deep neural networks for strategy or evaluation of all intelligences and improves sample efficiency through fast training. However the use of parameter sharing leads to the homogenization problem, for plain parameter sharing using a single common function, if the observed inputs to the intelligences are the same and the evaluation network estimates the same expected reward, even if the individual intelligences are rewarded individually they will choose the same action.

3.1.2. Intrinsic Rewards. In multi-intelligent body deep reinforcement learning, intrinsic reward refers to the reward signals generated by the intelligent body itself from the environment while performing a task, rather than the reward signals given externally. Different from extrinsic rewards, intrinsic rewards are generated through the intelligences' own behaviors and perceptions, which are usually related to the degree of task completion and learning progress. The introduction of intrinsic rewards can make multi-intelligentsia more autonomous and flexible in task execution. It can stimulate the intelligences to explore the environment on their own, thus enhancing learning and making the multi-intelligence upper and lower hierarchical instability problem better.

3.2. Algorithmic framework

In this section, the detailed design of R-PS, a multi-intelligentsia collaborative algorithm based on role parameter sharing, is presented, including the overall architecture of the algorithm and the connection between the various modules of the algorithm. The method firstly performs partial parameter sharing by dividing the roles into subgroups and dividing them into the same group of intelligences, and also introduces an intrinsic reward-based diversity policy learning mechanism, which also reduces the use of inefficient empirical sample data. This helps to improve the smooth performance of upper and lower layers and reduce the waste of training time and resources. The details are as follows:

3.2.1. Role Classification. In a multi-intelligent body setting, we assume that N intelligences can be partitioned into K roles, with partial parameter sharing by ground intelligences in the same role. A variational autoencoder (VAE) is a generative model that learns a density function on some

unobserved latent variables Z given input x. Given an unknown true posterior $p(z|x)$, the VAE approximates it with a parametric distribution $q_\psi(z|x)$ with parameter ψ. Calculate the KL scatter from the parameter distribution to the true posterior:

$$\begin{aligned} &D_{KL}\left(q_\psi(z|x)\|p(z|x)\right)=\log p(x) \\ &-\mathrm{E}_{z\sim q_\psi(z|x)}\left[\log p_u(x|z)\right]+D_{KL}\left(q_\psi(z|x)\|q(z)\right) \end{aligned} \tag{8}$$

where $\log p(x)$ is the logarithmic evidence as a constant term and the remaining two are negative evidence lower bounds ELBO, minimizing ELBO is equivalent to minimizing the KL scatter between the parameter and the true posterior.

Therefore, the parameter ψ of the parameterized Gaussian distribution is obtained by variational inference, i.e:

$$q_\psi(z|i)=N\left(\mu_\psi,\sigma_\psi;i\right) \tag{9}$$

A variational self-coding framework is used to optimize objective $D_{KL}\left(q_\psi(z|i)\|\ p(z|tr)\right)$. According to Eq. (8), the KL scatter is greater than 0, introducing a lower bound on the logarithmic evidence for $\log p(tr)$:

$$\log p(tr)\geq \mathrm{E}_{z\sim q_\psi(z|i)}\left[\log p_u(tr|z)\right]-D_{KL}\left(q_\psi(z|i)\|p(z)\right) \tag{10}$$

The ELBO reconstruction term is factorized as:

$$\begin{aligned} \log p_u\left(tr|z\right)&=\log p_u\left(r_t,o_{t+1}|a_t,o_t,z\right)p\left(a_t,o_t|z\right) \\ &=\log p_u\left(r_t|o_{t+1},a_t,o_t,z\right)+\log p_u\left(o_{t+1}|a_t,o_t,z\right)+c \end{aligned} \tag{11}$$

The last term $p\left(a_t,o_t|z\right)$ is discarded because a_t and o_t do not depend on the hidden variables z.

The encoder-one-decoder model is learned using the historical experience of all the intelligences and can represent the state transfer function $\hat{P}$ and reward function $\hat{R}$ of all the intelligences i, $i\in N$. The decoder f_e is utilized to generate the potential intent features from the id of the intelligences, and clustering is performed using K-means to classify the ones with high similarity of the potential intent features into the same roles, and intelligences belonging to the same roles are partially parameter-shared, and the dynamic role partitioning parameter sharing is realized in the After a certain time step, the roles are re-divided by the encoder to realize dynamic role division parameter sharing.

3.2.2. Diversity Strategy Learning. Encouraging strategy diversity through intrinsic rewards is a commonly used approach to promote exploration and learning in multi-intelligent body reinforcement learning.The loss function of VAE is divided into two terms, the reconstruction loss (RL) as well as the KL dispersion regularity term, which correspond to the two purposes that the model training process wishes to achieve, respectively. Namely:

$$\mathrm{L}_{vae}\left(\theta\right)=MSE\left(o,\hat{o}\right)+KL\left(N\left(\mu_\psi,\sigma_\psi^2\right),N(0,1)\right) \tag{12}$$

The reconstruction loss calculates the MSE loss between the vectors obtained from decoding by the decoder and the real observed vectors, i.e., it reflects the difference between the results generated by the VAE and the expectation, and is used to train the decoder to generate samples that approximate the real data. KL dispersion is used to constrain the distribution of hidden variables not to deviate too much from the prior distribution, and the standard normal distribution is usually used as the prior distribution in VAE, i.e., each latent variable z is independently and identically distributed $N(0,1)$ under the prior distribution.

3.2.3. Sharing of role parameters. In order to improve the representation ability of simple parameter sharing, Selective Parameter Sharing (SePS) is proposed.SePS divides all the intelligences into subgroups of shared parameters according to the clustering algorithm before the training, but once the subgroups are determined in this method, the objects of parameter sharing are no longer changed during the subsequent learning process of the intelligences, which is inflexible enough for the intelligences to learn, and it also causes a certain degree of homogeneity problem within the group. Therefore, this method proposes role-based partial parameter sharing, for K a role divided by the role division module, maintains K a Q function Q_R^S shared among the same roles, and at the same time, each intelligent body i maintains a local Q function Q_i^I of its own, i.e., by decomposing the Q values within the role, the intelligent bodies adaptively share the parameters and share the experience. The overall Q_i value of an intelligent body i belonging to role 1 can be expressed as:

$$Q_i\left(a_i \mid h_i\right)=Q_{R_1}^S\left(a_i \mid h_i\right)+Q_i^I\left(a_i \mid h_i\right) \tag{13}$$

Meanwhile, the local Q^I functions of the intelligences themselves take $L1$ regularization, which aims to enable the intelligences within a role to share parameters to share experiences as much as possible, and to avoid excessive diversification of the intelligences' strategies leading to the neglect of inter-task cooperation.

