Hindawi
Mobile Information Systems
Volume 2022, Article ID 1147819, 9 pages
https://doi.org/10.1155/2022/1147819

Research Article

A Method of Multi-UAV Cooperative Task Assignment Based on Reinforcement Learning

Xiaohu Zhao,[1,2] **Hanli Jiang**,[1] **Chenyang An**,[1] **Ruocheng Wu**,[1] **Yijun Guo**,[1] **and Daquan Yang**[1]

[1]*School of Information and Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*China Academic of Electronics and Information Technology, Beijing 100041, China*

Correspondence should be addressed to Yijun Guo; guoyijun@bupt.edu.cn and Daquan Yang; ydq@bupt.edu.cn

Received 26 April 2022; Accepted 29 June 2022; Published 12 August 2022

Academic Editor: Adarsh Kumar

With the increasing complexity of UAV application scenarios, the performance of a single UAV cannot meet the mission requirements. Many complex tasks need the cooperation of multiple UAVs. How to coordinate UAV resources becomes the key to mission completion. In this paper, a task model including multiple UAVs and unknown obstacles is constructed, and the model is transformed into a Markov decision process (MDP). In addition, considering the influence of strategies among UAVs, a multiagent reinforcement learning algorithm based on SAC algorithm and centralized training and decentralized execution framework, MA-SAC (Multi-Agent Soft Actor-Critic), is proposed to solve the MDP. Simulation results show that the algorithm can effectively deal with the task allocation problem of multiple UAVs in this scenario, and its performance is better than other multiagent reinforcement learning algorithms.

1. Introduction

Unmanned aerial vehicle, also known as UAV, has the characteristics of strong mobility, low safety risk coefficient, no need for personnel to take off, repeatability, and so on. UAV was first used in military fields [1], such as reconnaissance, target strike, air early warning, and electronic jamming. In recent years, UAV technology is developing rapidly, the size of UAV is decreasing, and the cost is getting lower and lower. Therefore, UAV is more and more widely used in civil fields such as sensing [2], cargo transportation, communication relay [3], fire monitoring, and aerial mapping.

With the increasingly complex application scenarios, such as the combination with the Internet of vehicles [4], a single UAV cannot effectively complete complex and diverse tasks. It is important to make multi-UAV perform tasks collaboratively not only to meet the requirement of complicated scenarios but also to make the accomplishment of tasks to cause less time-and-resource consumption.

Task planning is the most important part for the cooperative execution of multi-UAV, and task allocation is the basis of task planning. Task assignment refers to the complex task environment existing in several UAVs; after taking full account of the energy consumption, load, nature, role, and other constraints of UAVs, the coordination between UAVs and various resources is coordinated to assign one or more orderly tasks to each UAV, so as to minimize the time and cost and ensure the efficient and successful completion of tasks to the maximum extent.

The task allocation problem is generally approximated to the path planning problem [5], that is, how to generate a collision-free path from the starting site to the destination to ensure the safety of the vehicle [6]. However, in the multi-UAV environment, not only the collision between UAVs and obstacles but also the collision between UAVs should be considered. At the same time, with the increase of the number of UAVs, the variation of the environment is also increasing. In addition, every action decision of each UAV can be regarded as simultaneous, and no one UAV can know the current decision of other UAVs, so it is more difficult to avoid collisions between UAVs.

Fortunately, reinforcement learning (RL) techniques are emerging to help solve the problem of real-time decision-making in complex and changing environments. The technology allows the drone to learn a strategy to maximize returns or achieve a specific purpose through its constant interaction with the environment.

In this paper, a UAV task allocation model including UAV collision and communication energy consumption is presented; at the same time, an MA-SAC algorithm is proposed to assign tasks and plan paths to UAVs.

The specific works of this paper are as follows:

(i) A multi-UAV task assignment model based on collision and communication energy consumption is proposed

(ii) Based on this assignment model, the dynamic process of task assignment is transformed into MDP

(iii) A multi-agent reinforcement learning algorithm MA-SAC is proposed to solve the MDP process

The rest of this article is organized as follows. Section 2 describes the related work. In Section 3, the multi-UAV task assignment model is presented. Section 4 introduces the task assignment algorithm proposed in this paper. In Section 5, simulation is performed and the results are analyzed. Finally, the works of this paper are summarized in Section 6.

2. Related Work

In the past few years, many researchers have done a lot of research on multi-UAV task allocation model and the algorithm to solve the model. They not only make the model more close to the increasingly complex reality environment but also look for high-performance algorithms. This section will introduce relevant work from these two aspects.

2.1. Task Allocation Model. In various scenarios, different task allocation models need to be established based on a variety of problems that need to be solved by UAV. In the paper [7], and this problem is modeled as a traveling salesman problem (TSP), which minimizes the total flight time and total range of all UAVs by considering the flight capability of UAVs. Jia et al. [8] construct a heterogeneous UAV cooperative multitask allocation scenario by considering kinematic constraints, resource constraints, time constraints, and vehicle path model. Song et al. [9] describe the UAV logistics problem as a mixed integer linear programming problem considering UAV flight time, load, and other constraints. In addition, the task allocation problem of multi-UAV is usually described as multidimensional multiple choice knapsack problem (MMKP) [10, 11], dynamic network flow optimization (DNFO) problem [12], and multiple processors resource allocation (CMTAP) problem [13, 14].

