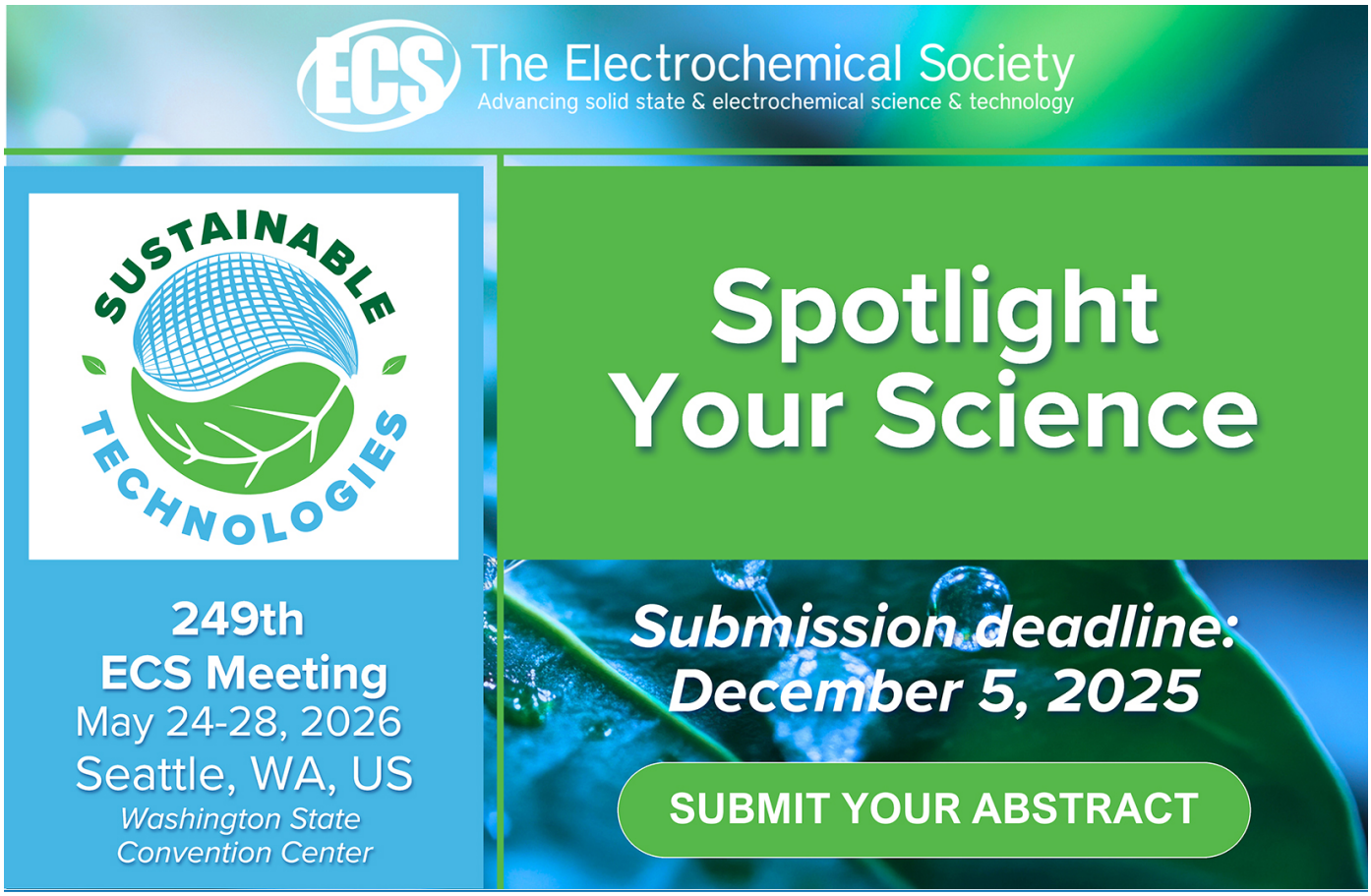
Deep Reinforcement Learning-based Collaborative Multi-UAV Coverage Path Planning



Journal of Physics: Conference Series **2833** (2024) 012017 doi:10.1088/1742-6596/2833/1/012017

Deep Reinforcement Learning-based Collaborative Multi-UAV Coverage Path Planning

Boquan Zhang[1,⋆,a], Tian Jing[1,b], Xiang Lin[1,c], Yanru Cui[1,d], Yifan Zhu[1,e], and Zhi Zhu[1,f]

