sensors

MDPI

Article

Cooperative Search Method for Multiple UAVs Based on Deep Reinforcement Learning

Mingsheng Gao * and Xiaoxuan Zhang

College of Internet of Things, Hohai University, Changzhou 213002, China
* Correspondence: gaoms@hhu.edu.cn

Abstract: In this paper, a cooperative search method for multiple UAVs is proposed to solve the problem of low efficiency of multi-UAV task execution by using a cooperative game with incomplete information. To improve search efficiency, CBBA (Consensus-Based Bundle Algorithm) is applied to designate the tasks area for each UAV. Then, Independent Deep Reinforcement Learning (IDRL) is used to solve Nash equilibrium to improve UAVs' collaborations. The proposed reward function is smartly developed to guide UAVs to fly along the path with higher reward value while avoiding the collisions between UAVs during flights. Finally, extensive experiments are carried out to compare our proposed method with other algorithms. Simulation results show that the proposed method can obtain more rewards in the same period of time as other algorithms.

Keywords: task assignment; multi-UAV; deep reinforcement learning

Citation: Gao, M.; Zhang, X. Cooperative Search Method for Multiple UAVs Based on Deep Reinforcement Learning. *Sensors* **2022**, *22*, 6737. https://doi.org/10.3390/s22186737

Academic Editor: Petros S. Bithas

Received: 8 August 2022
Accepted: 29 August 2022
Published: 6 September 2022

1. Introduction

Nowadays, unmanned aerial vehicles (UAVs), or drones, are widely used in the witness of a fast-paced development [1]. Equipped with radar, cameras and other equipment, UAVs can be used in military areas [2] such as for tracking, positioning and battlefield detection. However, due to the limitation of fuel load, it is difficult for a single UAV to search a large area. Compared with a single UAV, multiple UAVs can perform more complex tasks. Multiple UAVs sharing information and searching cooperatively can improve the efficiency of task execution. In the process of the search task, the path planning of the multi-UAV is a crucial problem [3].

For the above problem, many scholars have proposed some multi-UAV path planning algorithms. For instance, hierarchical decomposition is one of the effective way to solve the problem. The clustering algorithm is first used for the multi-UAV task assignment. Then the path planning is based on the Voronoi diagram [4] or genetic algorithm [5]. However, these path planning algorithms require a prior knowledge about the environment and centralized task assignment algorithms require a control center to communicate among UAVs, which is not suitable in dynamic scenarios. On the other hand, multi-agent reinforcement learning (MARL) is effective to solve the above problem. The essence of MARL is a stochastic game. MARL combines the Nash strategies of each state into a strategy for an agent while constantly interacting with the environment to update the Q value function in each state of the game. Nash equilibrium solution in MARL can replace the optimal solution to obtain an effective strategy [6].

In this paper, we explicit Independent Deep Reinforcement Learning (IDRL) to solve the problem of low efficiency when multiple UAVs perform tasks simultaneously. CBBA [7] (Consensus-Based Bundle Algorithm) is first used for task assignment for multiple UAVs under constraints of time and fuel consumption. Then the UAV chooses the best strategy to complete the task based on the states and actions of other UAVs. A new reward function is developed to guide the UAV to choose the path with high value and punish collisions between UAVs.

The main contributions of this paper are summarized as follows:

(1) Different from the centralized path planning algorithm that a central controller is required for task allocation, a distributed path planning algorithm is designed in this paper. UAVs can communicate with each other for task allocation and path planning in a more flexible way.
(2) A cooperative search method is proposed. Before selecting the next action, the UAV needs to adopt the corresponding cooperative method according to the incomplete information obtained to improve the efficiency of task execution.
(3) A new reward function is proposed to avoid collisions between UAVs while guiding UAVs to target points.

2. Related Work

In this section, we review the literature that settles multi-UAV target assignment and path planning (MUTAPP), figure out their pros and cons and clarify the remaining gaps and challenges for further investigations.

MUTAPP problem is an NP-hard problem in essence, which implies there is no perfect solution to an NP-hard problem. However, for small- and/or medium-sized problems, it is possible to be solved. Hierarchical decomposition is one of the effective methods to solve MUTAPP [8], which decomposes the MUTAPP problem into task assignment and path planning.

At present, MUTAPP is mainly divided into traditional MTSP and objective function optimization problems. For traditional MTSP, Wang et al. [9] try to use genetic algorithms for task assignment and cubic spline interpolation for path planning. In [10], Liu et al. use Overall Partition Algorithm (OPA) for task assignment and use cycle transitions to generate shortest paths. Simulation results show that the proposed algorithm achieves better performance than traditional algorithms based on GA to solve MTSP problems. In [11], Dubins curves are used to model the UAV kinematics model to make the generated path more realistic. The improved particle swarm optimization algorithm based on heuristic information is proposed to solve MTSP. The results show that the proposed algorithm can generate paths in a small number of iterations. However, in practical applications, the multi-UAV system not only needs to consider the total flight distance, but also the efficiency of the task which is usually evaluated by the objective function.

With a single UAV and no altitude effects, the standard coverage path planning (CPP) problem has been studied extensively in the literature [12,13]. The objective function of CPP is defined as the area of the covered region. Miles et al. [14] proposes rectangle partition relaxation (RPR) algorithm to divide the UAV flight area. In [15], based on the single UAV algorithm, a density-based sub-region of UAV coverage with a unique role is proposed to optimize the coverage area. Xie et al. [16] provides a mixed-integer programming formula for CPP and develops two algorithms based on this method to solve the TSP-CPP problem. Based on this research, Xie extends the proposed algorithm in [17] and proposes a branch-and-bound-based algorithm to find the optimal route. Although these algorithms continuously optimize the UAV coverage area, it is difficult to evaluate the efficiency of UAV execution with a single constraint.

In [18], the objective function of the multi-UAV system is to minimize energy loss. K-means is used to assign tasks to multiple UAVs, and then genetic algorithm is used to generate specific paths. In [19], simulated annealing algorithm is used to increase the coverage area of the UAV. Reference [18] uses the more advanced k-means++; the experimental results show that the generated path is shorter than k-means. In [20], the Minimum Spanning Tree (MST) is used to generate trajectories and simulation results show that compared with other algorithms, the generated trajectories can obtain more rewards during task execution. Also, there are some algorithms [21–25] that use clustering algorithm to solve MUTAPP related problems. However, clustering algorithm is sensitive to noise points. If the task point is far from the central point, it will be assigned to the UAV separately, which is unrealistic.

Wang et al. [26] use MST to decompose MTSP into multiple TSPs, and then Ant colony algorithm is used to solve TSP. In [27], a fuzzy approach with a linear complexity level is used to convert the MTSP to several TSPs, then Simulated Annealing (SA) is used to solve each problem. Similarly, Cheng et al. [28] decouples the MTSP problem into TSP and solves the subproblems through sequential convex programming. Reference [29] propose a task allocation algorithm based on maximum entropy principle (MEP). Simulation results show that the proposed MEP algorithm achieves better performance than SA algorithm. Cao et al. [30] introduces Voronoi diagram method into Ant colony algorithm and the unmanned aerial vehicle cooperative task scheduling strategy which conclude task allocation and path planning is gained. Compared with clustering algorithm, these algorithms are more flexible in task allocation and the number of tasks performed by each agent is reduced through reasonable task allocation, which increases the execution efficiency of the algorithm. However, since the cooperation among agents is not considered, which would affect the efficacy of these algorithms.

MARL provides a new solution for MUTAPP problems; it models the decision-making process in the multi-agent environment as a random game where each agent needs to make decisions according to the strategies of other agents. MARL has become a prevalent method to solve the problem of multi-agent cooperation.

In [31], DRL is used to generate paths for data collected by multiple UAVs without prior knowledge. Reference [32] uses MADDPG for the cooperative control of four agents; the experimental results show that MADDPG has good performance in complex environments and successfully learns the strategy of multi-agent collaboration. However, with the instability of the environment caused by the increase in the number of agents, the proposed algorithm has certain difficulties in the joint action space. In [33], MADDPG is used to control the formation of multiple agents during transportation in order to prevent the agent from colliding with other agents on the way to the target point. Chen et al. [34] use MARL for the collaborative welding of multiple robots. The way of cooperation between robots is also to prevent collisions between agents.