2.2. Task Assignment Algorithm. Task assignment algorithms are mainly divided into optimization algorithm, heuristic algorithm, and reinforcement learning algorithm.

Optimization methods include Hungarian algorithm [15, 16], branch-and-bound method [17], and other commonly used integer linear programming methods. These algorithms are only applicable to scenarios with simple tasks and small UAV scale. Their calculations grow exponentially as the number of UAVs increases, and these algorithms cannot generate an accurate trajectory for UAVs in complex environments. Heuristic algorithms are proposed relative to optimization algorithms, including GA [18], ACO, and PSO that simulate animal behavior in nature. These algorithms are generally combined with other algorithms to solve task assignment problems. In [18], GA is combined with clustering algorithm to solve the task allocation and path planning problems of multiple UAV. In [19], the author proposed two improved heuristic algorithms to solve TSP problems, one is IGA algorithm proposed by improving the coding rules of genetic algorithm, and the other is PSO-ACO algorithm combining PSO and ACO. In [20], the author improves swarm gap algorithm and puts forward three algorithms: location loop (AL), sorting and allocation loop (SAL), and limit and allocation loop (LAL), which solves the task allocation problem of the UAV team in a military operation. However, the heuristic algorithm has the disadvantage of falling into local optimum easily, and the real-time performance of the algorithm is worse and worse with the increase of environment complexity. Therefore, many researchers began to study the application of reinforcement learning in task assignment.

Reinforcement learning is a kind of algorithm that makes an agent learn the optimal strategy through trial and error in the environment. Reinforcement learning has been widely used in UAV mission assignment scenarios over the past few years. In [21], a transaction inspired multiagent reinforcement learning algorithm was proposed to solve the path planning and coordination problems of UAV clusters. In reference [22], the author proposed a MADOL algorithm to enable multiple UAVs to solve the ambiguous BSN allocation problem in an ambiguous boundary scenario. The literature [23] has developed a multiagent reinforcement learning framework, which solves the problem of dynamic resource allocation of UAV communication network in uncertain environment and realizes the balance between performance gain and UAV overhead. In reference [24], the author proposed a multiagent reinforcement learning algorithm, compound-action actor-critic (CA2C), which solves the problem that UAVs perform sensing tasks through cooperative sensing and transmission. In [25], the author proposed an FTA algorithm by combining DQN algorithm with priority experience replay, which effectively solved the problem of UAV task allocation in uncertain environment. In [11], the author proposed a DDQN-per algorithm to solve the task assignment problem of MCS. However, these single-agent algorithms regard the agents in the environment as independent and cannot train a good agent cooperation model. The proposed MADDPG [26] algorithm adopts the method of centralized training and distributed deployment, which well solves the problem of cooperation and competition among multiagent. In [27], the author proposed an MADDPG algorithm, trained the

MADDPG model offline, and then solved the resource allocation problem in the UAV-assisted vehicle network online. However, DDPG algorithm is a deterministic strategy, which may fall into local optimum due to greed. The proposed SAC algorithm [28] introduces entropy, which requires not only maximum reward but also maximum entropy to enhance the spatial exploration ability of agents. Based on the idea of centralized training and separate deployment, this paper applies SAC algorithm to the cooperative task assignment environment of multiple UAVs and proposes an MA-SAC algorithm.

3. Task Assignment Model

Multi-UAV should not only complete each task but also pay attention to their own safety and energy consumption. Figure 1 shows the task allocation framework of multi-UAV. In this paper, the distance from UAV to the mission positions, the collision of UAV, and the communication between UAV and base station are comprehensively considered to establish the task assignment model, and the specific modeling is as follows.

3.1. The Distance between the UAV and the Mission. This paper considers how to assign multiple UAVs to multiple task points and plan a safe path so as to achieve the goal of reducing the total cost while completing the task quickly and safely. In this paper, the UAV cluster is represented by $V = \{v_1, v_2, v_3, \ldots, v_n\}V = \{v_1, v_2, v_3, \ldots, v_n\}$. The position and track data of each UAV can be obtained by the GPS device carried by the UAV itself, and the data will be transmitted to the MEC layer for calculation. For each UAV $v_i \in V$, (s_{xi}, s_{yi}) is used to represent its current position.

The set of tasks to be completed is represented by $W = \{w_1, w_2, w_3, \ldots, w_n\}$. For each task $w_i \in W$, (s_{wxi}, s_{wyi}) is used to represent task position.

The distance between the UAV v_i and the mission location w_j can be calculated using the following formula:

$$L_{ij} = \sqrt{\left(s_{xi} - s_{wxi}\right)^2 + \left(s_{yj} - s_{wyj}\right)^2}. \tag{1}$$

3.2. UAV Collision. In order to simulate the real environment, some obstacles are added to the environment to block the route of UAV. At the same time, the collision between UAV and other UAVs is considered. As shown in the picture, there is a certain safety buffer area between the UAV and the obstacles.