[1]College of Systems Engineering, National University of Defense Technology, Changsha City, Hunan Province, China

E-mail: [⋆,a]zhangboquan18@alumni.nudt.edu.cn; [b]jingtian16@nudt.edu.cn; [c]linxiang@alumni.nudt.edu.cn; [d]cuiyanru@nudt.edu.cn; [e]yfzhu@nudt.edu.cn; [f]zhuzhi@nudt.edu.cn

Abstract. The coverage path planning problem has gained significant attention in research due to its wide applicability and practical value in various fields such as logistics and distribution, smart homes, and unmanned vehicles. This paper focuses on studying the coverage path planning problem under multi-UAV collaboration to maximize the coverage of the mission area within a given time. To address this problem, we propose a multi-objective optimization model and reformulate it with the framework of Decentralized Partially Observable Markov Decision Process (Dec-POMDP). We then employ a multi-agent deep reinforcement learning (MADRL) method to solve the problem. Specifically, we introduce the $\varepsilon-$Multi-Agent Twin Delayed Deep Deterministic Policy Gradient ($\varepsilon-$MADT3), which incorporates an exploration coefficient based on MATD3. This coefficient gradually decays with the number of iterations, allowing for a balance between exploration and exploitation. Numerous simulation results demonstrate that $\varepsilon-$MADT3 outperforms the baseline algorithm in terms of coverage rate and number of collisions.

1. Introduction

Coverage Path Planning (CPP) is a fundamental research issue in robotics and has extensive applications in areas such as unmanned patrols [1], search and rescue operations [2], and agricultural irrigation [3]. With advancements in artificial intelligence (AI) and communication technology, the utilization of Unmanned Aerial Vehicles (UAVs) for CPP is growing in popularity [4]. However, when faced with complex and dynamic mission environments, it has proven challenging for a UAV to autonomously complete the task [5]. As a result, employing multiple UAVs to collaborate on the coverage task has emerged as a viable solution. Multi-UAV systems possess advantages such as robustness, flexibility, and functional complementarity, which enhance the effectiveness of a single UAV. However, the autonomous coordination of multiple UAVs in planning their paths remains a challenging issue. Firstly, when multiple UAVs are simultaneously engaged in coverage tasks, it is crucial to ensure that their paths do not intersect or collide. Additionally, the coverage area may feature intricate terrain, obstacles, or restricted zones, necessitating real-time online planning of UAV coverage paths.

Recently, the advent of machine learning techniques, notably Reinforcement Learning (RL), has introduced innovative methods for solving cooperative CPP challenges with multiple UAVs.

In RL, an agent selects actions based on the environment's observed state and evaluates these actions using rewards provided by the environment [6]. Through continuous interaction, the agent learns from trial and error to optimize its action-selection strategy and maximize cumulative rewards. Deep Reinforcement Learning (DRL) [7] merges RL with Deep Learning, utilizing deep neural networks to approximate functions, thereby managing complex state spaces more effectively. Multi-agent deep reinforcement learning (MADRL) facilitates the online learning of multiple UAVs to coordinate in complex and dynamic environments. Therefore, further research should explore the application of MADRL techniques to tackle the cooperative coverage path planning challenge in multi-UAV systems.

2. Modeling and problem formulation

2.1. UAV dynamics model

The UAV model comprises M homogeneous UAVs in a distributed control model. The system of multiple UAVs is represented by $W = \{w_1, w_2, \cdots, w_M\}$. At time t, the position of UAV i is represented by $w_{i,t}$, and its velocity is denoted as $v_{i,t}$. Besides, the speed of the each UAV is controlled between $v_{\min}$ and $v_{\max}$. Therefore, the UAV dynamics model can be described as

$$\begin{cases} v_{i,t} = \frac{\text{clip}(\|v_{i,t-1}+a_{i,t-1}\Delta t\|_2, v_{\min}, v_{\max})}{\|v_{i,t-1}+a_{i,t-1}\Delta t\|_2}(v_{i,t-1} + a_{i,t-1}\Delta t) \\ w_{i,t} = w_{i,t-1} + v_{i,t-1}\Delta t \end{cases} \tag{1}$$

where $a_{i,t}$ is the input acceleration of UAV i, while Δt is the time increment used for updating the UAV's velocity and location.