Han et al. [35] use MADDPG for both task assignment and path planning, and a reward value function is designed to guide the UAV to the target point and avoid collisions between UAVs. In fact, the cooperative approach of avoiding conflict can improve the success rate of task execution but does not directly affect the efficiency of task execution. Also, the proposed algorithm only works in environments where each agent performs one task and cannot be used to solve the multiple traveling salesman problem. Moreover, the performance of value-based reinforcement learning is better than that of policy-based reinforcement learning in the task environment with few actions.

3. Overview of CBBA

In this section, we will review the CBBA algorithm, which is generally divided into two parts: bundle construction and conflict resolution.

3.1. Bundle Construction

In the process, each CBBA agent creates only one bundle and updates it during the allocation process. In the first phase of the algorithm, each agent keeps adding tasks to its bundle set until no other tasks can be added.

During the task assignment process, each agent needs to store and update the following four necessary information vectors: a bundle $b_i \in (\mathcal{J} \cup \{\varnothing\})^{L_t}$, the corresponding path $p_i \in (\mathcal{J} \cup \{\varnothing\})^{L_t}$, the winning agent list $z_i \in \mathcal{J}^{N_t}$ and the winning score list $y_i \in R_+^{N_t}$.

The sequence of tasks in the bundle is arranged according to the order in which the tasks are added to the collection, and the tasks in the path are arranged according to the order in which the tasks are best executed. Note that the vector size of b_i and p_i cannot be greater than the maximum assigned task number L_t. $S_i^{p_i}$ is defined as the total reward

score value of the task i performing the task along the path p_i. In CBBA, adding task j to bundle b_i will result in an increase in marginal scores:

$$c_{ij}[b_i] = \begin{cases} 0, & if\ j \in b_i \\ \max\limits_{n \le |p_i|+1} S_i^{p_i \oplus_n \{j\}} - S_i^{p_i}, & otherwise \end{cases} \quad (1)$$

where $|\cdot|$ represents the vector size of the list, $\oplus_n$ represents the insertion of a new element after the n-th element of the vector (in the later part of this article, $\oplus_{end}$ will also be used to indicate the addition of a new element at the end of the vector). CBBA's bundle scoring scheme inserts a new task into the position where the highest score increases, which will be the marginal score associated with the task in a given path. Therefore, if the task is already included in the path, there will be no extra scores.

The score function is initialized as $S_i^{\{\varnothing\}} = 0$, and the path and bundle are recursively updated to

$$b_i = b_i \oplus_{end} \{J_i\}, p_i = p_i \oplus_{n_i, J_i} \{J_i\} \quad (2)$$

where $J_i = argmax_j(c_{ij}[b_i \times h_{ij}),\ n_{i,J_i} = argmax_n S_i^{p_i \oplus_n \{J_i\}},\ h_{ij} = II(c_{ij} > y_{ij})$ *and* $II(\cdot)$ indicates an index function that having a value of 1 when the judgment result is true and a value of 0 when the judgment result is false. The bundle algorithm is continuously looped until $|b_i| = L_t$ or $h_i = 0$.

3.2. Conflict Resolution

In the conflict resolution phase, there are three aspects that need to be communicated to reach a consensus. The two vectors that have been introduced are the winning score list $y_i \in R_+^{N_t}$ and the winning agent list $z_i \in \mathcal{J}^{N_t}$. The third vector $s_i \in \mathrm{R}^{N_u}$ represents the timestamp of the last information update from each other agent. The time vector is updated by:

$$s_{ik} = \begin{cases} \tau_r, & if\ g_{ij} = 1 \\ \max\limits_{m: g_{im}=1} s_{mk}, & otherwise \end{cases} \quad (3)$$

where τ_r is the message reception time.

When agent i receives a message from agent k, z_i and s_i are used to determine the information of which agent in each task is up to date. For task j, agent i has three possible actions:

1. Update: $y_{ij} = y_{kj}, z_{ij} = z_{kj}$
2. Reset: $y_{ij} = 0, z_{ij} = \varnothing$
3. Leave: $y_{ij} = y_{ij}, z_{ij} = z_{ij}$

Table 1 in [7] outlines the decision rules for information interaction between agents.

Table 1. Task area coordinates and ROR of Map (a).

No	Coordinate	ROR	No	Coordinate	ROR
0	(35, 5)	4.2	8	(35, 45)	5.9
1	(65, 5)	4.9	9	(55, 55)	3.5
2	(25, 15)	4.4	10	(75, 55)	5.2
3	(55, 15)	4.2	11	(15, 65)	4.1
4	(5, 25)	3.7	12	(45, 65)	4.3
5	(45, 25)	4.6	13	(35, 75)	5.7
6	(75, 25)	3.8	14	(55, 75)	2.3
7	(15, 35)	5.6	15	(65, 35)	5.9

If the elements in the winning score list change due to communication, each agent will check whether the updated or reset tasks were in their bundle. If the task is actually in the bundle, then this task and all other tasks added to the bundle later will be released:

$$y_{i,b_{in}} = 0, z_{i,b_{in}} = \varnothing, \ \forall n > \overline{n_l} \tag{4}$$

$$b_{in} = \varnothing, \ n \geq \overline{n_l} \tag{5}$$

where b_{in} represents the $\boldsymbol{n}$-th element of the bundle, and $\overline{n_l} = \min\{n : z_{i,b_{in}} \neq i\}$. It should be noted that the task that adding to the winning agent and the winning list after $b_{i,\overline{n_l}}$ will be reset because the deletion can change all the task scores after b_{in}. After completing the second phase of conflict resolution, the algorithm will return to the first phase and add a new task.

4. IDRL Based Path Planning Algorithm

Independent Reinforcement Learning (IRL) is widely and successfully applied in the field of multi-agent autonomous decision-making. This paper uses IDRL to solve Nash equilibrium in a cooperative game with incomplete information, and each UAV chooses the optimal strategy according to the states and actions of other UAVs to maximize the total rewards.

4.1. System Model

In this paper, we establish a model based on IDRL to enhance the efficiency of task execution through multi-UAV cooperation. We make the following assumptions:

(1) Any two UAVs with intersected flight paths can communicate with each other to know the states and actions when the distance between them is less than a threshold. The game between UAVs belongs to incomplete information games.
(2) Each UAV can choose the optimal strategy according to the state and action of other UAVs, so the game between UAVs belongs to cooperative games.
(3) The UAVs do not choose actions at the same time, so the game between UAVs belongs to dynamic games.

The task environment of multiple UAVs is briefly divided into two-dimensional grids, as shown in Figure 1. The blue part represents the task area to be executed.

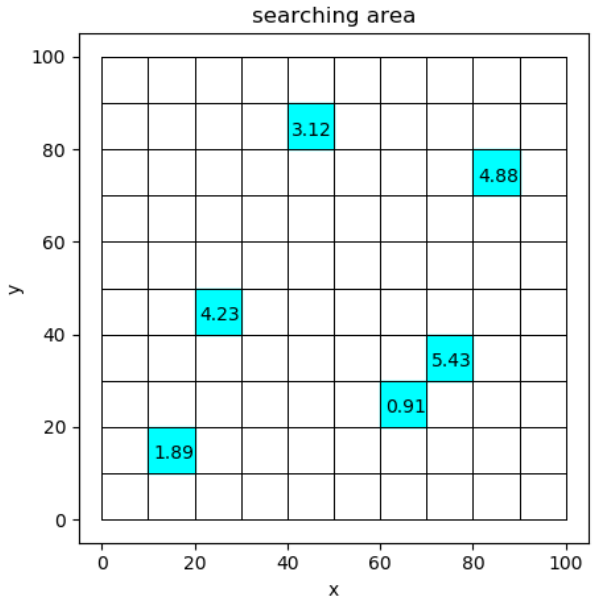


Figure 1. Grid partition of a searching area.

In the cooperative game with incomplete information, the objective function of UAVs is to maximize the search efficiency. ROR and revenue defined in [36] are used to evaluate the search efficiency of UAVs.

The detection function is used to estimate the detection ability in a probable target grid j with time consumption z. A common exponential form of regular detection function is given as:

$$b(j,z) = 1 - e^{-\varepsilon z} \tag{6}$$

where ε is a parameter related to the UAV equipment, z represents time consumption.