The distance between UAVs can be calculated using the following formula:

$$L_{uav} = \sqrt{\left(s_{xi} - s_{xj}\right)^2 + \left(s_{yi} - s_{yj}\right)^2}. \tag{2}$$

Once the distance between UAVs or between UAVs and obstacles is less than the safety zone, UAVs are considered to have a safety risk of collision.

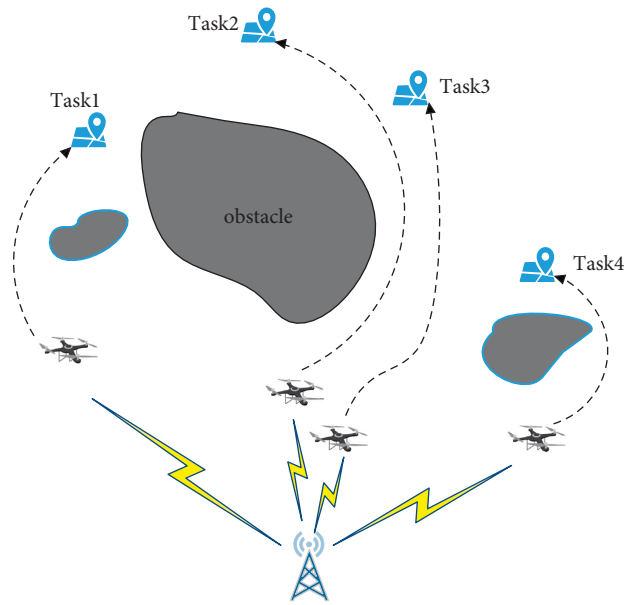

Figure 1: Multi-UAV task assignment model.

3.3. UAV Communication. In order to grasp the status of UAV in real time, the communication between UAV and base station needs to be considered, and the position of base station is represented by (B_x, B_y). In this paper, UAV's altitude to the ground is h,and the straight-line distance between UAV and base station can be calculated by the following formula:

$$L_{uav-base} = \sqrt{\left(s_{xi} - B_x\right)^2 + \left(s_{yj} - B_y\right)^2 + h^2}. \tag{3}$$

Transmitting the data collected by UAV sensors needs to consume the energy of the sensor node [29]. In order to study the energy loss of UAV transmission, we consider the path loss of UAV communication with base station. In Friis free space model [30], the relationship between signal transmitting power and signal receiving power can be calculated by the following formula:

$$P_R = \frac{P_T G_T G_R \lambda^2}{(4\pi)^2 d^2 \beta}, \tag{4}$$

where P_R is the receiving signal power, P_T is the transmitting signal power, G_T is the transmitting antenna gain, G_R is the receiving antenna gain, λ is the signal wavelength, β is the system loss factor unrelated to propagation, and d is the propagation distance. In this paper, d is the distance between each time slot UAV and the base station $L_{uav-base}$.

In order to ensure normal communication, the power of the attenuated UAV signal needs to be greater than the receiving power of the base station. Therefore, the signal transmitting power of each time slot n of UAV v_i must meet the formula

$$P_{Ti}[n] \geq \frac{(4\pi)^2 d^2 \beta}{G_T G_R \lambda^2} P_{Ri}. \tag{5}$$

The communication energy consumption of each UAV v_i to complete the task can be expressed as

$$E_{\text{com-i}} = \sum_{n\in\mathbb{N}} P_{Ti}[n]\delta, \tag{6}$$

where $\mathbb{N} = (n_1, n_2, n_3, \ldots, n_t)$ is the time slot set for the UAV to complete the task. In this paper, the time slot n is approximated to each step in the simulation. δ is the duration of each time slot n. In this model, δ is set to 1.

The total communication energy consumption of UAV cluster can be calculated by the formula

$$E_{\text{com}} = \sum_{v_i\in V} E_{\text{com}-i}. \tag{7}$$

4. Task Assignment Algorithm

In this section, we consider the application of reinforcement learning in multi-UAV task allocation, apply a soft actor-critic (SAC) algorithm to multiagent environment, and propose an MA-SAC algorithm. This algorithm is usually used to solve the problem described as Markov decision process (MDP). So, this section will introduce the MDP of this model, SAC algorithm and MA-SAC algorithm in turn.

4.1. Markov Decision Process. MDP is usually composed of state, action, and reward function. Therefore, the MDP of the model can be described as follows.

4.1.1. State. In this process, the state space is composed of the position and speed of the UAV, the distance between the UAV and the destination, and the collision risk of the UAV.

4.1.2. Action. The action space is usually the optional action set of all UAVs in different states. In this model, the action space of UAV is expressed as < front, back, left, right, hover >.

4.1.3. Reward. In this model, when multiple UAVs are faced with multiple tasks, this paper aims to reasonably allocate task targets and carry out path planning for each UAV, so that each task can be completed safely and quickly with the minimum total energy consumption. Therefore, for UAV v_i, the reward can be described as

$$R_i = R_F + R_L + R_c - E_{\text{com}-i}. \tag{8}$$

The task assignment problem can be described as

$$\max \sum_{v_i\in V} R_i, \tag{9}$$

$$\bigcup_{i=1}^{n} w_{ij} = V, \tag{10}$$

$$\bigcup_{i=1}^{n} v_{ij} = W, \tag{11}$$

where R_F is the reward for completing the task, and the value is constant. R_c is the collision reward. R_L is the distance reward. In order to guide the UAV to the mission point, it can be expressed as $R_L = -\min L_{ij}$, $j \in (1, 2, \ldots, n)$, w_{ij} indicates that the mission w_i is carried out by UAV v_j, and v_{ij} indicates that UAV v_i performs mission w_j. Formula (10) means that only one UAV can be assigned to perform each task, and formula (11) means that each UAV can only perform one task.

