ICIC Express Letters
Volume **17**, Number **8**, August **2023**

ICIC International ©2023 ISSN 1881-803X
pp. 881–887

COLLABORATIVE SCHEDULING TASK LEARNING FOR AFFORDANCE-HETEROGENEOUS MULTI-ROBOT SYSTEMS

Peiliang Wu[1,2,*], Shicheng Luo[1], Liqiang Tian[1], Bingyi Mao[1,2]
and Wenbai Chen[3]

[1]School of Information Science and Engineering
[2]The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province
Yanshan University
No. 438, West Hebei Avenue, Qinhuangdao 066004, P. R. China
2522079421@qq.com; tlq@stumail.ysu.edu.cn; ysdxmby@163.com
*Corresponding author: peiliangwu@ysu.edu.cn

[3]School of Automation
Beijing Information Science and Technology University
No. 12, Xiaoying East Road, Qinghe, Haidian District, Beijing 100192, P. R. China
chenwb@bistu.edu.cn

Received December 2022; accepted February 2023

Abstract. *The use of multi-robots to replace humans in multi-person collaborative tasks such as picking up and delivering items can avoid the risk of cross-contamination of staff during the epidemic, while also reduce the heavy workload of staff to some extent. Often, each person performs a different task in multi-person collaborative tasks, such as pushing, lifting and carrying. So the use of multi-robots with different affordances can better replace the work of human. How to rationally assign tasks to robots, plan delivery routes and improve service efficiency is one of the key technologies in delivery robot research. In this paper, based on multi-agent environment with different affordances, CQL multi-agent deep reinforcement learning algorithm of current distributed multi-robot control method is combined to study path planning and cooperative scheduling in multi-robot system and optimize the cooperation of multi-robot.*

Keywords: Affordance-heterogeneous multi-robot systems, CQL, Multi-agent deep reinforcement learning

1. **Introduction.** The demand for robots is no longer just for single robots, but also for multi-robot systems, thanks to the advancement of robotics and the continued rise in social demand. This is because some tasks cannot be undertaken by a single robot and research shows that the development of individual robots is far more complex and expensive than the development of robotic systems for highly dynamic and complex tasks. At the same time, with the advent of robotic production lines and the need for flexible processing plants, there is a growing desire for multi-robot systems for autonomous operations. In the 1980s, some researchers in robotics applied the theory of multi-agent from artificial intelligence to the study of multi-robot systems [1-5], thus starting the research of multi-robot technology in the field of robotics. Previously, only single-robot systems or Distributed Problem-Solving systems (DPS) that do not involve robots have been studied. If the study of individual robots is likened to the imitation of an individual, the study of multiple robots is the imitation of a social group. Multi-robot systems have many advantages over single robots, mainly in terms of wider application areas, higher efficiency, improved system performance, inherent parallelism, good fault tolerance, lower cost, ease of development, distributed perception and collaboration, scalability, and help in studying group intelligence.

DOI: 10.24507/icicel.17.08.881

In recent years, many researchers have carried out in-depth research on multi-robot systems. Wu et al. [11] developed a simulation environment for heterogeneous robots to carry cooperatively, and applied DDQN (Double Deep Q-Network) algorithm combined with spatial intention mechanism to the simulation environment, so as to improve the performance of agents. Although the algorithm used by Wu et al. has successfully trained multi-agents to complete the handling task, the algorithm still has some shortcomings: the decision algorithm (DDQN) cannot best dispatch all agents. Based on Wu et al., this paper adds CQL (Conservative Q-Learning) method to train multi-agents. In addition, MAPPO (Multi Agent Proximal Policy Optimization) algorithm is added into the experiment for comparison, and the results show that the improved method has better performance than the original method and MAPPO algorithm.

In the second chapter, multi-agent system is introduced. In the third chapter, distributed control method and multi-agent task modeling are first introduced. Then, the CQL algorithm is introduced. In Chapter 4, relevant experiments and experimental results and analysis are presented. Finally, Chapter 5 concludes the paper.