3.2.4. Overall optimization objectives. This method takes a QMIX approach to the decomposition of Q_{tot} and uses an end-to-end approach within this framework to update the network by minimizing the overall loss.The global environmental reward is used in QMIX, so the intrinsic reward is set to:

$$r^I=\mathrm{E}_{id}\left[dis\tan ce\left(o_i,\hat{o}_i\right)\right] \tag{14}$$

Combining the global environmental rewards r^e and r^I, the loss function is defined using a form of TD error:

$$\mathrm{L}_{TD}\left(\theta\right)=\left[r^e+\alpha r^I+\gamma\max_{a'}Q_{tot}\left(s',a';\theta^-\right)-Q_{tot}(s,a;\theta)\right]^2 \tag{15}$$

where α is a hyperparameter that adjusts the ratio of intrinsic to environmental rewards, and θ^- is a parameter that is periodically replicated and updated from θ. Combining Eqs. (12) and (15), the final overall optimization objective can be obtained:

$$\mathrm{L}\left(\theta\right)=\mathrm{L}_{TD}\left(\theta\right)+\mathrm{L}_{vae}\left(\theta\right)+\lambda\sum_z\mathrm{L}_{L_1}\left(Q_i^I\left(\theta_i^I\right)\right) \tag{16}$$

4. Empirical analysis of coordinated multi-intelligence control

4.1. Example Analysis of Multi-Intelligent Body Coordinated Control Strategy

This section describes two experiments in a goal-oriented, multi-intelligent body cooperative environment. One is a static goal scenario, a cooperative navigation task. The other is a dynamic target scenario, a multi-sensor multi-target coverage task. Comparative experiments are conducted for the two scenario tasks and the corresponding experimental results are analyzed.

4.1.1. Experimental environment. For the goal-oriented multi-intelligentsia cooperative task, two task scenarios are used, one is the cooperative navigation scenario with static goals, and the other is the multi-sensor multi-target coverage task scenario with dynamic goals. The cooperative navigation experiment is shown in Fig. 5, where n intelligences need to reach n landmarks. The landmarks are fixed and the intelligences can move around the scene, this task requires the intelligences to make autonomous decisions in an unknown environment by cooperating with other intelligences in order to achieve the goal of maximizing the overall reward.

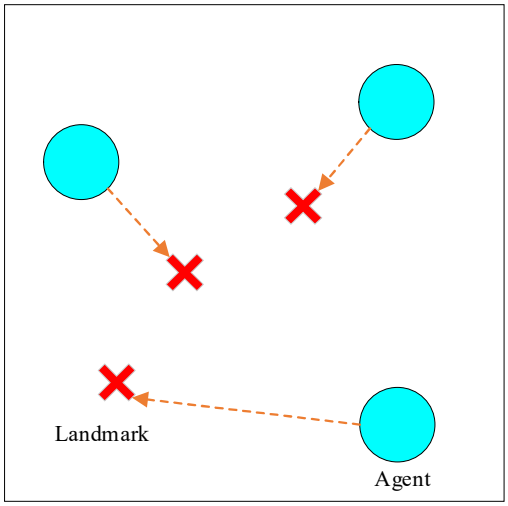


Fig. 5. Cooperative navigation scenario

Multi-sensor multi-target coverage experiment is a challenging task, the multi-sensor multi-target coverage scenario is shown in Fig. 6, n intelligences (sensors) need to work together to cover as many targets as possible. Each smart body has a certain monitorable range, which is represented by a dashed circular area. The direction in which the smart body is facing is the area being monitored is represented by the fan-shaped area, and the target is an orange dot that can move randomly in the scene. When a smart body monitors a target, it receives a corresponding reward, and the more targets it monitors, the greater the reward. However, due to the possibility of redundant targets between multiple intelligences, this may lead to a reduction in the overall monitoring range and potentially a reduction in monitored targets. Therefore, collaboration between intelligences is required to accomplish the task.

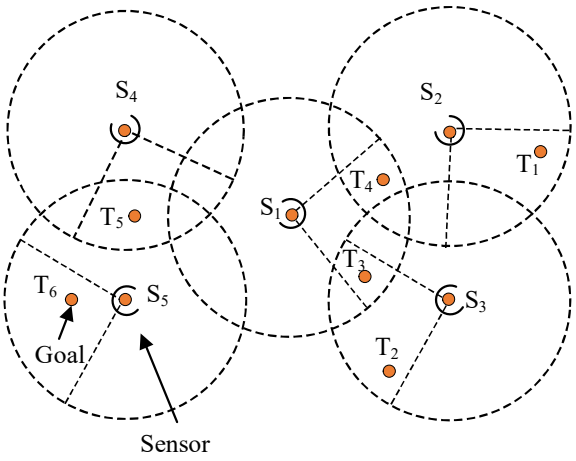


Fig. 6. Multi-sensor multi-target coverage scenario

4.1.2. Experimental setup. For the cooperative navigation experimental scenario, this paper sets the number of intelligences to be 6 and the number of landmarks to be 6. The landmarks are randomly initialized at their positions in the scenario but do not move, and the intelligences are randomly initialized in the scenario to keep exploring and moving and will be rewarded with a feedback of a negative value of the distance value from their nearest landmarks. At the same time, the intelligent body receives a -1 penalty if it collides in the experimental scene.