4.2. SAC Algorithm. SAC algorithm is a kind of off-policy reinforcement learning algorithm. This paper is improved based on SAC algorithm proposed in [31]. The algorithm improves the critical network on the first version of SAC algorithm [32]. It removes the value network and uses two Q networks. Therefore, the SAC algorithm has one actor network, two critic networks, and two target-critic networks. Among them, the actor network is used to give the corresponding action according to the change of state, and the critic network is used to calculate the Q value to evaluate the action. In order to solve the overestimation problem, the SAC algorithm adopts a pair of independent critic network and takes the smaller value of the two when updating. In order to stabilize the training of Q network, the SAC algorithm introduces a pair of target-critic networks whose update frequency is less than the critic network.

In order to prevent the strategy from getting into trouble due to greed, it is necessary to increase the random exploration ability of the algorithm, so SAC introduces entropy regularization. When the strategy distribution is more uniform, the entropy of the strategy is greater, and the random exploration ability of the algorithm is stronger. Therefore, the objective function of SAC algorithm not only requires the maximum final reward but also the maximum entropy. Its objective function can be expressed as

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t, a_t) + \alpha H(\pi(\cdot \mid s_t))\right]$$

$$\pi^*_{\max} = \operatorname{argmax}_\pi \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t, a_t) + \alpha H(\pi(\cdot \mid s_t))\right], \tag{12}$$

where $H(\pi(\cdot \mid s_t))$ is the entropy of strategy, $r(s_t, a_t)$ is the reward for time t, and $\pi^*_{\max}$ is the optimal strategy.

4.3. MA-SAC Algorithm. Figure 2 shows the MA-SAC algorithm that we proposed by improving SAC algorithm based on the multi-UAV task allocation model. MA-SAC algorithm is based on actor-critic network framework. In this multi-UAV environment, each UAV has an actor network, a target-actor network, two critic networks, and two target-critic networks, which are all composed of fully connected neural networks.

In the multi-UAV environment, UAV itself is not only an intelligent body but also a part of the environment of other UAVs. Therefore, for the critic network of each UAV, we not only input the environmental state into the critic network. The actions of other UAVs are also fed into the critic network to calculate the Q by a part of the overall environment. SAC, like DDPG and other algorithms, introduces the experience replay mechanism to reduce the

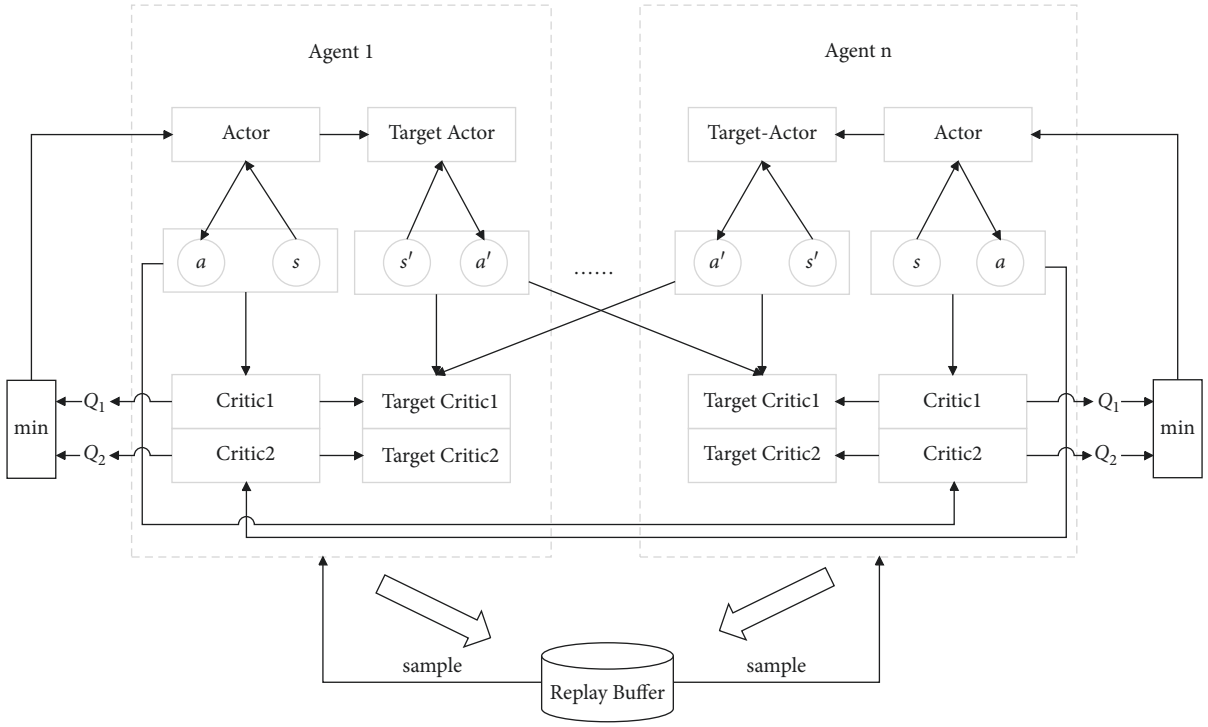

Figure 2: Actor and critic neural network of MA-SAC.