As the UAV's motion is continuous, it is necessary to map the UAV's coordinates to the corresponding cell:

$$c_{i,t} = \text{round}(\frac{w_{i,t}}{L}) \tag{2}$$

2.2. Environment model

Assuming that the UAV remains at a constant altitude, its movement is considered to occur within a surface. The task area is divided into $L_1 \times L_2$ cells using the raster method, each with a side length of L. Coverage for the cell at the x-th row and y-th column within the mission area is represented by $m_{x,y}$. Here, $m_{x,y} = 1$ represents a covered cell, while $m_{x,y} = 0$ represents an uncovered cell. At time t, each UAV has a distributed coverage map $M_{i,t}$, denoted as

$$M_{i,t} = \begin{pmatrix} m_{1,1}^{i,t} & \cdots & m_{1,L_2}^{i,t} \\ \vdots & \ddots & \vdots \\ m_{L_1,1}^{i,t} & \cdots & m_{L_1,L_2}^{i,t} \end{pmatrix} \tag{3}$$

In the distributed multi-UAV system, each UAV shares its coverage map with neighboring UAVs for information fusion. The neighbors of UAV i are denoted as

$$N_i = \{W_j | \, \|w_{j,t} - w_{i,t}\| < R_{com}, \forall j \neq i\} \tag{4}$$

where R_{com} is the maximum communication distance between UAVs. At time t, the UAV's coverage map is integrated into a new coverage map through information interaction:

$$m_{x,y}^{i,t} = \max(m_{x,y}^{i,t}, \max_{W_j \in N_i}\{m_{x,y}^{j,t}\}) \tag{5}$$

The mission area includes N randomly distributed circular no-fly areas, each with an impact scope of R_a. The location of the no-fly area is represented by n_i. To accommodate the UAV's boundary constraints, we establish a buffer region along the mission area's edge, which contains no information.

2.3. Problem Formulation

In the coverage path planning task, each UAV departs from a unique initial position with the objective of covering the mission area within a specified time to enhance the coverage rate. Additionally, they must avoid no-fly areas and prevent collisions between UAVs. Consequently, the objective function and constraints can be defined.

1) Objective function:

$$f = \sum_{x=1}^{L_1} \sum_{y=1}^{L_2} m_{x,y}^{T} \tag{6}$$

The objective function f is maximizing the coverage of the task area, and T represents the maximum duration allowed for completing the CPP task.

2) Constraints:

The UAV flies within the mission area, then the boundary constraint can be expressed as follows:

$$\begin{cases} 0 \leq w_{i,t}(x) \leq L_1 \\ 0 \leq w_{i,t}(y) \leq L_2 \end{cases} \tag{7}$$

Furthermore, the UAVs must navigate around the no-fly area and maintain a safe distance from each other. The anti-collision constraint is expressed as follows:

$$\|n_k - w_{i,t}\| > R_a, \forall k \in \{1, 2, \cdots, N\} \tag{8}$$

$$\|w_{j,t} - w_{i,t}\| > R_{\text{safe}}, \forall i, j \in W, j \neq i \tag{9}$$

where R_{safe} is the safe distance between UAVs for collision prevention.

Thus, the problem of CPP for multi-UAV is formulated as

$$\begin{cases} \max & \sum\limits_{x=1}^{L_1} \sum\limits_{y=1}^{L_2} m_{x,y}^{T} \\ \text{s.t.} & \|n_k - w_{i,t}\| > R_a, \forall k \in \{1, 2, \cdots, N\} \\ & \|w_{i,t} - w_{j,t}\| > R_{safe}, \forall i, j \in W, j \neq i. \\ & 0 \leq w_{i,t}(x) \leq L_1 \\ & 0 \leq w_{i,t}(y) \leq L_2 \end{cases} \tag{10}$$

3. Reinforcement learning

3.1. Dec-POMDP

UAVs typically cannot directly observe the entire environment and can only infer its state from partial information. Therefore, Eq. 10 can be reformulated utilizing the Partially Observable Markov Decision Process (POMDP). In the issue of coverage path planning with multi-UAV, the Decentralized POMDP (Dec-POMDP) [8], which is an extension of POMDP, is defined as a tuple $(N, A, S, O, T_s, T_o, R, \gamma)$. In the tuple, N is the total number of agents, A represents the set of joint actions, S represents the finite set of environment states, O represents the set of joint observations, R represents the immediate reward, T_s represents the transfer probability function, T_o represents the observation probability function, and γ represents the discount factor for the reward.