For the multi-sensor multi-target coverage experimental scenarios, this paper sets up different scenarios where the number of targets is 5 and the number of intelligibles ranges from 2 to 5. The

intelligences can take one action per time step (rotate 5° to the left, rotate 5° to the right, and keep still), and the rotation causes the intelligences to obtain an energy penalty of -0.1, which is for the intelligences to learn to minimize the energy loss while maximizing the reward.

In the cooperative navigation experimental scenario and the multi-sensor multi-target coverage experimental scenario, random seeds are used. The total step size of the two experiments is set to 5,000,000, and the maximum number of steps in an experimental round is 100. The learning rate is 0.0001 and the discount factor is 0.95. The experience replay buffer size for policy update is 10000.

4.1.3. Evaluation indicators. In order to verify the effectiveness of the method in this paper, different types of algorithms are selected for comparison, including HIT-MAC based on hierarchical, QMIX based on value decomposition, COMA based on counterfactual, MAPPO based on proximal policy optimization and independent IQL algorithm as baseline algorithms in the classical multi-intelligentsia collaboration scenarios of two kinds, static and dynamic goals, using the reward value as an evaluation index, and the final reward value to measure the performance of the algorithms.

4.1.4. Analysis of results. Figure 7 gives the reward mean values of IQL, COMA, QMIX, MAPPO, and this paper's algorithm after training under the cooperative navigation experimental task, and it can be seen that this paper's algorithm has a higher reward mean value than the other algorithms, and it plays a better performance in the task of landmarks covered by the intelligentsia, which is due to the coordinating role of this paper's algorithm for the process of cooperation between the intelligentsia.

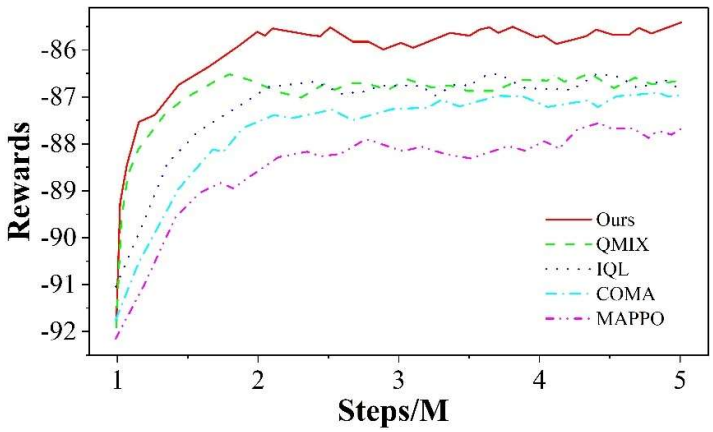


Fig. 7. Multi-agent cooperative navigation experiment different algorithm training

The algorithm in this paper is not only limited to cooperative navigation experiments with static targets, but also performs equally well in multi-sensor multi-target coverage experiments with dynamic targets. The experimental results, as shown in Fig. 8, show that this paper's algorithm obtains the highest global reward after 5 million time-steps of training in an environment with 4 intelligences and 5 targets. This shows that the high-level policy coordination effect of this paper's algorithm is also applicable in the multi-sensor multi-target coverage experiments and can improve the global reward.

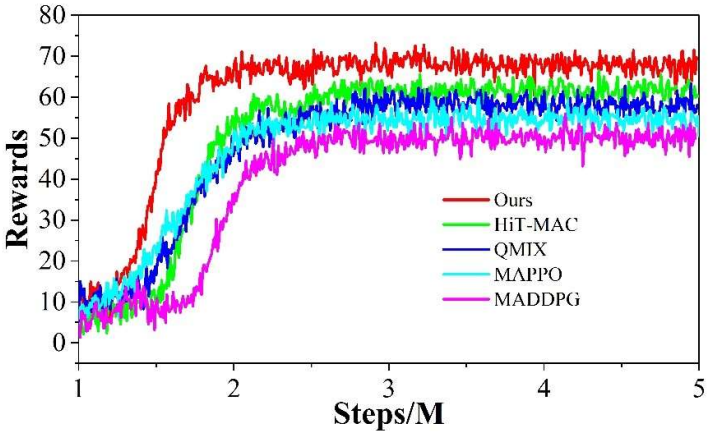


Fig. 8. Multi-objective coverage experiment with different algorithm training curves

The multi-sensor coverage experimental environment, shown in Fig. 9, explores the scalability and adaptability of this paper's algorithm to the number of intelligentsia in a multi-objective multi-sensor target coverage task. We conducted experiments with different number of intelligences to compare this paper's algorithm with other baseline algorithms and observe the performance of this paper's algorithm in different scenarios.

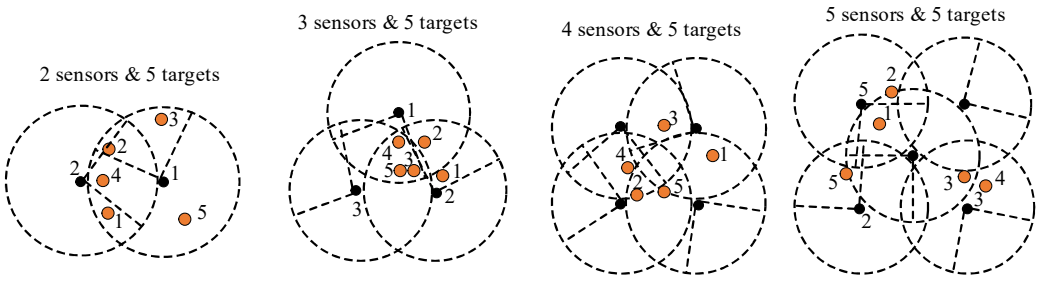


Fig. 9. Multi-sensor overlay experimental environment

The reward values of the algorithm for multi-sensor multi-target coverage experiments with different numbers of intelligences are shown in Fig. 10. The experimental results show that the algorithm of this paper shows good scalability and adaptability under different numbers of intelligent bodies. When the number of intelligent bodies increases by 5, the performance of this paper's algorithm does not show a significant downward trend, but maintains a more stable performance. This proves the scalability and adaptability of this paper's algorithm in multi-intelligent body collaboration tasks, which provides strong support for practical applications. The algorithm in this paper also shows excellent performance in multi-sensor multi-target coverage experiments, which further proves the effectiveness of the algorithm in coordinating intelligent bodies and improving global reward. The method in this paper is able to handle the feedback problem of high-level strategy reward in the case of a small number of intelligences, which ensures the stability of high-level strategy learning and provides effective coordination guidance for low-level strategies, thus improving the performance of the multi-intelligence coordination task.