When an UAV is searching in grid j, the revenue function defined as

$$e(j,z) = p(j)b(j,z) \tag{7}$$

where $p(j)$ represents the target probability in grid j.

The efficiency of a multi-UAV system to perform a search task is assessed by the amount of reward per unit of time earned by multiple UAVs. Therefore, the ROR of grid j is introduced with a definition as:

$$ROR(j,z) = \frac{d(e)}{d(z)} = \varepsilon \cdot e^{-\varepsilon z} \cdot p(j)b(j,z) \tag{8}$$

where indicates that the ROR value decreases as the search time z increases. In other words, a lower ROR indicates that the area is searched more thoroughly.

We assume that each UAV knows the ROR value of all grids, as shown in Figure 1. The problem can be solved into strategies on CBBA and DRL.

4.2. Nash Equilibrium in MARL

In MARL, $V_i(\pi_1, \cdots, \pi_i, \cdots, \pi_n)$ represents the expected reward of i-th agent under the joint strategy $(\pi_1, \cdots, \pi_i, \cdots, \pi_n)$. In a matrix game, if the joint strategy satisfies Equation (9), then the strategy is a Nash equilibrium.

$$V_i(\pi_1^*, \cdots, \pi_i^*, \cdots, \pi_n^*) \geq V_i(\pi_1^*, \cdots, \pi_i, \cdots, \pi_n^*) \tag{9}$$

The essence of MARL is a stochastic game. MARL combines the Nash strategies of each state into a strategy for an agent and constantly interacts with the environment to update the Q value function in each state of the game.

The random game consists of a tuple $\left\langle N, S, \left\{A^i\right\}_{i\in\{1,\ldots,N\}}, P, \gamma, \left\{R^i\right\}_{\{1,\ldots,N\}}\right\rangle$, N represents the number of agents, S is the state space of the environment, and A^i is the action space of agent i, P is the probability matrix of state transition, R^i is the reward function of agent i, and γ is the discount factor. For multi-agent reinforcement learning, the goal is to solve the Nash equilibrium strategy in each stage game and combine these strategies.

The optimal strategy of multi-agent reinforcement learning can be written as $\left(\pi_1^*, \cdots, \pi_n^*\right)$ and for $\forall s \in S$, $i = 1, \cdots, n$, it have to satisfy Equation (10).

$$V_i\left(s, \pi_1^*, \cdots, \pi_{i-1}^*, \pi_i^*, \pi_{i+1}^*, \cdots, \pi_n^*\right) \geq V_i\left(s, \pi_1^*, \cdots, \pi_{i-1}^*, \pi_i, \pi_{i+1}^*, \cdots, \pi_n^*\right) \tag{10}$$

$Q_i^*(s, a_1, \cdots, a_n)$ represents the action value function. In each phase game of state s, the Nash equilibrium strategy is solved by using Q_i^* as the reward of the game. According to Bellman's formula in reinforcement learning, MARL's Nash strategy can be rewritten as Equation (11).

$$\sum_{a_1,\cdots,a_n} Q_i^* \pi_1^* \cdots, \pi_{i-1}^*, \pi_i^*, \pi_{i+1}^*, \cdots \pi_n^* \geq \sum_{a_1,\cdots,a_n} Q_i^* \pi_1^* \cdots, \pi_{i-1}^*, \pi_i, \pi_{i+1}^*, \cdots \pi_n^* \tag{11}$$

In a random game, if the reward function of each agent is the same, the game is called complete cooperative game or team game. In order to solve the random game, stage game at each state s needs to be solved, and the reward obtained by taking an action is $Q_i(s)$.

4.3. Path Planning Algorithm Based on IDRL

4.3.1. Environment States

In the cooperative game, UAVs need to choose the optimal strategy according to the state and action of other UAVs. Thus, at timestep k, the state vector of the j-UAV is represented by:

$$s_k^j = \left[x_j, y_j, x_1, y_1, x_{t1}, y_{t1}, a_1, \rho\right]^T \tag{12}$$

where x_j and y_j represent the abscissa and ordinate of the j-UAV, respectively. x_1, y_1 represent the coordinates of the nearest UAV, a_1 represents the action of the nearest UAV at timestep k. x_{t1}, y_{t1} represent the current task coordinates of j-UAV. The value of ρ is 0 or 1, indicating whether the area surrounding the UAV has been searched by other UAVs.

4.3.2. Discrete Action Set

Since the length of the grid in the task environment is much larger than the turning radius of the UAV, it can be assumed that the UAV moves in a straight line in the grid. As shown in Figure 2, when the UAV is in the grid 0, it can perform eight actions to go to the corresponding grid. The numbers in the grids represent eight actions, including: left up, up, right up, left, right, left down, down, and right down.

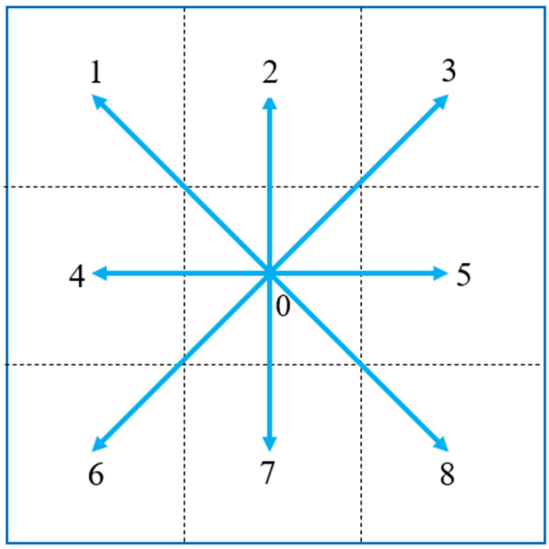


Figure 2. The action set of the UAV. The numbers represent the numbers of the 9 grids respectively.

4.3.3. Reward Function

The reward function is used to evaluate the quality of the action. In fact, there are many factors that could affect the action selection of UAV, but within the scope of research, the following three factors are mainly considered:

- Choosing the shortest path to the destination.
- Encouraging actions passing high ROR areas.
- Preventing collisions between UAVs.

Choosing the shortest path to the target area is not always optimized for path planning, but still has a very high priority in the process. In order to prevent the reward value from being too sparse and speed up the convergence of the IDRL algorithm, a continuous reward function is proposed for the discrete environment. The reward for taking the shortest path is formulated as follows:

$$R_1 = \begin{cases} 40 & if\ end \\ \frac{100}{d_t} * 10^{(-y)} & else \end{cases} \tag{13}$$

In which, y is the integer that increases with the distance of UAV from the target point, d_t is the current Euclidean distance from the UAV to the target point. We set the coordinates of j-UAV as (x_j, y_j), and the coordinates of the target point as (x_2, y_2), then

$$d_t = \sqrt{(x_j - x_2)^2 + (y_j - y_2)^2} \tag{14}$$

In the process of reward value learning, if the reward values of adjacent states are too close, the algorithm may fall into the trap of local optimization due to insufficient training samples. Therefore, for the discovery rate ε, the discovery rate is set to 0.4 to encourage exploration at the beginning of searching for the optimal path. When the algorithm tends to converge, the discovery rate should be reduced to make it approach 0.

The reward function of R_1 is shown in Figure 3. By using the reward function R_1, the UAV can choose the shortest path to the target point according to the reward value obtained.

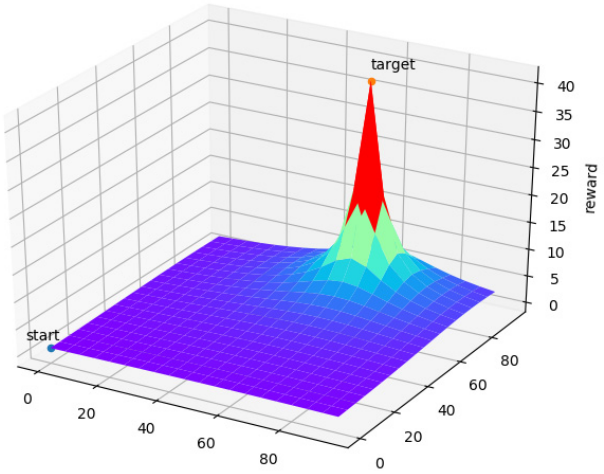


Figure 3. Image of reward function R_1.