The objective of MADRL is to learn a joint policy that maximizes the expected cumulative discounted rewards over time:

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{n=0}^{\infty} \gamma^n r_{t+n} \tag{11}$$

Q function represents the expected cumulative discounted rewards obtained by taking the joint action a_t under the joint observation o_t, which can be denoted as

$$Q_\pi(o, a) = E(G_t|o_t = o, a_t = a) \tag{12}$$

The Actor-Critic framework is widely employed by many MADRL algorithms, including COMA [9], MADDPG [10], and MATD3 [11]. This framework is used for centralized training and decentralized execution. In the centralized training phase, every agent receives additional global observations. In contrast, during the decentralized execution phase, agents only relies solely on their own observations to choose the actions to be taken. In the CPP problem with multi-UAV, the actor network for each UAV (with parameter θ_i) takes local observations o_i as inputs, and outputs a deterministic accelerate a_i. The parameter θ_i is updated through the stochastic gradient ascent method:

$$\nabla_{\theta_i} J(\mu_i) = E_{a,o\sim D}[\nabla_{\theta_i}\mu_i(a_i|o_i)\nabla_{a_i}Q_i(o, a)] \tag{13}$$

where $J(\mu_i)$ represents the goal of each agent, specifically the discounted expected return. Q_i denotes the Q function of the i-th agent. For training the actor network, a critic network with parameter ϕ_i is employed. The critic network is updated through gradient descent, with the Q loss function represented as follows:

$$L(\phi_i) = E_{a,o,r,o'}[(Q_i(a, o) - y_i)^2] \tag{14}$$

To reduce the overestimation bias of the value function, Q_i is updated using the minimum of the two Critic values. Consequently, y_i is denoted as

$$y_i = r_i + \gamma \min_{j=1,2} Q_{i,j}(o', a')|_{a'=\mu'(o)} \tag{15}$$

In addition, the replay buffer technique is employed to reduce the correlation between training data. This technique involves saving previous experiences in the replay buffer and randomly selecting a small batch of these experiences for network training. The replay buffer is structured as a tuple, denoted by $D : (a_1, \cdots a_N, o, o', r)$.

To achieve a balance between exploring and exploiting, a common approach is to use $\varepsilon-$greedy [12]. Exploring is performed with a probability of ε, where a random action is selected to discover new information. Exploiting, on the other hand, is performed with a probability of $1 - \varepsilon$, where the currently perceived best action is chosen. However, using the same exploration probability throughout the training process may lead to frequent invalid actions by the agent, resulting in inefficiency or even failure to converge. To address this issue, we introduce an exploration coefficient ε that gradually decays with the number of iterations. The decay starts with an initial value of ε_0 and reaches a minimum value of $\varepsilon_{\min}$:

$$\varepsilon = \begin{cases} \varepsilon_0 - \omega D, & if\ \varepsilon \geq \varepsilon_{\min} \\ \varepsilon_{\min} & if\ \varepsilon < \varepsilon_{\min} \end{cases} \tag{16}$$

where ω is the decay rate and D is the number of iterations. At the initial stage of training, a higher exploration rate allows the agent to explore the unknown environment when the knowledge of the environment is limited. However, as training progresses, the agent accumulates more knowledge, and a lower exploration rate becomes more useful in speeding up learning.

3.2. Action and state space

The action space of the UAV is controlled by the dynamics model shown in Eq. 4. The model employs a continuous action space to provide higher control accuracy.

The UAV's state space comprises four components: the coverage of the UAV projection cell, the UAV's own velocity and its position dictating its next movement direction, the UAV's relative position to other UAVs, and information on no-fly areas. This setup allows the UAV to identify no-fly areas within the field of view (FOV), preventing unintended entries.

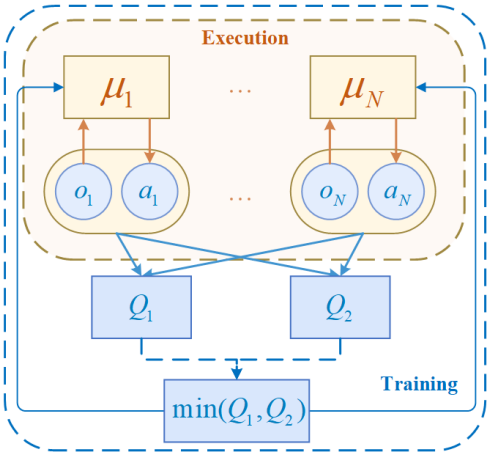


Figure 1. Structure of MATD3.