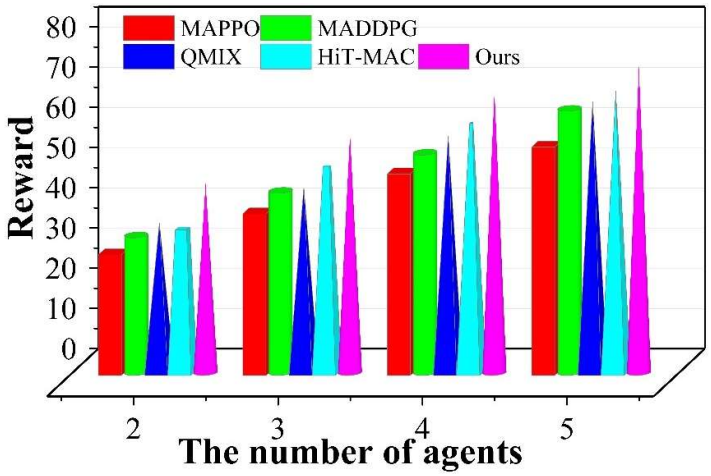


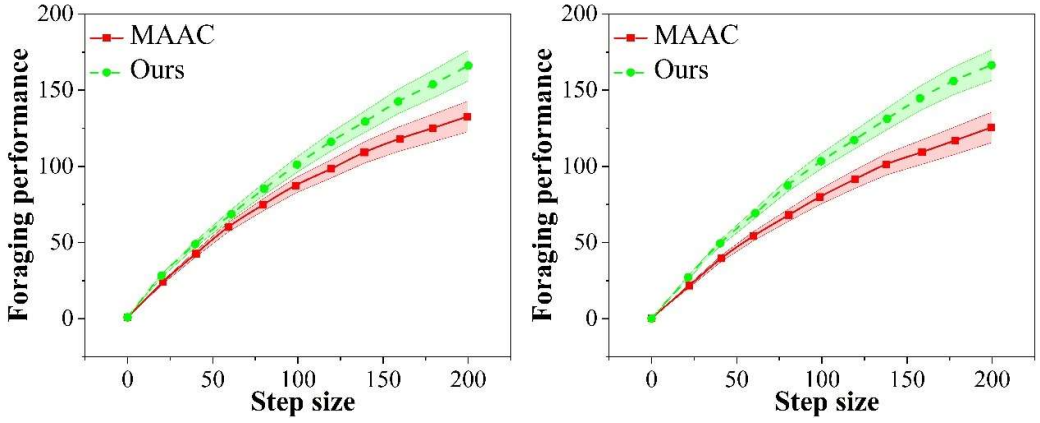
Fig. 10. Algorithm reward mean value under different quantity

4.2. Multi-Intelligence Collaboration Example Analysis

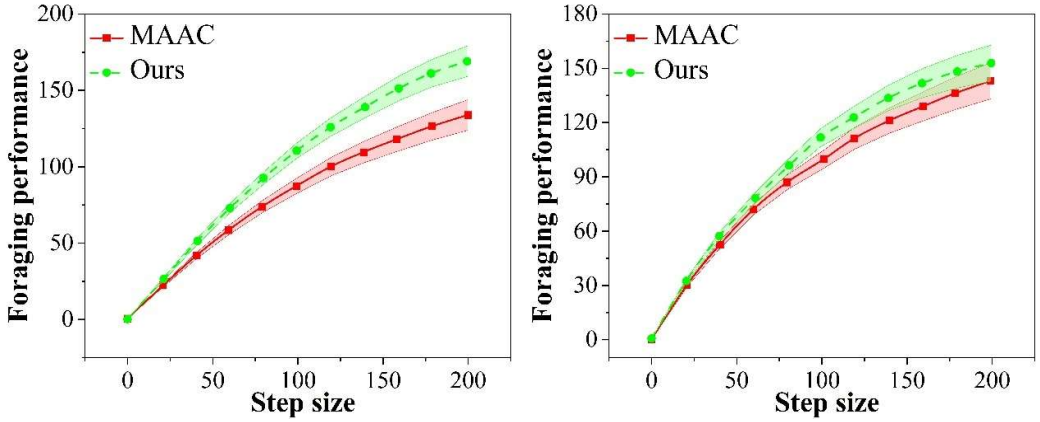
With the increase in the number of intelligences, the multiintelligence coordinated control method will have a nonsmooth problem, in this regard, the multiintelligence collaborative algorithm based on role parameter sharing is proposed. In order to evaluate the effectiveness of the multi-intelligent body cooperative algorithm based on role parameter sharing for the non-smooth problem between the high-level strategy and the low-level strategy, this section proves the prioritization of this paper's algorithm for the high bottom-level non-smooth problem in terms of algorithm performance, training time, and reward curve.

