symmetry

MDPI

Article

Multi-UAV Cooperative Task Assignment Based on Half Random Q-Learning

Pengxing Zhu and Xi Fang *

School of Science, Wuhan University of Technology, Wuhan 430070, China; zhupengxing@whut.edu.cn
* Correspondence: fangxi@whut.edu.cn

Abstract: Unmanned aerial vehicle (UAV) clusters usually face problems such as complex environments, heterogeneous combat subjects, and realistic interference factors in the course of mission assignment. In order to reduce resource consumption and improve the task execution rate, it is very important to develop a reasonable allocation plan for the tasks. Therefore, this paper constructs a heterogeneous UAV multitask assignment model based on several realistic constraints and proposes an improved half-random Q-learning (HR Q-learning) algorithm. The algorithm is based on the Q-learning algorithm under reinforcement learning, and by changing the way the Q-learning algorithm selects the next action in the process of random exploration, the probability of obtaining an invalid action in the random case is reduced, and the exploration efficiency is improved, thus increasing the possibility of obtaining a better assignment scheme, this also ensures symmetry and synergy in the distribution process of the drones. Simulation experiments show that compared with Q-learning algorithm and other heuristic algorithms, HR Q-learning algorithm can improve the performance of task execution, including the ability to improve the rationality of task assignment, increasing the value of gains by 12.12%, this is equivalent to an average of one drone per mission saved, and higher success rate of task execution. This improvement provides a meaningful attempt for UAV task assignment.

Keywords: task allocation; half-random Q-learning; UAV collaboration; random exploration

Citation: Zhu, P.; Fang, X. Multi-UAV Cooperative Task Assignment Based on Half Random Q-Learning. *Symmetry* **2021**, *13*, 2417. https://doi.org/10.3390/sym13122417

Academic Editors: Deming Le, Ming Li and Alexander Zaslavski

Received: 26 October 2021
Accepted: 8 December 2021
Published: 14 December 2021

1. Introduction

Unmanned aerial vehicles (UAVs) are aircraft that do not need a human pilot on board [1], and it is widely used in military operations to perform missions such as jamming [2], attacks [3], and reconnaissance [4] due to their low cost, no casualties, and flexibility [5]. Both task allocation and pre-mission path planning are the core parts of the operational mission execution process [6], where task allocation refers to finding the most reasonable set of allocation options under the condition of meeting environmental elements and mission requirements, so that the UAV system can achieve the optimal overall efficiency and resource allocation [7]. Due to the complex battlefield environment, often a single UAV does not meet the conditions for mission execution, so multi-UAV cooperation is a common form of UAV operations, and multi-UAV assistance can solve the problems of small single UAV loads, fast power consumption, and inability to perform large-scale missions, with the aim of maximizing operational effectiveness [8].

For UAV collaborative task assignment, it can usually be viewed as a multi-combination optimization problem [9]. The traditional methods for solving this project are mainly mathematical and heuristic methods. When the amount of data is small, mathematical methods can be used to list all combinations and find the optimal solution in the combination, but when the amount of data is large, the time taken to list all combinations is too long and the result is optimal but inefficient. Some scholars use improved heuristic algorithms to optimize the UAV mission planning problem, such as genetic algorithm [10], particle swarm algorithm [11], and ant colony optimization algorithm [12] can be selected

randomly and then gradually converge to the optimal method to get a near-optimal solution [13]. Reinforcement learning, a computational approach to learning from interaction, has been applied to many disciplines in the past few years [14]. Reinforcement learning seeks optimal solutions by continuous trial-and-error learning in an unknown environment. Although neural networks have gradually become a research hotspot in recent years, and they can solve the problem of large data volume in continuous space problems in reinforcement learning, but their time-consuming problems are also more prominent. Whether it is a heuristic algorithm or a reinforcement learning algorithm, there is the problem of randomly selecting a strategy in the process of solving the model [15,16]. The existence of a random strategy on the one hand avoids the possibility of the algorithm falling into a local optimum [17], and on the other hand too much invalid random exploration will reduce the execution efficiency of the algorithm. In the actual combat environment, the battle situation is often more urgent, so it is very important to make the optimal decision in the most limited time possible. Q-learning algorithms are widely used for reinforcement learning due to their simplicity and convergence [18].