3.3. Reward function

In Multi-Agent Reinforcement Learning, the thoughtful design of a reward function can significantly enhance collaboration among agents. Based on the objective function and constraints illustrated in Eq. 10, the reward function is formulated as follows.

1) Coverage reward: The coverage reward aims to encourage the UAV to cover as much of the mission area as possible, ultimately increasing coverage. The cognitive reward is represented as

$$r_{1,t} = \omega_1 \sum_{x=1}^{L_1} \sum_{y=1}^{L_2} (m_{x,y}^{t-1} - m_{x,y}^{t}) \tag{17}$$

2) Prevent path overlap: When the UAV repeatedly enters the already covered area, a penalty is given:

$$r_{2,t} = -\omega_2 (N_{cover} - 1) \tag{18}$$

where N_{cover} denotes the number of times the cell where the UAV is located gets overwritten.

3) Collision prevention: To deter UAVs from straying into no-fly areas and to maintain a safe distance from each other, penalties are applied if a UAV strays into a no-fly area or when the distance between them drops below a safe threshold:

$$r_{3,t} = \omega_3 (r_{entry} + r_{collision}) \tag{19}$$

where

$$r_{entry} = \begin{cases} -1 & \text{if} \|n_k - w_{i,t}\| < R_a, \forall k \in \{1, 2, \cdots, N\} \\ 0 & other \end{cases} \tag{20}$$

$$r_{collision} = \begin{cases} -1 & \text{if} \|w_{i,t} - w_{i,t}\|_2 < R_{\text{safe}}, \forall i, j \in W, j \neq i \\ 0 & other \end{cases} \tag{21}$$

4) Preventing out of bounds: A penalty is imposed when the UAV deviates from the designated mission area:

$$r_{4,t} = -\omega_4 \tag{22}$$

Therefore, the overall reward function is formulated as the sum of above four rewards:

$$r_{total} = r_{1,t} + r_{2,t} + r_{3,t} + r_{4,t} \tag{23}$$

Table 1. Parameters in Experiment

Parameter	Value
Width of each cell (L)	50
Maximum speed ($v_{\max}$)	50
Initial exploration coefficient (ε_0)	0.3
Minimum exploration coefficient ($\varepsilon_{\min}$)	0
Exploration coefficient decay rate (ω)	0.000001
ω_1	30
ω_2	10
ω_3	10
ω_4	30
Safe distance between UAVs (R_{safe})	50
Scope of the no-fly area (R_a)	50
Learning rate	0.01
Batch size	1024
Discount rate	0.95

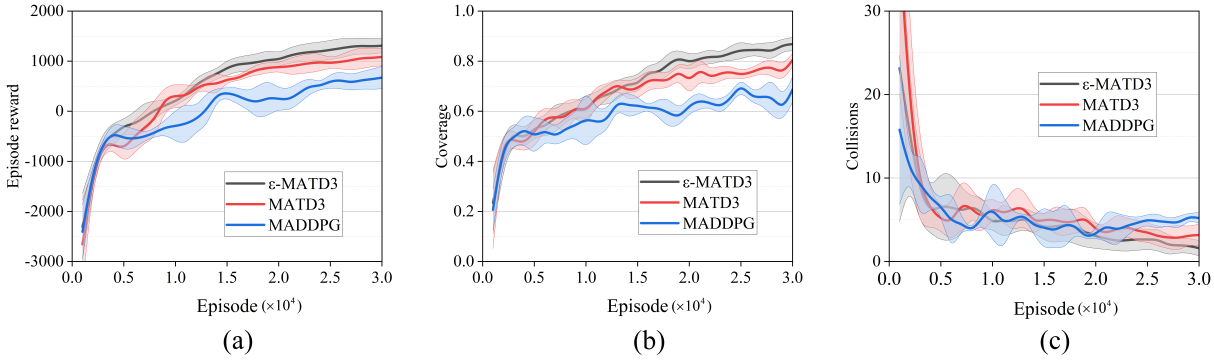


Figure 2. Convergence analysis of $\varepsilon-$MATD3, MATD3, and MADDPG. (a) Average Episode reward. (b) Coverage. (c) Collisions.