4.2.1. Comparative analysis of algorithm performance. Figure 11 demonstrates the analysis of the foraging performance of the number of intelligences during the expansion process, where (a) ~ (i) are 7~15 intelligences, respectively. The experiment was conducted on five random seeds, and the data presented is the average of the results of one hundred runs of twenty rounds under each of the five models, with the standard deviation being the corresponding shaded area on the graph. The intelligent body interacts with the environment for 200 steps in a round, and the current foraging progress of the intelligent body is calculated every 100 steps, and the algorithm is iterated a total of 20,000 times during the training process. Comparing the foraging efficiencies of the two methods in scenarios with different number of target points objectives, it can be seen that the multi-intelligent body collaborative algorithm R-PS (red) trains a network that can accomplish the task better, compared to the MAAC method (green) that gets more target points in all the settings of the number of intelligent bodies. Moreover, as the number of steps in the round progresses, the intelligent bodies are more able to accurately find the target point to pick up when the distribution of target points becomes sparse in the scene, and take the target point at a steady rate to complete the foraging task. The MAAC method, on the other hand, decreases the slope of the curve with the passage of time, and the rate of successful foraging gradually decreases, which differentiates it from the multi-intelligentsia collaborative algorithm R-PS. With the expansion of the number of intelligences, the R-PS method can still maintain the stability of the model. Even with a large number of intelligences in the scene, after one round, intelligences using the R-PS method are more dominant in the maintenance of foraging ability (147.3, 171.4, 171.9, 168.9, 150.8, 158.6, 133.9, 151.9, 126.6, 150.7), and can still successfully place even when it has been expanded to 15 intelligences 151.4 (value average) intelligences, indicating that the method in this paper can effectively solve the non-smooth problem validity between high-level and low-level strategies. While the MAAC method shows a gradual decrease in the model foraging ability during the expansion process (133.8, 137.5, 129.7, 134.6, 141.5, 135.9, 122.7, 122.7, 113.7, 94.4), when 15 intelligences forage in the scene at the same time, due to the presence of too many intelligences in the

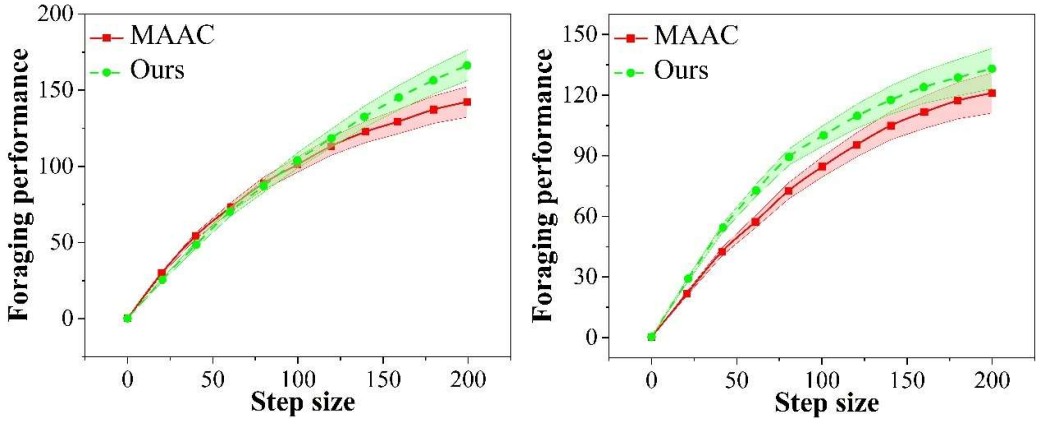
field, the rising interactions lead to the generation of their strategies more influenced by the rest of the intelligences, the strategy variance between intelligences is high, and the number of successes it has in placing target points drops to 94.4.



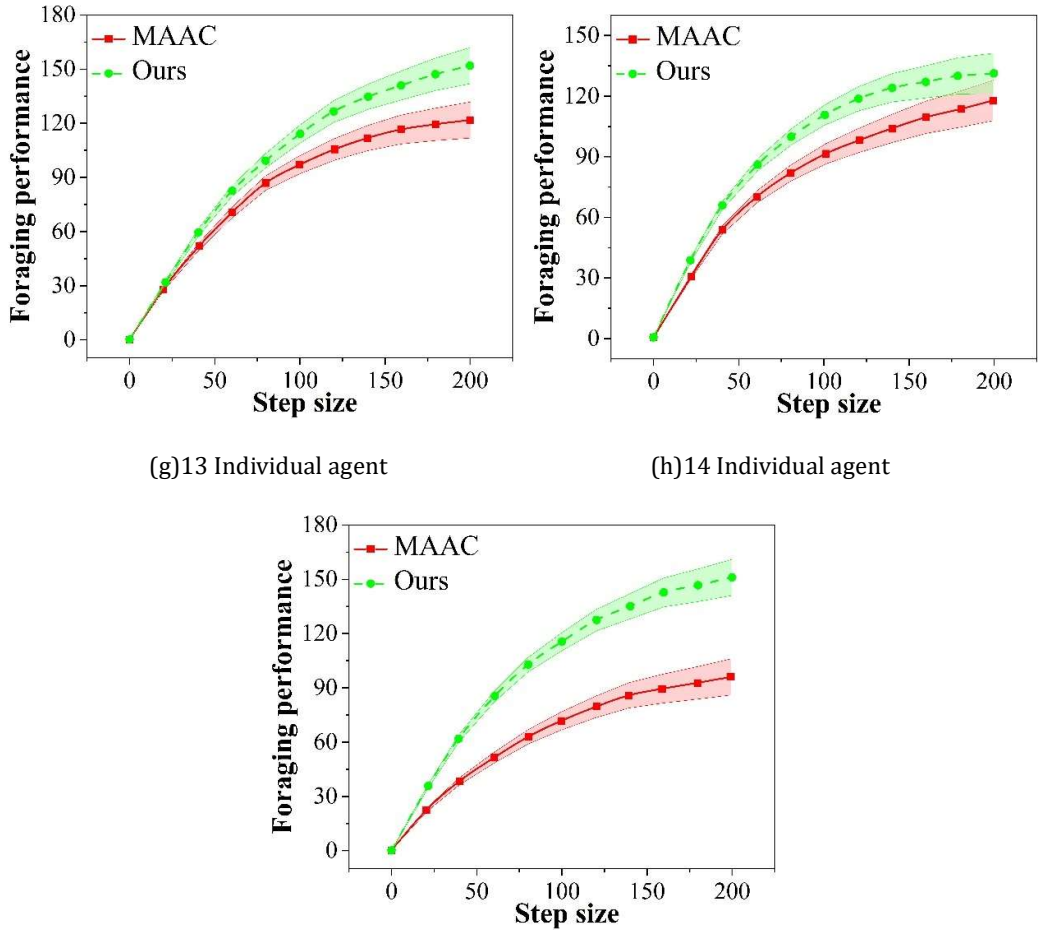
(a)7 Individual agent (b)8 Individual agent