correlation between data. Therefore, the whole training process is divided into two parts: experience collection and network training. In the experience gathering phase, the agent performs the actions generated in each step, and then stores the tuples that include states, action, next state, and reward $\langle S, A, S\prime, R\rangle$ into the replay buffer.

When the data in the replay buffer reaches the threshold, the network training stage can be entered. At each step, some data will be sampled from the replay buffer to update the parameters of actor networks and critic networks. The actor network is trained by the strategy gradient. For each UAV $v_i \in V$, the actor network update targets are as follows:

$$J(\theta_i) = \mathbb{E}_{X,a\sim D}\left[\alpha \log\left(\pi_i(a_i|s_i)\right) - Q_i^{\pi}(X, a_1, \ldots, a_n)|_{a_i=\pi_i(s_i)}\right], \tag{13}$$

where π_i represents the policy network of the agent i, $\theta_i \in \{\theta_1, \theta_2, \ldots, \theta_n\}$ represents the parameter of the policy network π_i, and X represents the current status of all agents.

Critic networks are updated by minimizing the loss function as a goal. The loss function is the mean square error that can be calculated by the formula:

$$\mathscr{L} = \mathbb{E}_{(X,a,r,X')\sim D}\left[\left(Q_i^{\pi}(X, a_1, \ldots, a_n) - y_i\right)^2\right], \tag{14}$$

$$y_i = r_i + \gamma \mathbb{E}\left[Q_i^{\pi}(X', a_1', \ldots, a_n')|_{a_i'=\pi_{\overline{\theta}_i}(s_i')} - \alpha \log\left(\pi_{\overline{\theta}_i}(a_i'|s_i')\right)\right], \tag{15}$$

where X' represents the next status of all agents, a_i' represents the next action of the agent i, and s_i' represents the next state of the agent i.

To ensure the stability of training, the parameters of actor networks and critic networks will be copied to the corresponding target networks in each iteration. Here, the algorithm adopts the soft update method, so in each step, some actor and critic network parameters are updated to the corresponding target network, which can be calculated by the formula

$$\overline{\psi} \longleftarrow \tau\psi + (1-\tau)\overline{\psi}, \tag{16}$$

$$\overline{\theta} \longleftarrow \tau\theta + (1-\tau)\overline{\theta}, \tag{17}$$

where $\overline{\psi}$ is the parameter of target-critic network, ψ is the parameter of the critic network, and τ is the update ratio.

The pseudocode of the MA-SAC algorithm is demonstrated in Algorithm 1, and the meanings of the parameters are shown in Table 1.

5. Experimental Results and Analysis

In this section, the performance of MA-SAC algorithm in multi-UAV task assignment environment is studied. We use the Pytorch deep learning framework to simulate this scenario and compare it with MADDPG algorithm. Table 2 shows the relevant hyperparameters of the algorithm simulation in this paper.

In this experiment, we constructed an environment in which multi-UAV cooperate to complete tasks. The environment consists of three UAVs, three mission positions, one obstacle, and a base station to communicate with the UAVs. Firstly, the MADDPG algorithm proposed in reference [26] is selected to compare the convergence performance. Figure 3 shows the convergence process of MA-

Table 1: Explanation of variables and functions in the algorithm of MA-SAC.

Variable	Explanation
episodes	The maximum number of iterations
steps	The maximum step length for each iteration
D_{size}	The amount of data in the replay buffer
B_{size}	Sampling number

(1) Initialize environment
(2) Initialize critic network and actor network
(3) Initialize max episodes, replay buffer, batch size
(4) **for** *episode* ∈ [1, *episodes*] **do**
(5) Reset environment
(6) Get current state s_i for each agent, i
(7) **for** *step* ∈ [1, *steps*] **do**
(8) Select actions a_i for each agent v_i
(9) Get all agents next states s_i' and rewards r_i
(10) Store $< a_i, s_i, s_i', r_i >$ to replay buffer D
(11) **if** $D_{\text{size}} > B_{\text{size}}$ **then**
(12) Sample batch B from replay buffer D
(13) **for** v_i, where $i = 1{:}N$ **do**
(14) Update the critic network
(15) Update the actor network
(16) Update the target network according to formulas (15), (16)
(17) **end for**
(18) **end if**
(19) **end for**
(20) **end for**

Algorithm 1: Algorithm of MA-SAC.

Table 2: The parameters of simulation.