2. Problem Statement and Preliminaries.

The technology of multi-agent systems is developing rapidly, but most of the research is based on multi-agent environments with the same affordance, such as starcraft game environments and multi-agent particle environments [6-10]. Wu et al. [11] developed a new multi-agent environment, in which agents have different affordances, such as lifting objects. This environment is more suitable for real environments such as logistics factories, and more complex than other virtual environments (such as multi-particle environments) that simulate factories. Therefore, research in this environment is more meaningful. However, the training effect of multi-agent algorithm realized in this environment is not ideal. Based on the multi-agent simulation environment with different affordances, this paper studies the path planning and cooperative scheduling of multi-agent to optimize the multi-agent algorithm.

Path planning is the process of finding a feasible optimal path between task points during movement and avoiding collisions during travel [12]. Path planning is based on the correctness of the searched paths, the safety of the robots during operation and the parallelism of the robots in the system. Path planning algorithms are mainly divided into two types: traditional algorithms [13,14] and intelligent heuristic algorithms. Traditional algorithms first load environmental information and then construct paths based on the environmental information. These methods are simple, operate with high reliability and can effectively solve the single robot path planning problem. In multi-robot system path planning problems, traditional methods are no longer effective in solving problems of excessive computational complexity, so intelligent heuristic algorithms are beginning to be used.

Intelligent heuristics mainly include particle swarm algorithms, ant colony algorithms, genetic algorithms, etc. The ant colony algorithm is an iterative random search algorithm that simulates natural organisms. The basic idea is to imitate the process of ants going out for food in nature, by sensing the changes in the pheromone concentration released by ants in the surrounding environment, gradually move to the path with higher pheromone concentration based on self-recognition, and find the shortest path to the target location through multiple iterations of search. Jiang and Zhang proposed an artificial potential field method combined with ant colony algorithm for path planning of mobile robots in static environments to ensure the global search efficiency of the algorithm while avoiding stagnation of the algorithm [15]. Wang proposed a Mixed Max-Min Ant System (MMMAS) for local search, which first finds an approximate optimal solution of a local path using the MMMAS algorithm, and then converts the local paths with four adjacent vertices in the approximate optimal solution into local optimal paths with four vertices and three unequal lines to obtain a better approximation [16]. Luo et al. established a two-step planning fusion optimization method for robot path planning in complex environments where the correctness of the solution cannot be guaranteed. The suboptimal solution of the planning result is first found by the Dijkstra algorithm, and then the approximate optimal solution of the path planning is found on the basis of the suboptimal solution by the exact tracing of the ant colony algorithm [17]. Zhe and Fang addressed the problem of local optimality of ant colony algorithms by subjecting pheromones to statistical analysis to enhance the diversity of the algorithm, so that the improved algorithm can effectively jump out of the local optimal solution [18]. The basic idea of the particle swarm algorithm is to simulate the action of a group of animals and to use the sharing of information between individuals to bring the whole group from disorder to order in order to obtain an optimal solution. Tanweer et al. proposed a new self-regulating particle swarm optimization algorithm that introduces learning schemes to search for the best results in path planning, which uses both self-regulation of inertia weights and self-awareness of the overall search direction [19]. Yan and Hucy introduced an elite ant colony algorithm pheromone selection method based on the original pheromone update affordance to solve the problem that the standard particle swarm optimization algorithm tends to fall into local optimum. This allows the overall effect of the algorithm to maintain a relatively high convergence rate, but also to reduce the possibility of falling into a local optimum solution to a certain extent, making the robot path planning more accurate [20]. Pu et al. proposed a path planning method based on the fusion of improved particle swarm algorithm and ant colony algorithm for multi-objective optimization in path planning of mobile robots, which enables the robot to improve its search capability and stability while ensuring no collision [21].

3. Control Design.

3.1. **Distributed control method.** The structure of the distributed control method is shown in Figure 1. It removes some of the affordances of the centralized central controller, which is only responsible for sending robot tasks and monitoring robot status. Task allocation, path planning, and conflict resolution are all carried out in the robot's internal processor system. Robots calculate independently and synchronize task assignment results by communicating with neighboring robots to achieve consistency within the system. This method of control will be slightly less effective than centralized, but will require much less communication bandwidth. In addition, each robot retains a complete set of "central controllers", so that when one robot crashes, the impact on the others is minimal and does not cause the entire multi-robot system to crash, but task conflict resolution takes more time. This paper uses this improved control for the subsequent assignment of tasks.