When UAVs perform search tasks in the same area without a preset mode of cooperation, different UAVs may detect repeated messages, causing meaningless time loss. In addition, performing search tasks in the same area can easily lead to UAV collisions.

To prevent collisions between UAVs, we add a small penalty when the distance between two UAVs is less than $(2 * \sqrt{2} * d_g)$ in length.

$$R_2 = \begin{cases} -e^{-100 d_u}, & d_u \leq 2 * \sqrt{2} * d_g \\ 0, & d_u > 2 * \sqrt{2} * d_g \\ -1, & d_u = 0 \end{cases} \tag{15}$$

where d_u is the minimum distance between the i-th UAV and the nearest UAV. d_g is the length of the grid in the task model.

When an UAV flies to the assigned target area, the UAV needs to choose a reasonable path. Specifically, UAVs need to make decisions before moving to target areas. As shown in Figure 4, r_1 is the shortest path for the UAV to the target area. If the path is always the shortest route, the UAV will sometimes miss the target grids with high ROR values. Compared with the path r_1, the path r_2 is a more reasonable path. In order to improve the efficiency of UAVs to perform search tasks, the reward function needs to guide the UAV to the target point while passing through the high ROR area on the way. Thus, the reward function R_3 is related to the ROR value of each grid. The combination of reward functions R_1 and R_3 is shown in Figure 5.

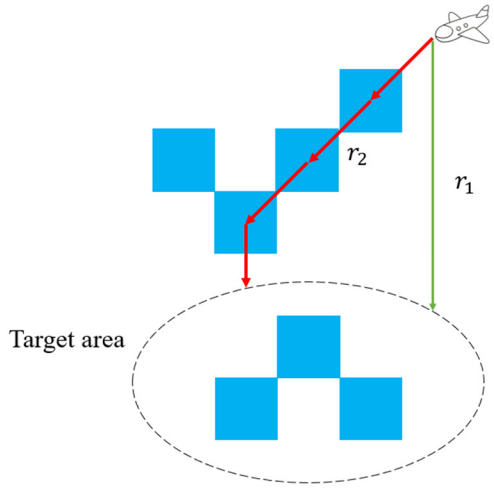


Figure 4. Path planning.

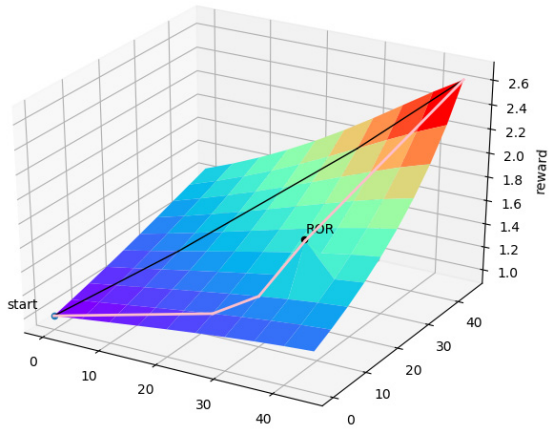


Figure 5. The combination of reward functions R_1 and R_3.

When the reward function R_1 is used to train the UAV, the UAV will choose the straight path. When R_1 is combined with R_3, the UAV will choose the detour path and will not fall into the local optimum.

However, when a task area has been searched by UAV, it will waste time for other UAVs to search this area again, so it is more reasonable to choose path r_1. Therefore, UAV needs to decide which path to choose according to the following formula:

$$R_3 = \begin{cases} 0, & if\ \rho = 1 \\ ROR, & if\ \rho = 0 \end{cases} \tag{16}$$

Therefore, the final reward value function is the sum of the reward values of all parts, each part of the reward value multiplied by an appropriate coefficient.

$$R_{total} = \sum_{i=1}^{3} R_i * k_i \tag{17}$$

where k_i is the coefficient for rewarding of each reward.

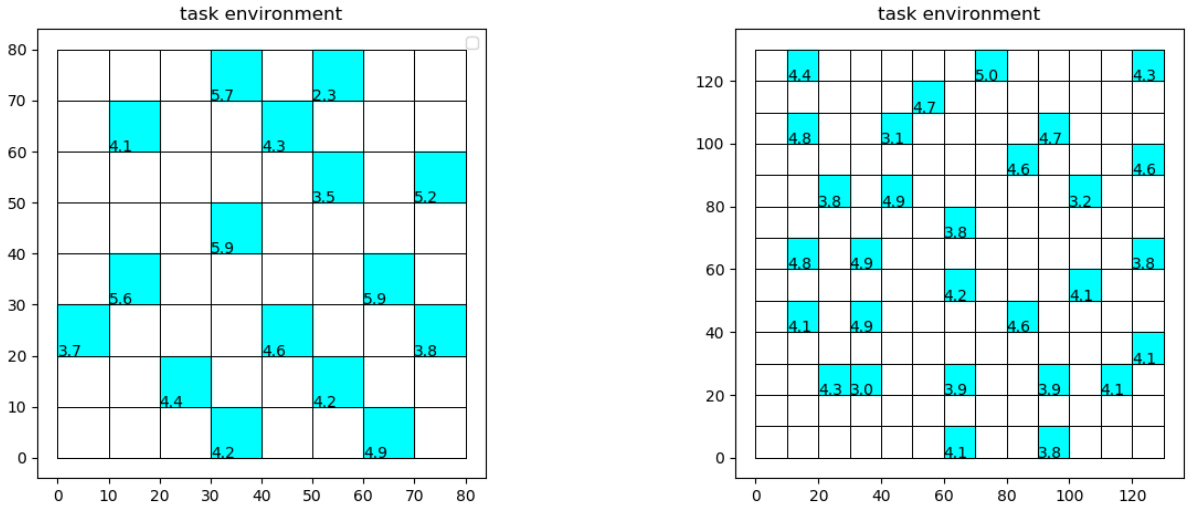
These coefficients represent the proportion of importance of each reward, which can be different between UAVs. Variation of these coefficients could alternate the output results. For example, getting more rewards can use a high value for the coefficient of R_3, while avoiding collisions that can use a low value for the coefficient of R_2.

5. Experiments and Discussions

In this section, we build two simulated search task environments, with 16 points and 29 points, respectively. Experiments are carried out using the above method as well as other algorithms.

5.1. Simulation Environment

Map (a) in Figure 6A is 80 × 80 (m^2) while Map (b) in Figure 6B is 130 × 130 (m^2). The coordinate and ROR of each task area in two maps are shown in Tables 1 and 2.



(**A**) (**B**)

Figure 6. Task environment of two maps. (**A**) task environment of Map (a). (**B**) task environment of Map (b).

Table 2. Task area coordinates and ROR of Map (b).

No	Coordinate	ROR	No	Coordinate	ROR
0	(65, 5)	4.1	15	(125, 65)	3.8
1	(95, 5)	3.8	16	(65, 75)	3.8
2	(25, 25)	4.3	17	(25.85)	3.8
3	(35, 25)	3.0	18	(45, 85)	4.9
4	(65, 25)	3.9	19	(105, 85)	3.2
5	(95, 25)	3.9	20	(85, 95)	4.6
6	(115, 25)	4.1	21	(125, 95)	4.6
7	(125, 35)	4.1	22	(15, 105)	4.8
8	(15, 45)	4.1	23	(45, 105)	3.1
9	(35, 45)	4.9	24	(95, 105)	4.7
10	(85, 45)	4.6	25	(55, 115)	4.7
11	(65, 55)	4.2	26	(15, 125)	4.4
12	(105, 55)	4.1	27	(75, 125)	5.0
13	(15, 65)	4.8	28	(125, 125)	4.3
14	(35, 65)	4.9			

5.2. Parameters Setting

The parameter Settings of the experiment are shown in Table 3. In the experiment, parameters of CBBA and IDRL need to be set, respectively. For CBBA, the maximum number of tasks each UAV can carry out is 9. There is no termination time for each task, and the end condition of the UAV search task in each area is that the ROR of the current task area is less than 0.15 times the initial ROR of the task area.

Table 3. Experimental parameter setting.