Therefore, based on reinforcement learning, this paper proposes an improved Q-learning algorithm. This algorithm can get a better allocation result than Q-learning algorithm in almost the same time, which reduces resource consumption and expands actual benefits. The point of random exploration is to accept a poorer action in the present in order to maximize future gains [19]. In this paper, the specific approach to the improvement of the Q-learning algorithm is to allow the acceptance of poorer actions in the random exploration of the next action, but to remove the probability of selecting an action that will always yield poor returns in the future, thus increasing the efficiency of the exploration and making it more likely that a better solution will be obtained. The main contributions of this paper are as follows:

1. A collaborative UAV mission allocation model is constructed, which takes into account the types of UAVs and the resource requirements of the targets, and each type of UAV has different mission execution capabilities; the final evaluation takes into account a series of factors such as reward value, cost value, and adaptability when multiple UAVs collaborate to execute a mission, and integrates all factors and their weights to evaluate the rationality of the allocation scheme.

2. An improved reinforcement learning algorithm HR Q-learning algorithm (HR Q-learning) is proposed, which is based on the characteristics of the model to improve the efficiency of random exploration by judging the invalid action in advance when exploring the strategy randomly, and can obtain the task assignment scheme with higher profit value.

The rest of the paper is organized as follows. In Section 2, a brief review of the literature on UAV tasking assignment is presented. In Section 3, the simulation model and mathematical formulation are given based on the actual battlefield environment. In Section 4, the Q-learning algorithm for reinforcement learning is introduced and an improved algorithm is proposed. In Section 5, experimental parameters are set, simulation experiments are performed, the results are compared and analyzed. Section 6 concludes the whole paper and provides an outlook on future work.

2. Literature Review

In recent years, the task allocation of UAVs based on intelligent algorithms have been extensively studied by many scholars.

Many scholars have conducted modeling and algorithmic studies on the task assignment problem of UAVs. For the study of the algorithm, most of the current research on UAV task assignment is still based on heuristic algorithms and mathematical methods. Some scholars have proposed some new heuristic algorithms based on biological behavior, for example, Kurdi et al. [20] proposes a bacteria-inspired heuristic for the efficient distribution of tasks among the deployed UAVs. The algorithm is based on a simple dynamic energy model of the biological population and combines three allocation strategies to improve the performance of the algorithm. Some scholars have improved the existing

algorithm. Gao et al. [21] proposed a grouped ant colony optimization algorithm, and negative feedback mechanism is introduced to accelerate convergence speed of the algorithm; Bong-Kyun Kim et al. [22] designed a heuristic algorithm to solve the operational planning problem for military aviation. In addition to heuristic algorithms, Ye et al. [23] developed an extended CBBA with task coupling constraints (CBBA-TCC) to solve the multi-task assignment problem with task coupling constraints in the heterogeneous multi-UAV system. In addition, in [24], a new collaborative task assignment method of multi-type UAV based on cross entropy is introduced and the resources required for the problem are considered. Zhou et al. [25] proposed a new two-segment nested scheme generation strategy task planning method based on genetic algorithm and cuckoo search, so as to solve the multi-dimensional and complex problems in the process of UAV action. Heuristic algorithms are mainly characterized by their high randomness and unstable convergence; the main problem of mathematical methods is their long-time consumption. In terms of model, [26] proposed a new dynamic ant colony's labor division (DACLD) model, which enables agents with low intelligence level to perform complex tasks, and this model is highly self-organized and flexible in dynamic environment. In [27], a centralized distributed hybrid control framework for task allocation and scheduling is proposed, and Dynamic Data Driven Application System (DDDAS) is applied to the framework to make it adapt to changing environments and tasks.