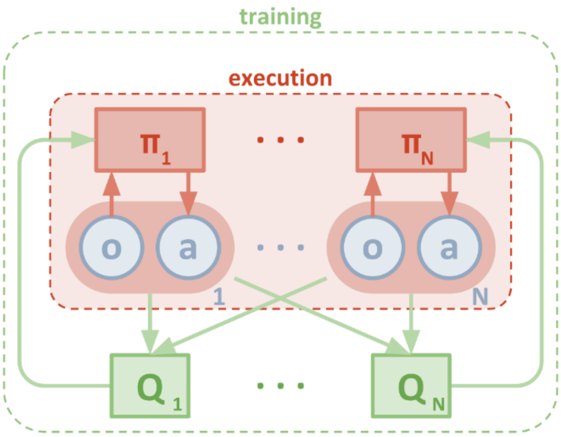


Figure 1. The structure of the distributed control method

3.2. **Multi-robot task assignment modelling.** Firstly, we assume that there are n robots in the multi-robot system that need to handle m tasks. These mobile robots can perform a single task or multiple tasks, and the set of mobile robots can be represented as $S = \{Robot_1, Robot_2, \ldots, Robot_n\}$, tasks can be represented as $T = \{task_1, task_2, \ldots, task_m\}$, $A_i = \{task_1, task_2, \ldots, task_m\}$ represents the combination of robot i and task. The robots all perform their tasks independently and do not consider collaboration. The task assignment goal is to achieve the maximum benefit value for all tasks performed by the robot within the constraints, i.e., the global benefit. Global benefits comprise the benefits of task allocation and path planning. The task allocation model can be described as

$$F(x) = \max \sum_{i=1}^{N} \sum_{j=1}^{M} R_{ij} \left(L_{ij}, \theta\right) \cdot s_{ij} \tag{1}$$

where L is the task effectiveness variable, and L_{ij} represents the value of task j for robot i which is related to the path length of the task. R_{ij} denotes mission proceeds, and θ is the set of variables that affect the effectiveness of the task. s_{ij} is the judgmental formula for performing tasks, $s_{ij} = 1$ represents robot i performing task j, and $s_{ij} = 0$ represents robot i not performing task j. $N = \{1, 2, 3, 4, \ldots, n\}$ is the robot index collection. $M = \{1, 2, 3, 4, \ldots, m\}$ is the collection of task indexes.

3.3. **Conservative Q-Learning for DDQN.** In this paper, we use the convolutional neural network (CNN) [22] to process the robot vision information and feed the vision information to the Conservative Q-Learning for DDQN algorithm after processing. CNN is shown in Figure 2.

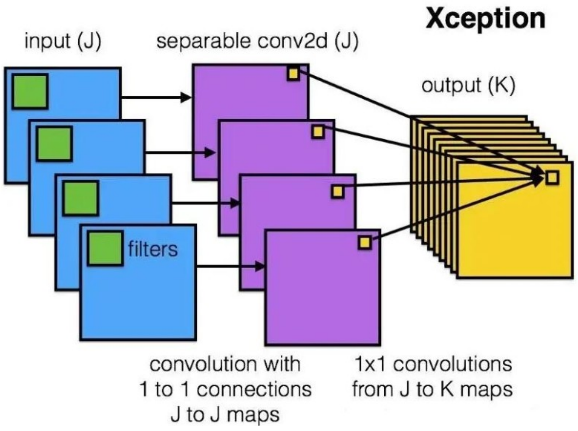

Xception
input (J)
separable conv2d (J)
output (K)
filters
convolution with 1 to 1 connections J to J maps
1x1 convolutions from J to K maps

Figure 2. CNN diagram

The key of offline reinforcement learning algorithm is to avoid the overestimation problem caused by distribution deviation. The CQL algorithm directly starts from the value function, aiming to find the lower bound estimate of the original value function, and then uses it to optimize the policy with a more conservative policy value. Conservative Q-Learning for DDQN (CQL+DDQN) adopts the training framework of DDQN algorithm, and adopts the updating mode of Q function of CQL.