Parameters	Values
number of UAVs	4
number of task areas	29
max bundle capacity	9
speed of UAV	4 m/s
mission start time	0
mission end time	$ROR_{current} < 0.15 \times ROR_{initial}$
max episode	2000
discount factor	0.95
learning rate	0.01
reward	R_{total}
number of neurons per layer	100
memory size	500
batch size	30
number of iterations to replace the target	200

For IDRL, the number of iterations of UAV is 20,000 times. When each UAV completes its task, it stops moving and communicating.

5.3. Results and Discussions

We first compare our proposed algorithm with *k*-means algorithm and minimum spanning tree algorithm in the same simulation environment.

In terms of task allocation, the results of using the clustering algorithm on two maps are shown in Figure 7.

Compared with CBBA, the advantage of using clustering algorithm for task allocation is that the task areas of each UAV are concentrated, and the UAV will not collide with other UAVs during flight. However, the dispersed task area makes it difficult to cooperate among multiple UAVs.

The result of using CBBA for task allocation is shown in Figure 8. For CBBA, since CBBA is essentially an auction algorithm, each UAV chooses tasks with the goal of maximizing rewards. Compared with clustering algorithm, the task area assigned by CBBA is more dispersed. At the same time, due to the consideration of time constraints, multiple UAVs can complete the tasks around the same time. The task completion time of each UAV using CBBA algorithm is shown in Table 4.

However, the disadvantage of using CBBA for path planning is that UAVs are prone to collision and crash, and the cooperation between UAVs is not considered in CBBA. IDRL overcomes the above shortcomings. As shown in Figure 9, collisions between UAVs can be avoided by using IDRL and UAVs can choose the path with higher rewards.

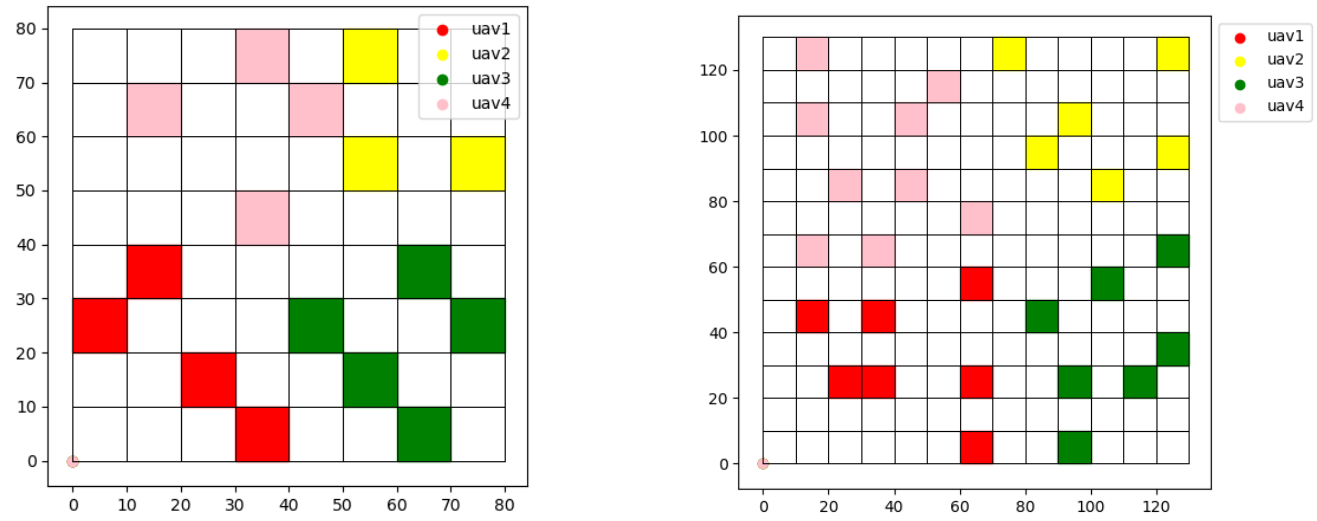


Figure 7. Task assignment using clustering algorithm on two maps. (**A**) Map (a). (**B**) Map (b).

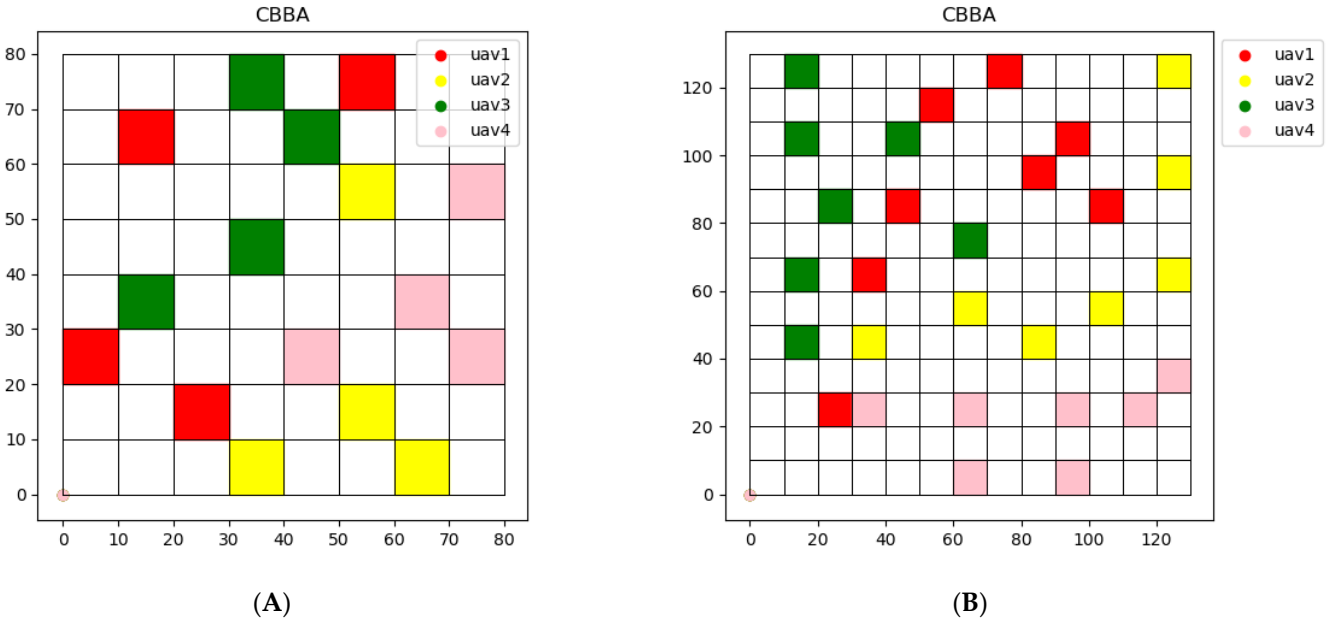
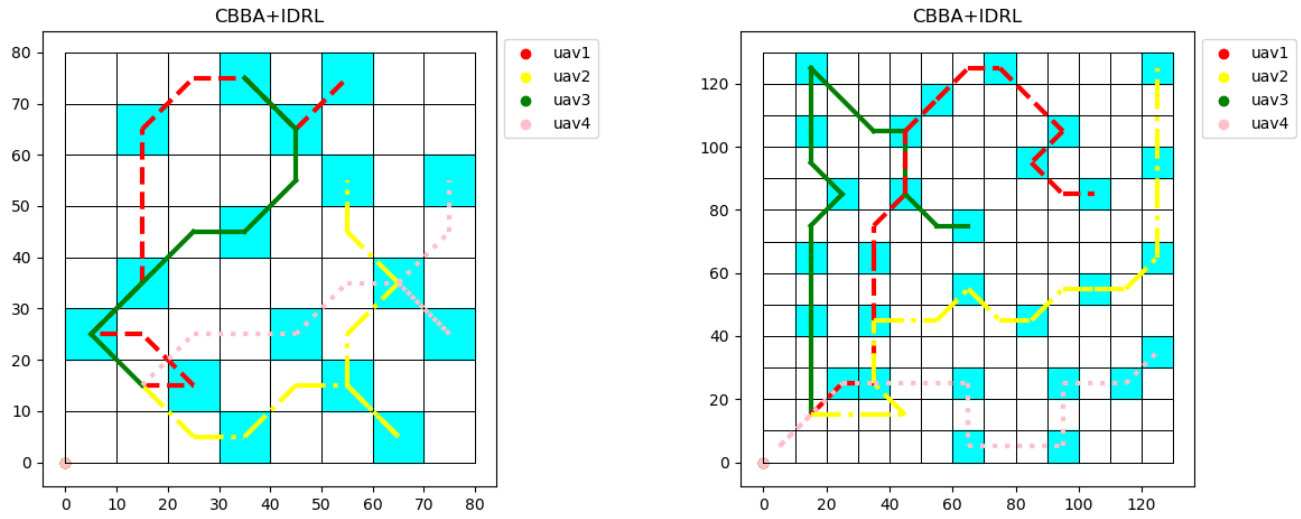


Figure 8. Task assignment using CBBA on two maps. (**A**) Map (a). (**B**) Map (b).