Parameter	Value
Number of UAVs	3
Number of tasks	3
Number of obstacles	1
Number of base stations	1
Steps of episode	35
Capacity of replay buffer	1000000
Number of network neurons	128
Learning rate	0.001
Discount factor of reward	0.99
Update ratio of target network τ	0.001

SAC algorithm and MADDPG algorithm during training in this environment. In this experiment, we performed 50,000 training episodes and averaged the rewards every 1,000 episodes. By comparing the two algorithms, it can be found that the proposed MA-SAC algorithm can finally converge to around 300, while the MADDPG algorithm finally converges to around 220. It can be seen that the convergence speed of the two algorithms is similar in this scenario, but the convergence result of the MA-SAC algorithm is better than that of the MADDPG algorithm, because the training goal of the MA-SAC algorithm is not only to maximize the reward of the drone but also to maximize the entropy of the UAV

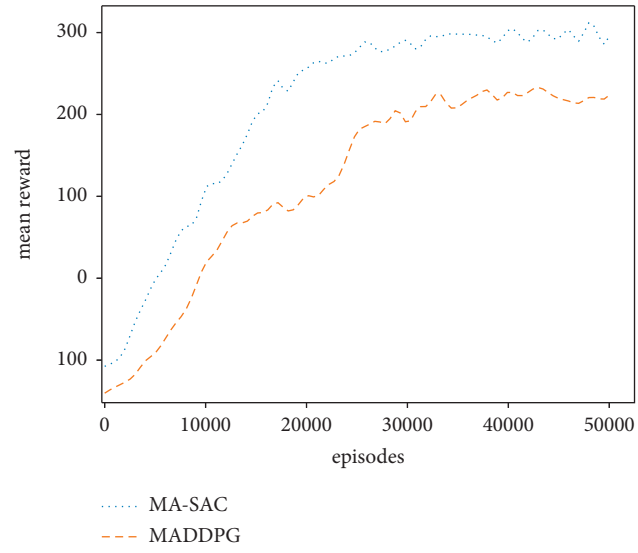

Figure 3: Reward of different algorithms.

strategy. This increases the ability of the UAV to explore the space, thereby improving the performance of the algorithm.

To verify the effectiveness of the algorithm in this scenario, we conducted 500 episodes of tests on the MA-SAC algorithm in this environment and compared it with other

Table 3: Task completion rate.

Algorithm	Task completion rate (%)
MA-SAC	95.16
MADDPG	92.76
COMA	82.67
VDN	68.34

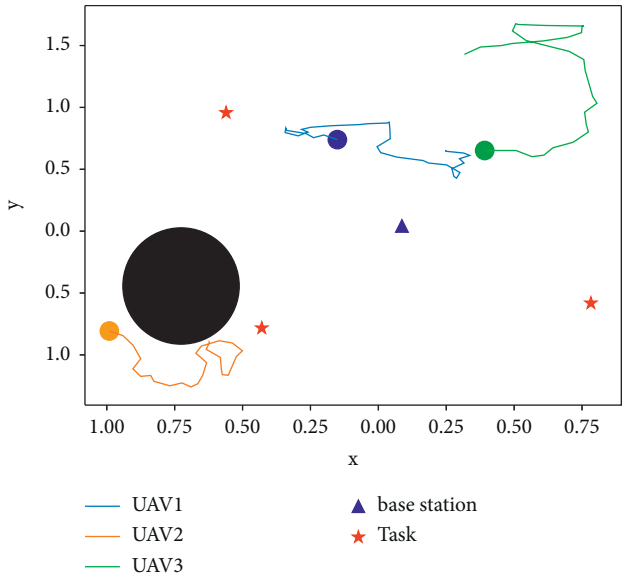

Figure 4: Rendering of task assignment during 0 w episodes of training.

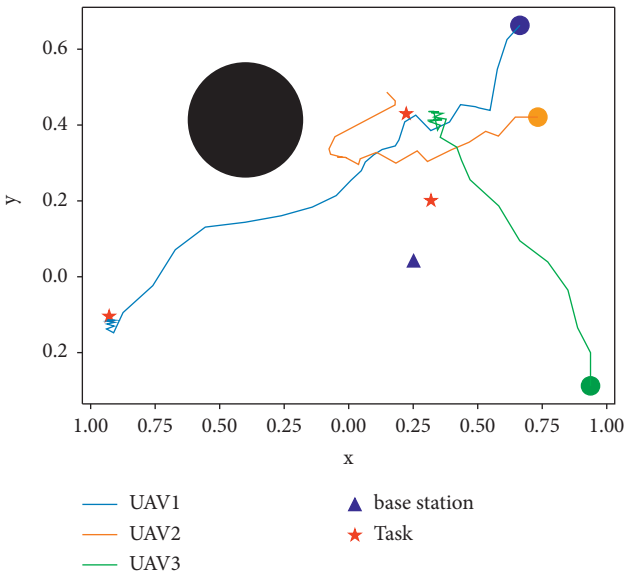

Figure 5: Rendering of task assignment during 2 w episodes of training.

multiagent reinforcement learning algorithms. As shown in Table 3, the task completion rate of the MA-SAC algorithm reaches 95.16%, which is a great improvement compared with that of the COMA and VDN algorithms, and the task