In Figure 3, we show that the resulting Q-function, $\hat{Q}^{\pi} := \lim_{k\to\infty} \hat{Q}^{k}$, lower-bounds Q^{π} at all (s, a).

4. **Main Results.** The simulation environment is a number of robots with different affordances (pushing, grasping, collecting, throwing) transporting scattered objects to a specified location in an enclosed space or collected by a collecting robot. The process of robot generate action is shown in Figure 4. And the simulation environment is shown in Figure 5.

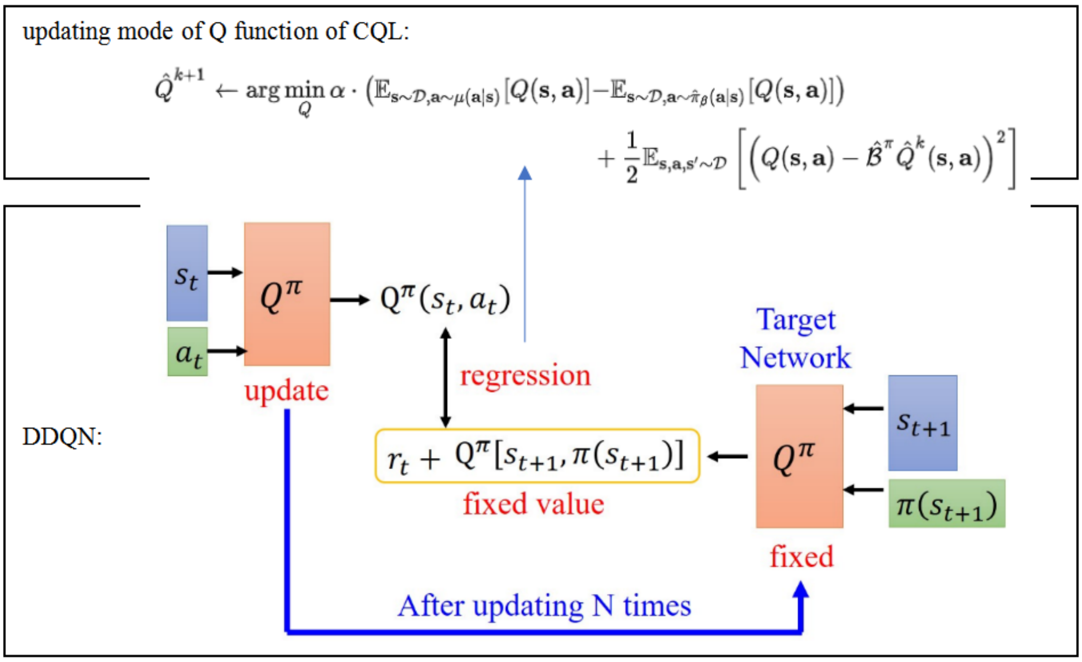

updating mode of Q function of CQL:
$\hat{Q}^{k+1} \leftarrow \arg\min_Q \alpha \cdot \left(\mathbb{E}_{s\sim\mathcal{D},a\sim\mu(a|s)}[Q(s,a)] - \mathbb{E}_{s\sim\mathcal{D},a\sim\hat{\pi}_\beta(a|s)}[Q(s,a)]\right) + \frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q(s,a) - \hat{\mathcal{B}}^\pi \hat{Q}^k(s,a)\right)^2\right]$
DDQN:
s_t
a_t
Q^π
update
$Q^\pi(s_t, a_t)$
regression
$r_t + Q^\pi[s_{t+1}, \pi(s_{t+1})]$
fixed value
Target Network
Q^π
s_{t+1}
$\pi(s_{t+1})$
fixed
After updating N times