Table 4. CBBA task assignment results.

Type of Map	No	Bundle List	Path List	End Time
Map (a)	UAV1	[2, 5, 15, 10, 14]	[2, 5, 15, 10, 14]	31.430
Map (a)	UAV2	[7, 4, 11]	[7, 4, 11]	23.227
Map (a)	UAV3	[0, 3, 1, 6]	[0, 3, 1, 6]	23.942
Map (a)	UAV4	[8, 13, 12, 9]	[8, 13, 12, 9]	28.168
Map (b)	UAV1	[2, 14, 18, 25, 27, 24, 20, 19]	[2, 14, 18, 25, 27, 24, 20, 19]	39.301
Map (b)	UAV2	[9, 11, 10, 12, 15, 21, 28]	[9, 11, 10, 12, 15, 21, 28]	38.952
Map (b)	UAV3	[8, 13, 17, 22, 26, 23, 16]	[8, 13, 17, 22, 26, 23, 16]	37.366
Map (b)	UAV4	[3, 4, 0, 1, 5, 6, 7]	[3, 4, 0, 1, 5, 6, 7]	31.301



(**A**) (**B**)

Figure 9. Paths generated using CBBA and IDRL. (**A**) Map (a). (**B**) Map (b).

The reward value curves of UAVs in the training process are shown in Figure 10. The reward curve of the UAV in the training process represents the convergence of the algorithm. For map (b), with a more complex task model, as shown in Figure 9, the four agents will undergo a lot of trial and error at the beginning of training. The proposed algorithm is applied to map (b), convergence can be achieved in 1000 iterations, and the collision-free path can be formed eventually.

The changing rules of total revenue of the different algorithms are shown in Figure 11. In two experimental environments, our proposed algorithm can obtain more rewards in the same time period than the K-means+MST algorithm. It is proved that our proposed algorithm has higher search efficiency. In addition, in order to prove that the proposed algorithm can improve the efficiency of search task execution through cooperation, we compared the proposed algorithm with the classical DRL algorithm under the premise that CBBA task assignment is also adopted. The results show that although DRL can get more reward value after completing the task, the time to complete the task is higher than IDRL and the reward value obtained by DRL is lower than IDRL in the same time. It is proved that IDRL can improve the efficiency of task execution through cooperation.

The residual ROR after simulations is shown in Table 5. For CBBA and IDRL, the ROR of most target grids can be reduced to a lower level due to the reasonable path optimization. Though some grids are also fully searched in another algorithm, there are more target grids with high ROR. The results show that our proposed algorithm can search the area more thoroughly.

For a multi-UAV system, the time for each UAV to complete the task needs to be as short and close as possible. We compare the time for each UAV to complete the task between the two algorithms. As shown in Figure 12a, in small-scale scenarios, the proposed algorithm is close to the MST method in terms of time variance and mean value. However, in large-scale scenarios, our proposed algorithm shows obvious advantages, compared with k-means and MST algorithms, and the average time to complete the task with IDRL and the variance of the time to complete the task with four UAVs are smaller.

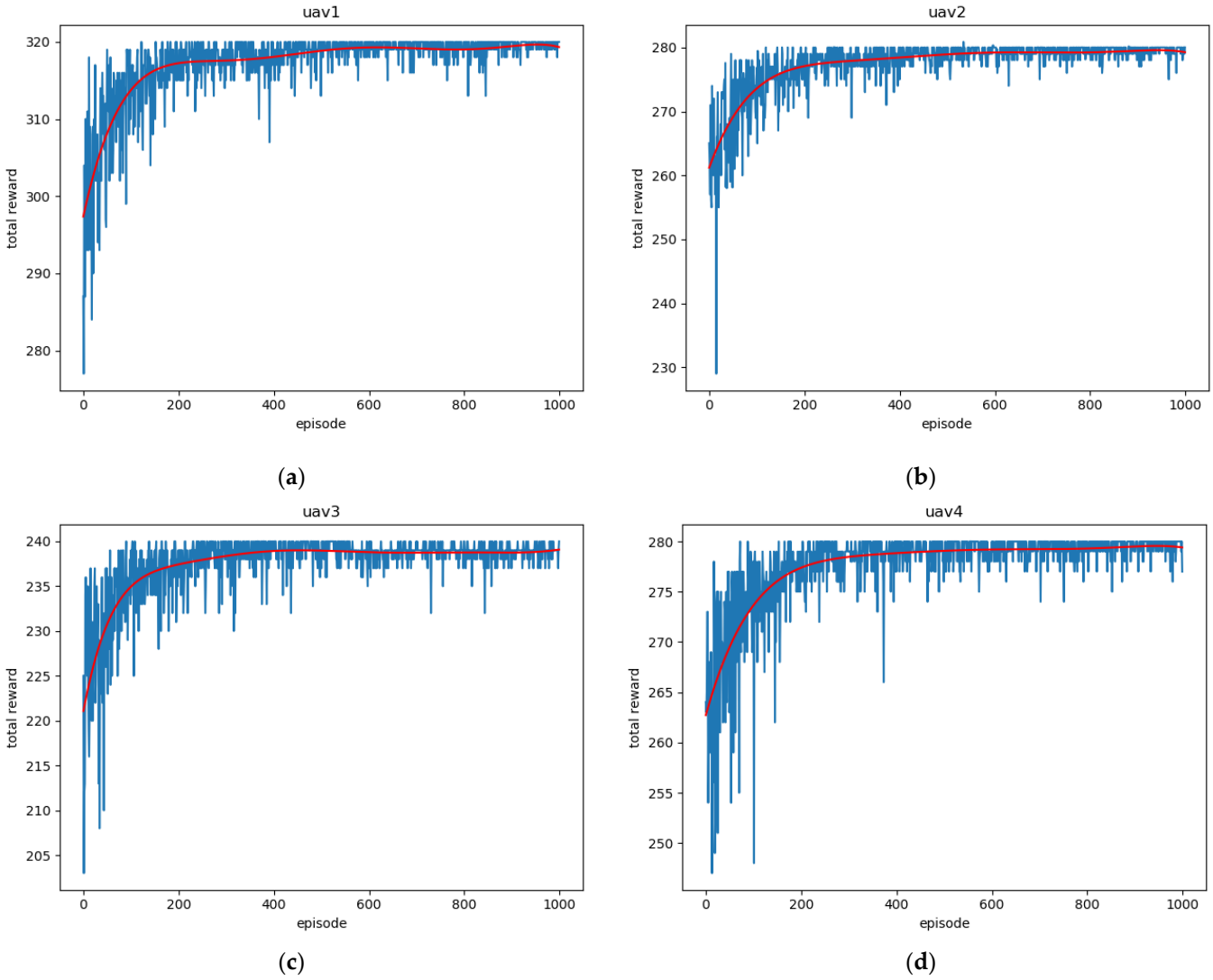


Figure 10. Reward value curves of four UAVs during training: (**a**) UAV1; (**b**) UAV2; (**c**) UAV3 (**d**) UAV4.