(c)9 Individual agent (d)10 Individual agent



(e)11 Individual agent (f)12 Individual agent



(g)13 Individual agent (h)14 Individual agent

(i)15 Individual agent

Fig. 11. Comparison of foraging performance under different agent number scenarios

4.2.2. Comparative analysis of training time. The policy network with shared role parameters is used, which can effectively reduce the training time by reducing the network parameters that need to be updated during the training process, while guaranteeing the training effect, so that the algorithm can be more capable of facing the upper and lower layer stability problems due to the expansion of the number of intelligences, and the results of the training time comparison analysis are shown in Fig. 12. The model is averaged over 5 random seeds. From left to right, the time needed to update the policy network for 35,000 rounds of algorithm iterations is shown for both methods as the number of intelligences in the scenario expands, taking the last two digits of seconds. The method in this paper allows multiple intelligences to be divided into multiple roles at each training, which effectively reduces the total number of updates to the network, especially as the number of intelligences continues to expand and the total policy network of intelligences continues to grow. The total number of layers of the network increases with the number of intelligences, and this increase consists of two main aspects; first, the Actor-Critic network maintains an encoder for each intelligence to process the intelligence's observations and an encoder for generating values, and for each increase in the number of intelligences, each of these modules constructs another corresponding network, with an increase in the number of layers of the network.15 Second, the number of layers of the strategy network, for each increase in the number of intelligent body, the number of layers rises by 4. The method in this paper makes no changes to the Actor-Critic network, and only classifies the intelligent bodies into multiple roles, which improves the stability of the upper and lower layers. So it can also be seen from the experimental results that the time used by both methods increases as the number of intelligences rises,

but with the use of this paper's method 20,000 iterations can be completed in less time with a smoother rise.

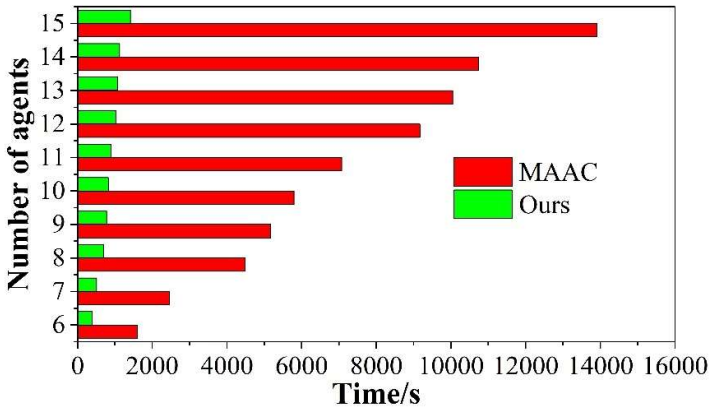
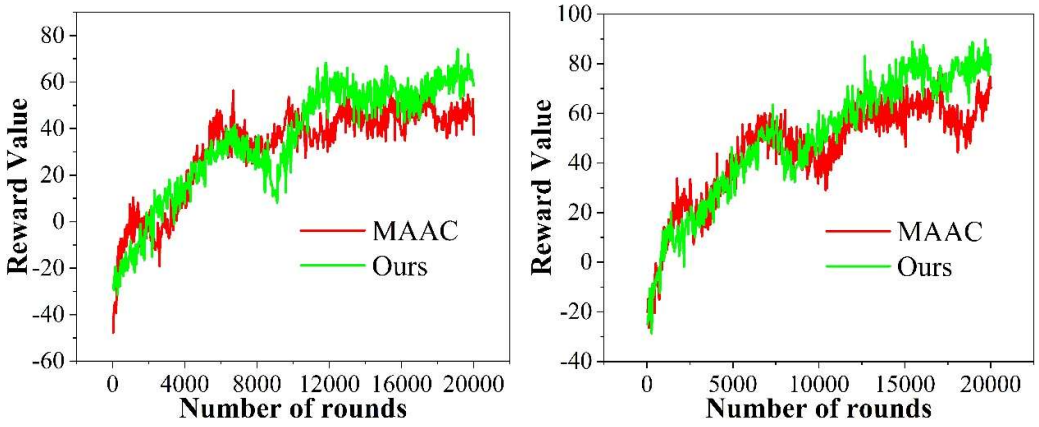
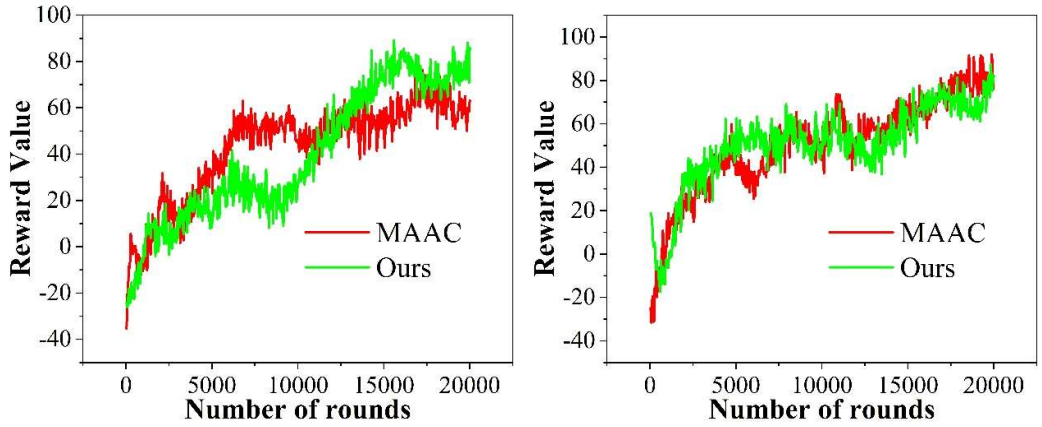


Fig. 12. Training time comparison and analysis results

4.2.3. Comparative analysis of incentive curves. Figure 13 shows the reward curves of MAAC and the method in this paper under different settings of the number of intelligences, where (a) to (d) are 6 intelligences, 8 intelligences, 10 intelligences, and 12 intelligences, respectively, and the experiments are run in a foraging scenario for 20,000 rounds, with a single intelligence interacting with the environment 100 times per round. Every 200 steps, the intelligences will interact with the environment without exploration and get the reward value of the feedback as the evaluation criterion of the current coordinated control strategy. It can be seen that in most cases, the role parameter sharing algorithm can help the network reach a more optimal strategy faster, with a steeper curve, and better avoid the occurrence of upper and lower level instability. Although in some tasks, the reward value of the intelligences did not rise as fast as that of the benchmark algorithm at the beginning, which may be due to the fact that the intelligences were not greedy to obtain higher values at the beginning, but were more concerned about exploring into different state spaces, and eventually the reward value of the intelligences based on the role parameter sharing algorithm stabilized at a level higher than the reward value of the MAAC method.