Figure 3. Conservative Q-Learning for DDQN (CQL+DDQN) algorithm

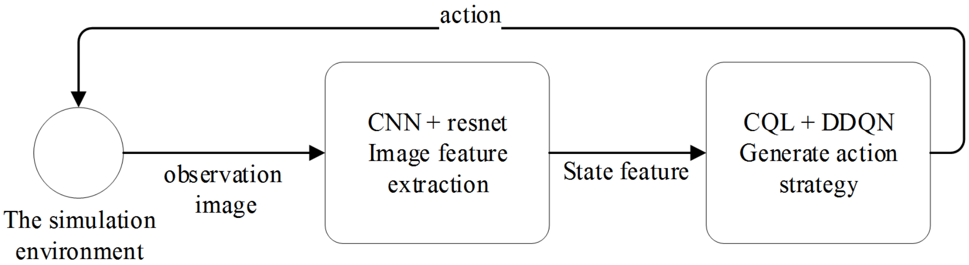
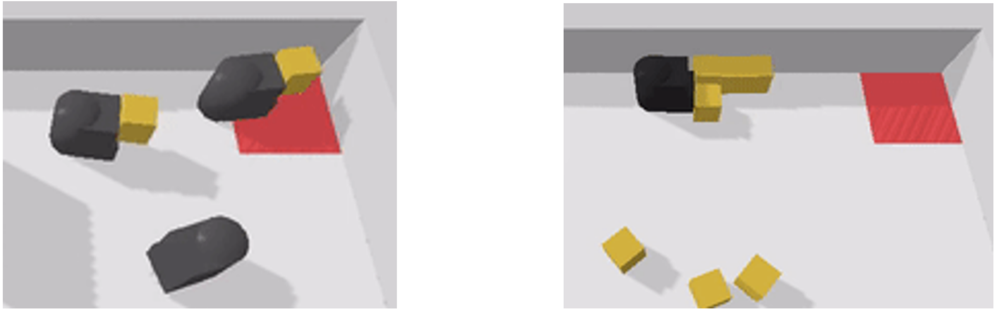

action
The simulation environment
observation image
CNN + resnet Image feature extraction
State feature
CQL + DDQN Generate action strategy