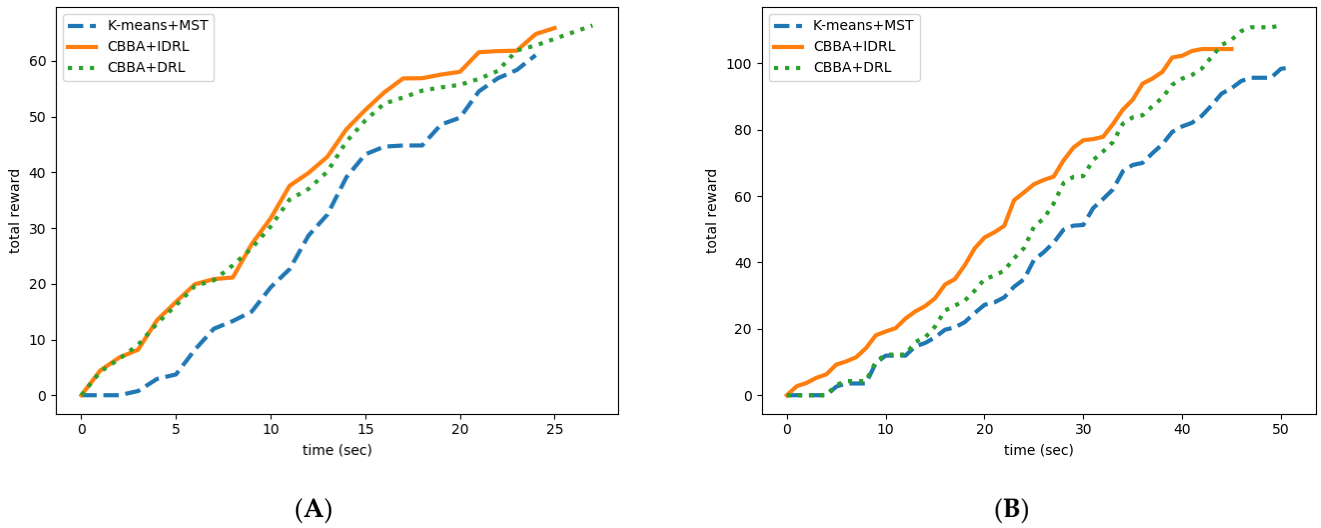
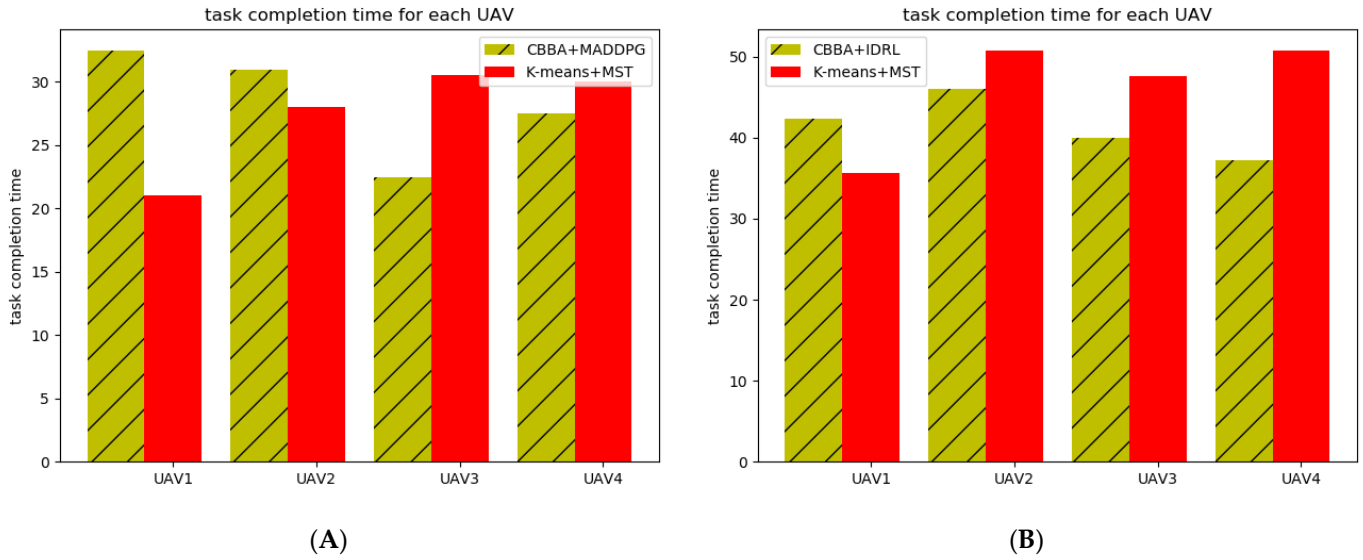


Figure 11. Total revenue variation in different algorithms. (**A**) Map (a). (**B**) Map (b).

Table 5. Task area coordinates and ROR.

Type of Map	No	CBBA + IDRL	K-Means + MST	No	CBBA + IDRL	K-Means + MST
Map (a)	0	0.3810	0.3810	8	0.4382	0.3587
Map (a)	1	0.2439	0.4445	9	1.2875	0.3175
Map (a)	2	0.3991	0.2675	10	1.9129	0.3862
Map (a)	3	0.0422	0.2554	11	0.3719	0.2493
Map (a)	4	0.0204	0.3356	12	0.0237	0.3901
Map (a)	5	0.4173	0.4173	13	0.1275	2.0969
Map (a)	6	0.2310	0.3447	14	0.4643	1.0334
Map (a)	7	0.0301	2.0601	15	0.0019	2.1704
Map (b)	0	0.5549	0.5549	15	0.6281	0.7672
Map (b)	1	0.5142	0.344	16	1.3979	0.2310
Map (b)	2	0.0353	0.096	17	0.2822	0.1548
Map (b)	3	0.0033	0.0819	18	0.0492	0.6631
Map (b)	4	0.5278	0.5278	19	1.1772	0.2902
Map (b)	5	0.5278	0.5278	20	0.2797	0.2797
Map (b)	6	0.3719	0.2493	21	0.6225	0.6225
Map (b)	7	1.0110	0.2493	22	0.6496	0.6496
Map (b)	8	0.5548	0.5548	23	0.0380	0.1885
Map (b)	9	0.0897	0.6631	24	0.7769	0.2858
Map (b)	10	0.7603	0.4173	25	0.2858	0.2858
Map (b)	11	0.3810	1.545	26	0.3991	0.5954
Map (b)	12	0.5548	0.3719	27	0.1118	0.1668
Map (b)	13	0.6496	0.6496	28	1.5818	1.5818
Map (b)	14	0.663	1.803			



(**A**) (**B**)

Figure 12. Task completion time for each UAV. (**A**) Map (a). (**B**) Map (b).

6. Conclusions

This paper first summarizes the existing path planning algorithms and points out their shortcomings. Then the search task model is introduced. On this basis, a cooperative search method of multiple UAVs is proposed. For task points with different reward values, CBBA is first used for task assignment. Then we use IDRL for UAV path planning and propose a new reward function. The proposed reward function consists of three parts, which are respectively used to guide UAV to the target point, avoid collision between UAVs and encourage UAV to choose the path with higher rewards. Experimental results show that compared with the other method, our proposed method can obtain more reward values in the same time and it is feasible and effective for multi-UAV path planning. In our future work, our focus will be on the constraints of the UAV kinematics by integrat-ing the Dubins curve model, which would make the proposed framework more practical.