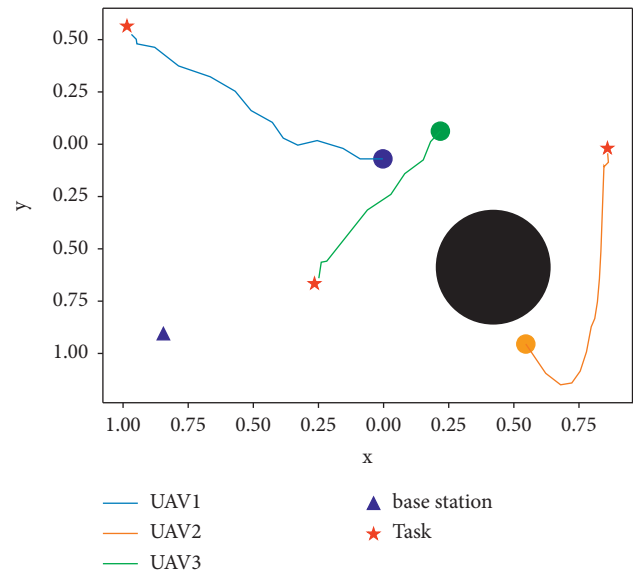

Figure 6: Rendering of task assignment during 5 w episodes of training.

completion rate is also increased by 2.4% compared with the MADDPG algorithm.

Figure 4 shows the dynamic assignment process of UAVs in the task area before training. At this time, none of the three UAVs has learned any strategy, so they are in an exploration state in the environment. It can be seen from the route of the UAV in the task assignment process that the UAV does not have a clear mission target at this time, and they move randomly in space. UAV 2 even collides with obstacles.

Figure 5 shows the rendering of the multi-UAV task assignment process when using the proposed MA-SAC algorithm for 20,000 episodes of training. It can be seen that although the UAVs have learned to approach the mission point at this time, there is no coordination between them. Both UAV 2 and UAV 3 flew to the same mission location, resulting in not all missions being completed.

Figure 6 shows the effect of the task assignment process of the UAV when the training reaches 50,000 episodes. At this point, the trained model can already solve the task assignment problem in this environment well. UAVs not only consider their distance when assigning tasks but also take into account the strategies of other UAVs and cooperate with each other to complete all tasks in the mission area. At the same time, UAVs have also learned to stay away from obstacles to reduce their own risks when completing tasks. It can be seen that UAV 2 is relatively close to the obstacle at the beginning, so there is a possibility of collision. In order to ensure its own safety, it first flies away from the obstacle, and then flies to the mission location after reaching the safe area.

6. Conclusions

In this paper, a multi-UAV cooperative task assignment model in complex environment is constructed by considering UAV distance, collision, and communication. Meanwhile, we

propose an MA-SAC algorithm to solve the model by combining the SAC algorithm of deep reinforcement learning with multiagent framework of centralized training and decentralized execution. Simulation results show that the MA-SAC algorithm is superior to the MADDPG algorithm in convergence result in multi-UAV task allocation environment. In terms of task completion rate, the model trained by the MA-SAC algorithm also achieved a better result.

In the future work, more complex factors will be considered in the environment, such as making the communication model more suitable for real scenes and weather changes. At the same time, it will also study the larger-scale dynamic task allocation of UAV. Since this paper only studies the UAV cooperation scenario, the UAV task allocation in the countermeasure scenario will be studied in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China (11974058 and 61901050); Beijing Nova Program (Z201100006820125) from Beijing Municipal Science and Technology Commission; Beijing Natural Science Foundation (Z210004); and State Key Laboratory of Information Photonics and Optical Communications (IPOC2021ZT01), BUPT, China.

References

[1] L. Bertuccelli, H. L. Choi, and P. Cho, "Real-time multi-UAV task assignment in dynamic and uncertain environments," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, p. 5776, Ontario, Canada, September, 2009.

[2] S. Huang, A. Liu, S. Zhang, N. N. Wang, and N. N. Xiong, "BD-VTE: a novel baseline data based verifiable trust evaluation scheme for smart network systems," *IEEE transactions on network science and engineering*, vol. 8, no. 3, pp. 2087–2105, 2021.

[3] K. Zhu, X. Xu, and Z. Huang, "Energy-efficient routing algorithms for UAV-assisted mMTC networks," in *Proceedings of the 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, IEEE, Istanbul, Turkey, September, 2019.

[4] K. Gao, F. Han, P. Dong, N. Xiong, and R Du, "Connected vehicle as a mobile sensor for real time queue length at signalized intersections," *Sensors*, vol. 19, no. 9, p. 2059, 2019.

[5] X. Tao and A. S. Hafid, "Trajectory design in UAV-aided mobile crowdsensing: a deep reinforcement learning approach," in *Pocedings of the IEEE ICC*, June, 2021.

[6] H. Li, J. Liu, K. Wu, Z. Yang, R. W. Liu, and N. Xiong, "Spatio-temporal vessel trajectory clustering based on data mapping and density," *IEEE Access*, vol. 6, Article ID 58939, 2018.

[7] N. Ozalp, U. Oztop, and E. Oztop, "Cooperative multi-task assignment for heterogonous UAVs," in *Proceedings of the 2015 International Conference on Advanced Robotics (ICAR)*, pp. 599–604, (ICAR), Istanbul, Turkey, July, 2015.