4. Experiment Evaluation

Consider a virtual mission environment that is divided into 10×10 square cells, with a buffer area surrounding it. The UAVs' positions and the locations of no-fly areas are initialized at the beginning of each episode. The UAVs have 30 steps for movement. Detailed information on the remaining simulation parameters can be found in Table 1.

The convergence performance of $\varepsilon-$MATD3 is assessed using MATD3 and MADDPG as baselines. The experiments compare three metrics: episode reward, coverage, and number of collisions. The results are sampled and averaged across every 1,000 episodes. Each data point in Fig. 2 reflects an aggregation from five parallel experiments.

Fig. 2 shows that MATD3 outperforms the baseline algorithm MADDPG in terms of convergence. This is because MATD3 uses the minimum of the two critics to estimate Q function, effectively reducing the overestimation bias of the value function. As depicted in Fig. 2(c), collisions in the MADDPG algorithm shows minimal fluctuation before and after training. This suggests that the UAVs may not have encountered other no-fly areas and the agents might not have adequately explored the environmental space.

On the other hand, $\varepsilon-$MATD3 exhibits the fastest convergence and superior results across all three metrics: episodic reward, coverage, and collision numbers. This is because $\varepsilon-$MATD3 introduces the gradually decaying exploration coefficient ε, which enables effective exploration of the ambient space during the early stages of training. As the training progresses, the degree of exploration gradually reduces, relying more on what has been learned in the later stages of

training, thereby resulting in quicker convergence compared to MATD3. The coverage in Fig. 2(b) does not reach 1, possibly due to the presence of inaccessible no-fly areas for UAVs in the environment.

5. Conclusion

This paper addresses the issue of multiple UAVs cooperative coverage path planning by proposing a novel MADRL approach named $\varepsilon-$MATD3. The algorithm is based on MADT3 and employs two critics to estimate Q function. This successfully mitigates the overestimation bias associated with the value function. Additionally, we introduce an exploration coefficient ε that gradually decreases as the number of iterations increases, thereby achieving a balance between exploration and exploitation. The simulation results demonstrate that $\varepsilon-$MADT3 outperforms the baseline algorithm in terms of coverage and number of collisions, effectively resolving the problem of multi-UAV cooperative coverage path planning.

References

[1] Xiang H, Han Y, Pan N, Zhang M, and Wang Z 2023 Study on multi-uav cooperative path planning for complex patrol tasks in large cities *Drones* **6** 367
[2] Zhang B, Lin X, Zhu Y, Jing T, and Zhu Z 2024 Enhancing multi-uav reconnaissance and search through double critic ddpg with belief probability maps *IEEE Trans. Intell. Veh.*
[3] Li J, Sheng H, Zhang J and, Zhang H 2023 Coverage path planning method for agricultural spraying uav in arbitrary polygon area *Aerospace* **10** 755
[4] Hu W, Yu Y, Liu S, She C, Guo L, Vucetic B, and Li Y 2023 Multi-uav coverage path planning: a distributed online cooperation method *IEEE Trans. Veh. Technol.* **72** 11727-40
[5] Muñoz J, López B, Quevedo F, Monje C A, Garrido S, and Moreno L E 2021 Multi uav coverage path planning in urban environments *Sensors* **21** 7365
[6] AlMahamid F and Grolinger K 2022 Autonomous unmanned aerial vehicle navigation using reinforcement learning: a systematic review *Eng. Appl. Artif. Intell.* **115** 105321
[7] Mnih V et al. 2015 Human-level control through deep reinforcement learning *Nature* **518** 529-33
[8] Tilak O and Mukhopadhyay S 2011 Partially decentralized reinforcement learning in finite, multi-agent markov decision processes *AI Commun.* **24** 293-309
[9] Foerster J N, Farquhar G, Afouras T, Nardelli N, and Whiteson S 2018 Counterfactual multi-agent policy gradients *AAAI Conf. Artif. Intell.* **1** 2974-82
[10] Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, and Mordatch I 2017 Multi-agent actor-critic for mixed cooperative-competitive environments *Adv. neural inf. proces. syst.* 6380-91
[11] Ackermann J, Gabler V, Osa T, and Sugiyama M 2019 Reducing overestimation bias in multi-agent domains using double centralized critics *Preprint arXiv:1910.01465*
[12] Liu X, Zhang P, Fang H, and Zhou Y 2021 Multi-objective reactive power optimization based on improved particle swarm optimization with ε-greedy strategy and pareto archive algorithm *IEEE Access* **9** 65650-59