Reinforcement learning, as an important branch in the field of machine learning and artificial intelligence, has become an important direction for our research problems, especially in recent years when numerous results beyond human level have been achieved in solving problems such as Atari games and chess game confrontation [28–30]. In the field of UAV task assignment, reinforcement learning algorithms are also gradually gaining more and more attention from scholars. In [31], a fast task assignment algorithm based on Q-learning is proposed. Through neural network approximation and priority experience replay, online learning is transformed into offline learning, and the problem of UAV assignment under uncertain environment is studied. In [32], a double-screening sampling method is designed, which combines deep learning with deep deterministic strategy gradient algorithm to break the correlation of continuous experiments in the experience base and improve the convergence of the algorithm. In [33], based on the complexity and dynamics of the future battlefield, an autonomous decision-making method for UAV is developed by combining the deep belief network decision-making model with genetic algorithm. In UAV collaborative task assignment, most of the improved algorithms based on reinforcement learning are based on the fact that Q-table stored data in reinforcement learning is not easy to be too large and using neural networks to transform it into deep reinforcement learning, but for discrete data processing like task assignment, deep learning is time-consuming and does not take advantage in fast task assignment.

Based on the above studies, it is found that the difficulties of task assignment are mainly in the unacceptability of time during real-time tasks, such as mathematical traversal methods and neural network methods, and the success rate of task assignment and the stability of algorithms, such as heuristic algorithms that easily fall into local optima and have poor robustness in some cases.

3. Multi–UAV Tasking Model Construction

This section is about the model construction in the task assignment process of UAV. This model is based on the premise that the operational capability requirements of each task on the battlefield are known before the operation, or the resources that need to be carried to complete each mission have been reconnoitered during the operation. The purpose is to assign a reasonable combination of heterogeneous UAVs for each task, so that the combination obtained can achieve greater actual benefits and minimum resource consumption. The model considers three different performance indicators of heterogeneous UAVs and missions, combining various factors such as profit, loss, and fitness, and multiple aspects

to measure the advantages and disadvantages of the allocation scheme in a comprehensive manner, which is more realistic.

3.1. Problem Statement

Suppose there are N tasks to be performed on the battlefield, each with three different resource requirements and a reward value for being successfully executed, the set is $T = \{T_1, T_2, \ldots, T_N\}$. Each task $T_i = [\alpha_{Ti}, \beta_{Ti}, \theta_{Ti}, \psi_{Ti}]$ $(T_i \in T)$ can be described by its respective element group. The execution capabilities required to perform the task T_i are $\alpha_{Ti}, \beta_{Ti}, \theta_{Ti}$, respectively. The practical meaning of its representation can be respectively attack capability, defense capability, and jamming capability, etc. ψ_{Ti} represents the revenue that can be obtained by executing task T_i. Moreover, suppose there are M types of heterogeneous UAVs $A = \{A_1, A_2, \ldots, A_M\}$, and the resource vector carried by each type of UAV is $A_j = \left[\alpha_{Aj}, \beta_{Aj}, \theta_{Aj}, \lambda_{Aj}\right] (A_j \in A)$, where α_{Ai}, β_{Aj}, and θ_{Aj} represent the corresponding indicators of attack ability, jamming ability, and defense ability in the task elements. Each type of UAV has three execution capabilities, but each type of UAV has different values of the three capabilities, such as attack UAV may have a high attack capability, but has a low defense capability, so multiple UAVs need to cooperate to perform the mission to guarantee the maximum value of the benefits. λ_{Aj} represents the cost of UAV of type A_j, it was also an important factor in whether to send the drone. It is assumed that the number of UAVs of each type is limited but can meet the task assignment requirements by default, and each mission is performed by up to H UAVs of the same type. Moreover, different types of UAVs take different times to perform different tasks, where T_{ij} represents the time spent for task T_i to be performed by a UAV of type A_j. The UAVs perform their tasks in concert roughly as shown in Figure 1.