(a)6 Individual agent (b)8 Individual agent



(c)10 Individual agent (d)12 Individual agent

Fig. 13. The reward curve under different agent number Settings

5. Conclusion

This paper focuses on the research of multi-intelligent body coordination and control strategy under the software system architecture, which gives the hierarchical multi-intelligent body coordination and control strategy to coordinate the action selection among the intelligences in the software system, and to improve the overall performance and efficiency of the software system in order to achieve the purpose of collaborating to accomplish a specific task. With the increase of the number of intelligences, the hierarchical multi-intelligence coordinated control strategy will have the problem of upper and lower level instability, for this type of problem, on the basis of the original, the multi-intelligence collaborative algorithm based on the sharing of role parameters is proposed, so as to make the multi-intelligence coordinated control under the framework of the software system more stable, and at the same time also to accelerate the convergence performance of the multi-intelligence deep reinforcement learning. Combining the above theories, the research scheme of this paper is empirically analyzed. The algorithm in this paper has a high reward value (40~80) in the two experimental environments, which is better compared to the other four algorithms, indicating that the algorithm still has a good coordinated control performance in complex environmental scenarios. Even as the number of agents increases, The multi-agent collaborative algorithm based on role parameter sharing still has good foraging ability (147.3, 171.4, 171.9, 168.9, 150.8, 158.6, 133.9, 151.9, 126.6, 150.7), which proves that the method can effectively deal with the upper-level non-stationary problems caused by the increase in the number of multi-intelligences, and ensures the coordinated control performance of multi-agents under the software system architecture, which has reference value for the development of artificial intelligence technology in the future.

Funding

National Natural Science Foundation of China (51709195): Research on Multi-objective Water Resources Optimal Allocation Model and Its Application in Semi-humid Drought-prone Irrigation Areas Under Uncertain Conditions.

References

[1] Gasparetto, A., & Scalera, L. (2019). A brief history of industrial robotics in the 20th century. Advances in Historical Studies, 8, 24-35.

[2] Bodenhagen, L., Suvei, S. D., Juel, W. K., Brander, E., & Krüger, N. (2019). Robot technology for future welfare: meeting upcoming societal challenges–an outlook with offset in the development in Scandinavia. Health and Technology, 9, 197-218.

[3] Dzedzickis, A., Subačiūtė-Žemaitienė, J., Šutinys, E., Samukaitė-Bubnienė, U., & Bučinskas, V. (2021). Advanced applications of industrial robotics: New trends and possibilities. Applied Sciences, 12(1), 135.

[4] Xie, J., & Liu, C. C. (2017). Multi-agent systems and their applications. Journal of International Council on Electrical Engineering, 7(1), 188-197.

[5] Gorodetsky, V. I., Kozhevnikov, S. S., Novichkov, D., & Skobelev, P. O. (2019). The framework for designing autonomous cyber-physical multi-agent systems for adaptive resource management. In Industrial Applications of Holonic and Multi-Agent Systems: 9th International Conference, HoloMAS 2019, Linz, Austria, August 26–29, 2019, Proceedings 9 (pp. 52-64). Springer International Publishing.

[6] Herrera, M., Pérez-Hernández, M., Kumar Parlikad, A., & Izquierdo, J. (2020). Multi-agent systems and complex networks: Review and applications in systems engineering. Processes, 8(3), 312.

[7] Li, Y., & Tan, C. (2019). A survey of the consensus for multi-agent systems. Systems Science & Control Engineering, 7(1), 468-482.

[8] Yi-Chen, L. (2020, December). Simulation Research on Consistency of Multi-Agent Systems with Different Topologies and Time-Delay. In 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE) (pp. 767-772). IEEE.

[9] Saim, M., Ghapani, S., Ren, W., Munawar, K., & Al-Saggaf, U. M. (2017). Distributed average tracking in multi-agent coordination: Extensions and experiments. IEEE Systems Journal, 12(3), 2428-2436.

[10] Terán, J., Aguilar, J., & Cerrada, M. (2017). Integration in industrial automation based on multi-agent systems using cultural algorithms for optimizing the coordination mechanisms. Computers in Industry, 91, 11-23.

[11] Scrimieri, D., Afazov, S. M., & Ratchev, S. M. (2021). Design of a self-learning multi-agent framework for the adaptation of modular production systems. The International Journal of Advanced Manufacturing Technology, 115(5), 1745-1761.

[12] Attajer, A., & Mecheri, B. (2024). Multi-Agent Simulation Approach for Modular Integrated Construction Supply Chain. Applied Sciences, 14(12), 5286.

[13] Davoodi, M., Faryadi, S., & Velni, J. M. (2021). A graph theoretic-based approach for deploying heterogeneous multi-agent systems with application in precision agriculture. Journal of intelligent & robotic systems, 101, 1-15.

[14] Gorodetsky, V., Skobelev, P., & Marik, V. (2020). System engineering view on multi-agent technology for industrial applications: Barriers and prospects. Cybernetics and Physics, 9(1), 13-30.

[15] Liu, C., Sun, S., Tao, C., Shou, Y., & Xu, B. (2021). Sliding mode control of multi-agent system with application to UAV air combat. Computers & Electrical Engineering, 96, 107491.