[8] Z. Jia, J. Yu, X. Ai, X. Xu, and D. Yang, "Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm," *Aerospace Science and Technology*, vol. 76, pp. 112–125, 2018.

[9] B. D. Song, K. Park, and J. Kim, "Persistent UAV delivery logistics: MILP formulation and efficient heuristic," *Computers & Industrial Engineering*, vol. 120, pp. 418–428, 2018.

[10] L. R. Rodrigues, J. P. P. Gomes, and J. F. L. Alcântara, "Embedding remaining useful life predictions into a modified receding horizon task assignment algorithm to solve task allocation problems," *Journal of Intelligent and Robotic Systems*, vol. 90, no. 1-2, pp. 133–145, 2018.

[11] M. Alighanbari and J. How, "Robust decentralized task assignment for cooperative UAVs," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 6454, San Francisco, California, August, 2006.

[12] K. E. Nygard, P. R. Chandler, and M. Pachter, "Dynamic network flow optimization models for air vehicle resource allocation," in *Proceedings of the American Control Conference*, pp. 1853–1858, Arlington, VA, USA, June, 2001.

[13] S. Fei, C. Yan, and S. Lin-Cheng, "UAV cooperative multi-task assignment based on ant colony algorithm," *Acta Aeronautica et Astronautica Sinica*, vol. 29, no. 5, pp. 188–s189, 2008.

[14] E. Edison and T. Shima, "Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms," *Computers & Operations Research*, vol. 38, no. 1, pp. 340–356, 2011.

[15] K. Kim and C. S. Hong, "Optimal task-UAV-edge matching for computation offloading in UAV assisted mobile edge computing," in *Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–4, IEEE, Matsue, Japan, September, 2019.

[16] B. S. MirzaeiniaA and M. Hassanalian, "Drone-station matching in smart cities through Hungarian algorithm: power minimization and management," in *Proceedings of the AIAA Propulsion and Energy 2019 Forum*, p. 4151, Indianapolis, August, 2019.

[17] S. J. Rasmussen and T. Shima, "Branch and bound tree search for assigning cooperating UAVs to multiple tasks," in *Proceedings of the 2006 American Control Conference*, p. 6, June, 2006.

[18] Y. Ma, H. Zhang, Y. Zhang, R. Gao, Z. Yang, and J. Yang, "Coordinated optimization algorithm combining GA with cluster for multi-UAVs to multi-tasks task assignment and path planning," in *Proceedings of the 2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pp. 1026–1031, Edinburgh, UK, July, 2019.

[19] J. Chen, F. Li, and Y. Li, "Travelling salesman problem for UAV path planning with two parallel optimization algorithms," in *Proceedings of the 2017 Progress in Electromagnetics Research Symposium - Fall (PIERS - FALL)*, pp. 832–837, Singapore, November, 2017.

[20] J. Schwarzrock, I. Zacarias, A. L. C. Bazzan, R. Q. de Araujo Fernandes, L. H. Moreira, and E. P. de Freitas, "Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence," *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 10–20, 2018.

[21] A. A. Khalil, A. J. Byrne, and M. A. Rahman, "Efficient uav trajectory-planning using economic reinforcement learning," 2021, https://arxiv.org/abs/2103.02676.

[22] Y. Zhang, Z. Mou, F. Gao, L. Xing, J. Jiang, and Z. Han, "Hierarchical deep reinforcement learning for backscattering data collection with multiple UAVs," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3786–3800, 2021.

[23] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 729–743, 2020.

[24] J. Hu, H. Zhang, L. Song, R. Schober, and H. V. Poor, "Cooperative Internet of UAVs: distributed trajectory design by multi-agent deep reinforcement learning," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6807–6821, 2020.

[25] X. Zhao, Q. Zong, B. Tian, B. Zhang, and M. You, "Fast task allocation for heterogeneous unmanned aerial vehicles through reinforcement learning," *Aerospace Science and Technology*, vol. 92, pp. 588–594, 2019.

[26] H. Qie, D. Shi, T. Shen, X. Xu, Y. Wang, and L. Wang, "Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning," *IEEE Access*, vol. 7, Article ID 146264, 2019.

[27] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 131–141, 2021.

[28] R. Lowe, Y. Wu, and A. Tamar, "Multi-agent actor-critic for mixed cooperative-competitive environments," 2017, https://arxiv.org/abs/1706.02275.

[29] M. Wu, L. Tan, and N. Xiong, "A structure fidelity approach for big data collection in wireless sensor networks," *Sensors*, vol. 15, no. 1, pp. 248–273, 2014.

[30] T. S. Rappaport, *Wireless communications -- principles and practice*, Prentice Hall PTR, Hoboken, New Jersey, 2002.

[31] T. Haarnoja, A. Zhou, and K. Hartikainen, "Soft actor-critic algorithms and applications," 2018, https://arxiv.org/abs/1812.05905.

[32] T. Haarnoja, A. Zhou, and P. Abbeel, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the International Conference on Machine Learning*, pp. 1861–1870, PMLR, Chengdu, China, July, 2018.