Figure 4. Interaction process between robot and environment

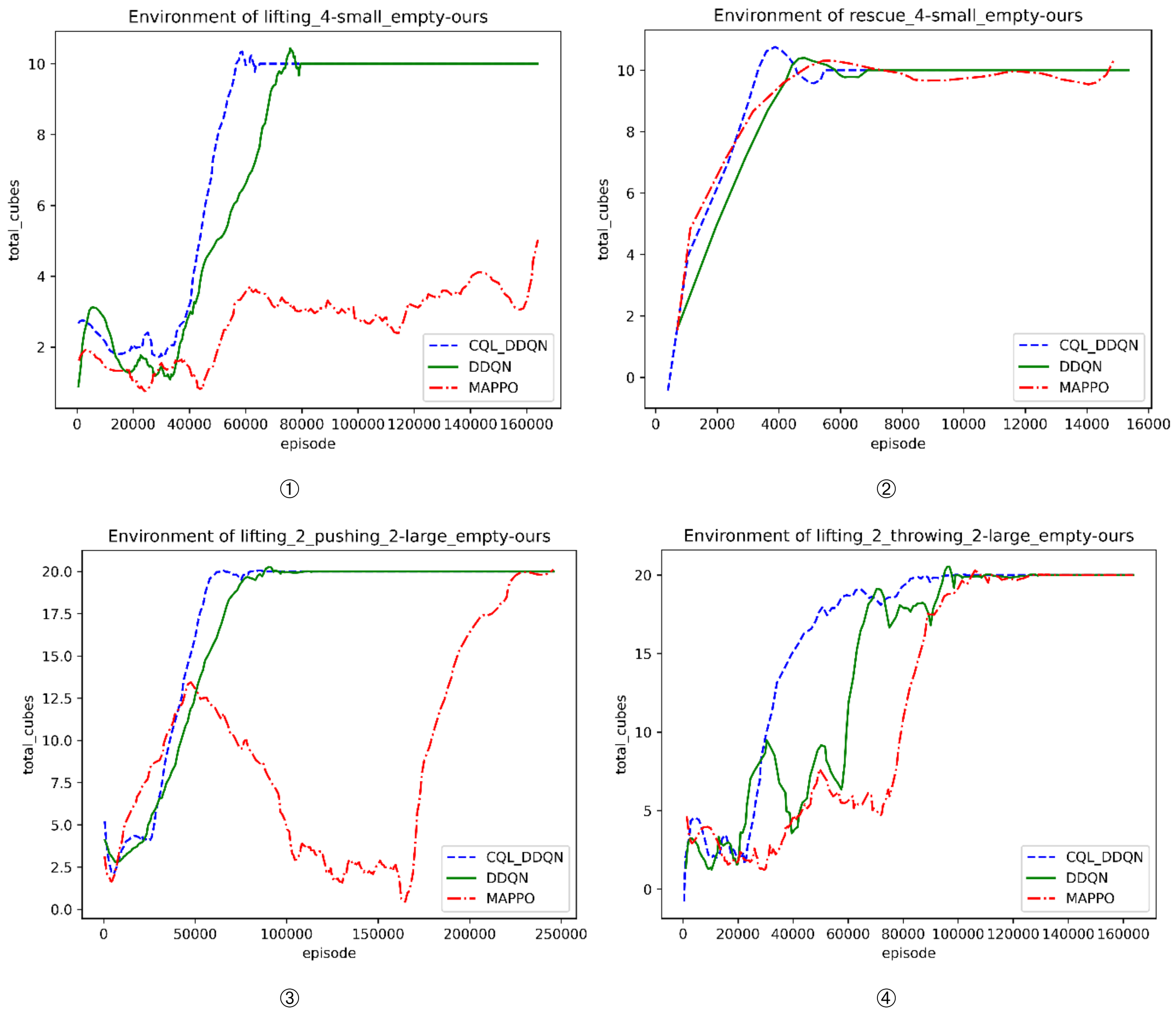


"Lift" robots "Push" robots

Figure 5. Partially affordance-heterogeneous robot simulation diagram

In this paper, we have performed simulations for four scenarios: ① four "lift" robots working together; ② four "rescue" robots working together; ③ two "lift" robots working together with two "push" robots; ④ two "lift" robots working together with two "throw" robots. Experiments using the MAPPO algorithm [23] are used as comparison experiments. The experimental results showed that the training effect of the four groups was improved under the restriction of 164000 rounds of training steps.

Figure 6 shows the change curve of total cubes during training in four simulation environments. It is not difficult to find that CQL converges faster than others in all


Environment of lifting_4-small_empty-ours

Environment of rescue_4-small_empty-ours

Environment of lifting_2_pushing_2-large_empty-ours

Environment of lifting_2_throwing_2-large_empty-ours

Figure 6. Experimental results

simulation environments. And the MAPPO algorithm does not perform well in all four environments.

All experiments were performed on the same device with a GPU of 3090, and the hyperparameters such as the learning rate during algorithm training were consistent with the experimental settings of Wu et al.

5. **Conclusions.** This paper adds the state-of-the-art multi-agent deep reinforcement learning algorithm CQL to the distributed multi-robot control method to achieve an autonomous learning method for robots without human manipulation, which can more accurately and faster recognize blocks of objects. The multi-robot learns autonomously from the reward values fed back by the environment and eventually succeeds in transporting multiple object blocks to the target areas.

Acknowledgment. This work is partially supported by the National Key R&D Program of China (2018YFB1308300), National Natural Science Foundation of China (U20A20167), Natural Science Foundation of Hebei Province (F202103079), Innovation Capability Improvement Plan Project of Hebei Province (22567626H), and Beijing Natural Science Foundation (4202026).

REFERENCES

[1] Y. T. Chen and W. J. Chen, Optimizing the obstacle avoidance trajectory and positioning error of robotic manipulators using multigroup ant colony and quantum behaved particle swarm optimization algorithms, *International Journal of Innovative Computing, Information and Control*, vol.17, no.2, pp.595-611, 2021.

[2] Y. U. Cao, A. S. Fukunaga and A. Kahng, Cooperative mobile robotics: Antecedents and directions, *Autonomous Robots*, vol.4, no.1, pp.7-27, 1997.

[3] D. Carmel and S. Markovitch, Opponent modeling in multi-agent systems, in *Adaptation and Learning in Multi-Agent Systems*, G. Weiß and S. Sen (eds.), Heidelberg, Springer, 1996.