Author Contributions: Methodology, M.G.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z. and M.G.; visualization, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Parameter	Definition
s	state of UAV
a	action of UAV
R	reward function
d	euclidean distance
ρ	switch for R_3
k	discount factor
π	strategy of the agent
p	action choice probability
y_i	winning score list
z_i	winning agent list
V	state value function
Q	action value function
P	state transition matrix
e	revenue function of the searched area
z	time
ε	search capability of UAV
α	learning rate
b_i	bundle of agent
$c_{ij}[b_i]$	score function
L_t	maximum assigned task number

Abbreviations

The following abbreviations are used in this manuscript:

CBBA	Consensus-based bundle algorithm
UAV	Unmanned aerial vehicle
IDRL	Independent deep reinforcement learning
MARL	Multi-agent reinforcement learning
MUTAPP	Multi-UAV target assignment and path planning
TSP	Traveling salesman problem
MTSP	Multiple traveling salesman problem
GA	Genetic algorithm
OPA	Overall partition algorithm
MST	Minimum spanning tree
MEP	Maximum entropy principle
SA	Simulated annealing
MADDPG	Multi-agent deep deterministic policy gradient
ROR	Rate of return
CPP	Coverage path planning

References

1. Moon, J.; Papaioannou, S.; Laoudias, C.; Kolios, P.; Kim, S. Deep Reinforcement Learning Multi-UAV Trajectory Control for Target Tracking. *IEEE Internet Things J.* **2021**, *8*, 15441–15455. [CrossRef]
2. Yan, C.; Xiang, X. A Path Planning Algorithm for UAV Based on Improved Q-Learning. In Proceedings of the 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS), Wuhan, China, 23–25 June 2018; pp. 1–5.
3. Pei-bei, M.; Zuo-e, F.; Jun, J. Cooperative control of multi-UAV with time constraint in the threat environment. In Proceedings of the 2014 IEEE Chinese Guidance, Navigation and Control Conference, Yantai, China, 8–10 August 2014; pp. 2424–2428.
4. Schouwenaars, T.; How, J.; Feron, E. Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, American Institute of Aeronautics and Astronautics, Providence, RI, USA, 16–19 August 2004.
5. Li, L.; Gu, Q.; Liu, L. Research on Path Planning Algorithm for Multi-UAV Maritime Targets Search Based on Genetic Algorithm. In Proceedings of the 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 26–29 May 2020; pp. 840–843.
6. Zhao, H.; Liu, Q.; Ge, Y.; Kong, R.; Chen, E. Group Preference Aggregation: A Nash Equilibrium Approach. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 679–688.
7. Choi, H.-L.; Brunet, L.; How, J.P. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926. [CrossRef]
8. Chaieb, M.; Jemai, J.; Mellouli, K. A Hierarchical Decomposition Framework for Modeling Combinatorial Optimization Problems. *Procedia Comput. Sci.* **2015**, *60*, 478–487. [CrossRef]
9. Wang, Y.; Cai, W.; Zheng, Y.R. Dubins curves for 3D multi-vehicle path planning using spline interpolation. In Proceedings of the OCEANS 2017-Anchorage, Anchorage, AK, USA, 18–21 September 2017; pp. 1–5.
10. Liu, J.; Zhang, Y.; Wang, X.; Xu, C.; Ma, X. Min-max Path Planning of Multiple UAVs for Autonomous Inspection. In Proceedings of the 2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 21–23 October 2020; pp. 1058–1063.
11. Qingtian, H. Research on Cooperate Search Path Planning of Multiple UAVs Using Dubins Curve. In Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), Bhubaneswar, India, 22–24 January 2021; pp. 584–588.
12. Han, W.; Li, W. Research on Path Planning Problem of Multi-UAV Data Acquisition System for Emergency Scenario. In Proceedings of the 2021 International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA), Huaihua, China, 10–12 December 2021; pp. 210–215.
13. Yaguchi, Y.; Tomeba, T. Region Coverage Flight Path Planning Using Multiple UAVs to Monitor the Huge Areas. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 15–18 June 2021; pp. 1677–1682.
14. Krusniak, M.; James, A.; Flores, A.; Shang, Y. A Multiple UAV Path-Planning Approach to Small Object Counting with Aerial Images. In Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 10–12 January 2021; pp. 1–6.
15. Shao, X.-X.; Gong, Y.-J.; Zhan, Z.-H.; Zhang, J. Bipartite Cooperative Coevolution for Energy-Aware Coverage Path Planning of UAVs. *IEEE Trans. Artif. Intell.* **2022**, *3*, 29–42. [CrossRef]
16. Xie, J.; Carrillo, L.R.G.; Jin, L. Path Planning for UAV to Cover Multiple Separated Convex Polygonal Regions. *IEEE Access* **2020**, *8*, 51770–51785. [CrossRef]

17. Xie, J.; Chen, J. Multiregional Coverage Path Planning for Multiple Energy Constrained UAVs. *IEEE Trans. Intell. Transp. Syst.* **2022**. [CrossRef]
18. Pan, S. UAV Delivery Planning Based on K-Means++ Clustering and Genetic Algorithm. In Proceedings of the 2019 5th International Conference on Control Science and Systems Engineering (ICCSSE), Shanghai, China, 14–16 August 2019; pp. 14–18.
19. Yue, X.; Zhang, W. UAV Path Planning Based on K-Means Algorithm and Simulated Annealing Algorithm. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 2290–2295.
20. Ling, H.; Zhu, T.; He, W.; Zhang, Z.; Luo, H. Cooperative search method for multiple AUVs based on target clustering and path optimization. *Nat. Comput.* **2019**, *20*, 3–10. [CrossRef]
21. Steven, A.; Hertono, G.F.; Handari, B.D. Implementation of clustered ant colony optimization in solving fixed destination multiple depot multiple traveling salesman problem. In Proceedings of the 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 15–16 November 2017; pp. 137–140.
22. Almansoor, M.; Harrath, Y. Big Data Analytics, Greedy Approach, and Clustering Algorithms for Real-Time Cash Management of Automated Teller Machines. In Proceedings of the 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Zallaq, Bahrain, 29–30 September 2021; pp. 631–637.
23. Zhou, S.; Lin, K.-J.; Shih, C.-S. Device clustering for fault monitoring in Internet of Things systems. In Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 14–16 December 2015; pp. 228–233.
24. Trigui, S.; Koubâa, A.; Cheikhrouhou, O.; Qureshi, B.; Youssef, H. A Clustering Market-Based Approach for Multi-robot Emergency Response Applications. In Proceedings of the 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC), Bragana, Portugal, 4–6 May 2016; pp. 137–143.
25. Tang, Y. UAV Detection Based on Clustering Analysis and Improved Genetic Algorithm. In Proceedings of the 2021 International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), Yilan County, Taiwan, 10–12 December 2021; pp. 4–9.
26. Wang, J.; Meng, Q.H. Path Planning for Nonholonomic Multiple Mobile Robot System with Applications to Robotic Autonomous Luggage Trolley Collection at Airports. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 2726–2733.
27. Hassanpour, F.; Akbarzadeh-T, M.-R. Solving a Multi-Traveling Salesmen Problem using a Mamdani Fuzzy Inference Engine and Simulated Annealing Search Algorithm. In Proceedings of the 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 29–30 October 2020; pp. 648–653.
28. Cheng, Z.; Zhao, L.; Shi, Z. Decentralized Multi-UAV Path Planning Based on Two-Layer Coordinative Framework for Formation Rendezvous. *IEEE Access* **2022**, *10*, 45695–45708. [CrossRef]
29. Baranwal, M.; Roehl, B.; Salapaka, S.M. Multiple traveling salesmen and related problems: A maximum-entropy principle based approach. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 3944–3949.
30. Ze-ling, C.; Qi, W.; Ye-qing, Y. Research on Optimization Method of Multi-UAV Collaborative Task Planning. In Proceedings of the 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), Xiamen, China, 10–12 August 2018; pp. 1–6.
31. Bayerlein, H.; Theile, M.; Caccamo, M.; Gesbert, D. Multi-UAV Path Planning for Wireless Data Harvesting with Deep Reinforcement Learning. *IEEE Open J. Commun. Soc.* **2021**, *2*, 1171–1187. [CrossRef]
32. Wang, Z.; Wan, R.; Gui, X.; Zhou, G. Deep Reinforcement Learning of Cooperative Control with Four Robotic Agents by MADDPG. In Proceedings of the 2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC), Chongqing, China, 6–8 November 2020; pp. 287–290.
33. Miyazaki, K.; Matsunaga, N.; Murata, K. Formation path learning for cooperative transportation of multiple robots using MADDPG. In Proceedings of the 2021 21st International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, 12–15 October 2021; pp. 1619–1623.
34. Chen, W.; Hua, L.; Xu, L.; Zhang, B.; Li, M.; Ma, T.; Chen, Y.-Y. MADDPG Algorithm for Coordinated Welding of Multiple Robots. In Proceedings of the 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 15–17 July 2021; pp. 1–5.
35. Han, Q.; Shi, D.; Shen, T.; Xu, X.; Li, Y.; Wang, L. Joint Optimization of Multi-UAV Target Assignment and Path Planning Based on Multi-Agent Reinforcement Learning. *IEEE Access* **2019**, *7*, 146264–146272.
36. Baum, M.; Passino, K. A Search-Theoretic Approach to Cooperative Control for Uninhabited Air Vehicles. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA, USA, 5–8 August 2002.