[16] Wiedemann, T., Manss, C., & Shutin, D. (2018). Multi-agent exploration of spatial dynamical processes under sparsity constraints. Autonomous Agents and Multi-Agent Systems, 32, 134-162.

[17] Lamamuri, P. S., Singh, J., Karwa, Y., & Nayak, S. K. (2024, April). Implementation and Optimization of Swarm Based System for Multi-agent Coordination and Task Execution in Marine Environment. In International Conference on Machine Intelligence, Tools, and Applications (pp. 91-101). Cham: Springer Nature Switzerland.

[18] Laport, F., Serrano, E., & Bajo, J. (2020). A multi-agent architecture for mobile sensing systems. Journal of Ambient Intelligence and Humanized Computing, 11(11), 4439-4451.

[19] Dong, S., Liu, M., Dong, S., Zheng, R., & Wei, P. (2024). Hierarchical Heterogeneous Multi-Agent Cross-Domain Search Method Based on Deep Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems.

[20] Ricardo, A. R., Benítez, I. F., González, G., Nuñez, J. R., & Llosas, Y. (2022). Multi-agent system for steel manufacturing process. International Journal of Electrical and Computer Engineering (IJECE), 12(3), 2441-2453.

[21] Li, S., Zhang, J., Li, X., Wang, F., Luo, X., & Guan, X. (2017). Formation control of heterogeneous discrete-time nonlinear multi-agent systems with uncertainties. IEEE Transactions on Industrial Electronics, 64(6), 4730-4740.

[22] Ismail, Z. H., Sariff, N., & Hurtado, E. G. (2018). A survey and analysis of cooperative multi-agent robot systems: challenges and directions. Applications of Mobile Robots, 5, 8-14.

[23] Benaboud, R., & Marir, T. (2020). Flexibility measurement model of multi-agent systems. Multiagent and Grid Systems, 16(3), 309-341.

[24] Bao, G., Ma, L., & Yi, X. (2022). Recent advances on cooperative control of heterogeneous multi-agent systems subject to constraints: A survey. Systems Science & Control Engineering, 10(1), 539-551.

[25] Hu, G., Zhu, Y., Zhao, D., Zhao, M., & Hao, J. (2021). Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning. IEEE Transactions on Neural Networks and Learning Systems, 34(8), 3966-3978.

[26] Cao, L., Chi, C., Yumuk, E., Ionescu, C. M., & Liu, G. P. (2023, July). Coordinated control of distributed multi-agent systems using event-triggered predictive control strategy. In 2023 2nd Conference on Fully Actuated System Theory and Applications (CFASTA) (pp. 378-383). IEEE.

[27] Wang, J., Hong, Y., Wang, J., Xu, J., Tang, Y., Han, Q. L., & Kurths, J. (2022). Cooperative and competitive multi-agent systems: From optimization to games. IEEE/CAA Journal of Automatica Sinica, 9(5), 763-783.

[28] Zhang, N., Wang, J., Li, Z., Li, S., & Ding, H. (2022). Multi-agent-based coordinated control of ABS and AFS for distributed drive electric vehicles. Energies, 15(5), 1919.

[29] Yuan, C., Lin, Y., Shen, J., Chen, L., Cai, Y., He, Y., ... & Yu, Q. (2023). Research on active collision avoidance and hysteresis reduction of intelligent vehicle based on multi-agent coordinated control system. World Electric Vehicle Journal, 14(1), 16.

[30] Li, J., Yu, T., & Yang, B. (2021). Coordinated control of gas supply system in PEMFC based on multi-agent deep reinforcement learning. International Journal of Hydrogen Energy, 46(68), 33899-33914.

[31] Wang, X., Zhou, J., Qin, B., & Guo, L. (2023). Coordinated control of wind turbine and hybrid energy storage system based on multi-agent deep reinforcement learning for wind power smoothing. Journal of Energy Storage, 57, 106297.

[32] Zhang, G., Hu, W., Cao, D., Zhang, Z., Huang, Q., Chen, Z., & Blaabjerg, F. (2022). A multi-agent deep reinforcement learning approach enabled distributed energy management schedule for the coordinate control of multi-energy hub with gas, electricity, and freshwater. Energy Conversion and Management, 255, 115340.

[33] Jing, Y., Zhiqiang, D., Jiemai, G., Siyuan, C., Bing, Z., & Yajie, W. (2022). Coordinated control strategy of electricity-heat-gas integrated energy system considering renewable energy uncertainty and multi-agent mixed game. Frontiers in Energy Research, 10, 943213.

[34] Wang, Q., Wang, Y., Chen, Z., & Soares, J. (2024). Multi-agent system consistency-based cooperative scheduling strategy of regional integrated energy system. Energy, 295, 130904.

[35] Rahman, M. S., & Oo, A. M. T. (2017). Distributed multi-agent based coordinated power management and control strategy for microgrids with distributed energy resources. Energy conversion and management, 139, 20-32.

[36] Wang, X., Su, H., & Jiang, G. P. (2021). Interval observer-based robust coordination control of multi-agent

systems over directed networks. IEEE Transactions on Circuits and Systems I: Regular Papers, 68(12), 5145-5155.

[37] Li, C., & Zong, X. (2022). Group hybrid coordination control of multi-agent systems with time-delays and additive noises. IEEE/CAA Journal of Automatica Sinica, 10(3), 737-748.

[38] Du, W., & Ding, S. (2021). A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. Artificial Intelligence Review, 54(5), 3215-3238.

[39] Jaiswal, M. (2019). Software architecture and software design. International Research Journal of Engineering and Technology (IRJET) e-ISSN, 2395-0056.

[40] Hua Xu,Juntai Tao,Lingxiang Huang,Chenjie Zhang & Jianlu Zheng. (2025). A Deep Reinforcement Advantage Actor-Critic-Based Co-Evolution Algorithm for Energy-Aware Distributed Heterogeneous Flexible Job Shop Scheduling. Processes,13(1),95-95.

[41] Xin Wen,Xianping Guo & Li Xia. (2024). Finite horizon partially observable semi-Markov decision processes under risk probability criteria. Operations Research Letters,57,107187-107187.