[4] C. Claus and C. Boutilier, The dynamics of reinforcement learning in cooperative multi-agent systems, *Proc. of the 15th National Conference on Artificial Intelligence and 10th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI-1998)*, Madison, US, pp.746-752, 1998.

[5] N. V. Findler and G. D. Elder, Multiagent coordination and cooperation in a distributed dynamic environment with limited resources, *Artificial Intelligence in Engineering*, vol.9, no.3, pp.229-238, 1995.

[6] Y. Yang, J. Hao, B. Liao, K. Shao, G. Chen, W. Liu and H. Tang, Qatten: A general framework for cooperative multiagent reinforcement learning, *arXiv.org*, arXiv: 2002.03939, 2020.

[7] I. J. Liu, U. Jain, R. A. Yeh and A. Schwing, Cooperative exploration for multi-agent deep reinforcement learning, *International Conference on Machine Learning*, pp.6826-6836, 2021.

[8] J. Hu, S. Hu and S. W. Liao, Policy regularization via noisy advantage values for cooperative multi-agent actor-critic methods, *arXiv.org*, arXiv: 2106.14334, 2021.

[9] C. Muise, V. Belle, P. Felli, S. McIlraith, T. Miller, A. R. Pearce and L. Sonenberg, Efficient multi-agent epistemic planning: Teaching planners about nested belief, *Artificial Intelligence*, vol.302, 2022.

[10] L. Yuan, J. Wang, F. Zhang, C. Wang, Z. Zhang, Y. Yu and C. Zhang, Multi-agent incentive communication via decentralized teammate modeling, *Proc. of AAAI Conference on Artificial Intelligence*, 2022.

[11] J. Wu, X. Sun, A. Zeng, S. Song, S. Rusinkiewicz and T. Funkhouser, Spatial intention maps for multi-agent mobile manipulation, *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp.8749-8756, 2021.

[12] J. H. Wang, T. Y. Weng and Q. F. Zhang, A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows, *IEEE Transactions on Cybernetics*, vol.49, no.7, pp.2467-2478, 2019.

[13] G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux and M. Djemai, Decentralized motion planning and scheduling of AGVs in an FMS, *IEEE Transactions on Industrial Informatics*, vol.14, no.4, pp.1744-1752, 2017.

[14] R. J. Alitappeh, K. Jeddisaravi and F. G. Guimarães, Multi-objective multi-robot deployment in a dynamic environment, *Soft Computing*, vol.21, no.21, pp.6481-6497, 2017.

[15] J. Jiang and H. C. Zhang, Research on robot path planning based on improved potential field ant colony algorithm, *Computer Simulation*, vol.33, no.9, pp.329-334, 2016.

[16] Y. Wang, Hybrid max-min ant system with four vertices and three lines inequality for traveling salesman problem, *Soft Computing*, vol.19, no.3, pp.585-596, 2015.

[17] Q. Luo, H. Wang, Y. Zheng and J. He, Research on path planning of mobile robot based on improved ant colony algorithm, *Neural Computing and Applications*, vol.32, no.6, pp.1555-1566, 2020.

[18] W. Zhe and L. Fang, Self-adaptive ant colony algorithm based on statistical analysis and its application, *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pp.313-317, 2019.

[19] M. R. Tanweer, S. Suresh and N. Sundararajan, Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems, *Information Sciences*, vol.326, pp.1-2, 2015.

[20] X. Yan and Y. Hucy, Elite particle swarm optimization algorithm and its application in robot path planning, *Optics & Precision Engineering*, vol.21, no.12, pp.3160-3168, 2013.

[21] X. C. Pu, J. J. Li, H. C. Wu and Y. Zhang, Mobile robot multi-goal path planning using improved particle swarm optimization, *CAAI Transactions on Intelligent Systems*, vol.12, no.3, pp.301-309, 2017.

[22] M. Edwards and X. Xie, Graph convolutional neural network, *British Machine Vision Conference*, 2016.

[23] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. M. Bayen and Y. Wu, The surprising effectiveness of MAPPO in cooperative multi-agent games, *arXiv.org*, arXiv: 2103.01955, 2021.