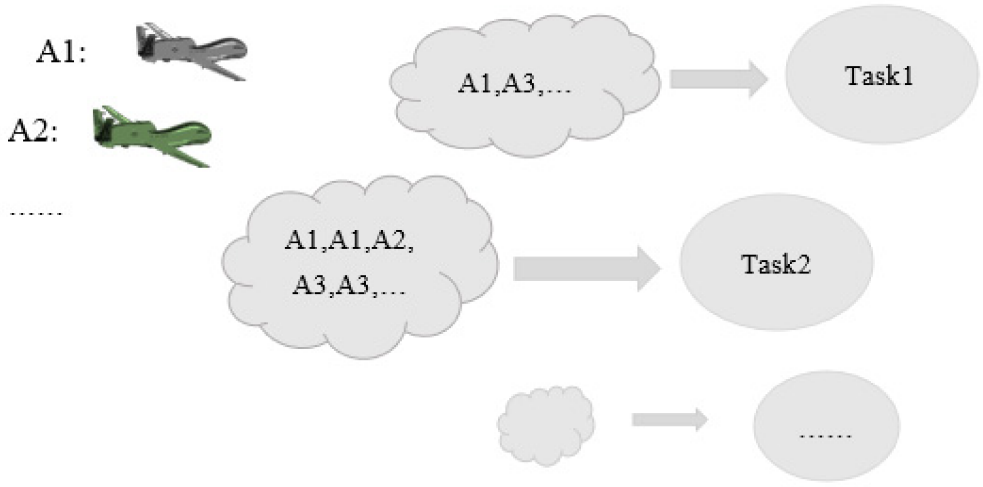


Figure 1. Simulated collaborative task allocation diagram.

3.2. Mathematical Description Based on Resource Constraints

The constraints that should be considered in the modeling of task assignment of UAV cluster include the constraints of UAV's own capability, the constraints of the weapon and ammunition load, and the non-redundant constraints in the cooperative attack process. Because of the different types of UAVs and the different resource capabilities required for each mission to be performed, the combination of UAVs assigned to perform a mission needs to meet the conditions under which the mission can be performed and to guarantee the rational use of resources, which is translated into the following mathematical equation:

(1) When the UAV performs a task, each type of UAV has a different ability to perform the task, a type of UAV can perform the task on the premise that each UAV itself has the basic ability to satisfy the conditions that can perform the task, each task itself has a certain ability to interfere, when the type of UAV can meet the resistance to the interference of the

task, the type of UAV can perform the task without being destroyed due to the jamming, that is:

$$\Delta\theta_{ij} = \theta_{Aj} - \theta_{Ti} \begin{cases} \geq 0 \text{ A}_j \text{ satisfy the condition to execute T}_i \\ < 0 \text{ A}_j \text{ not satisfy the condition to execute T}_i \end{cases} \quad (1)$$

where θ_{Ti} represents the interference capability of task T_i, θ_{Aj} denotes the anti-interference capability possessed by the UAV of type A_j. When the anti-interference ability of the UAV is greater than the interference power of the task, the UAV can perform the task.

(2) Due to the different types of UAVs and the different resources required for the mission, in order to successfully execute the mission, under the condition that the mission can be executed, it is also necessary to ensure that each combat capability of the assigned UAV combination is greater than the combat capability required for the mission, such as for mission T_i, the sum of the attack capability and the sum of the defense capability of the assigned UAV combination that can execute the mission needs to be greater than the mission's required attack capability and defense capability. The formula is as follows:

$$\tau_i = \begin{cases} 1, \sum_n \alpha_n \geq \alpha_{Ti} \text{ and } \sum_n \beta_n \geq \beta_{Ti} \\ 0, \text{ otherwise} \end{cases} \quad (2)$$

where n represents the combination of UAV carrying out mission T_i. For example, the heterogeneous UAV combinations assigned to task T_i are $[\text{A1}, \text{A1}, \text{A2}, \text{A3}, \text{A3}, \text{A3}]$, i.e., three UAVs of type A1, one of type A2, and two of type A3 perform the task T_i together. So $n \in (\text{A1}, \text{A1}, \text{A2}, \text{A3}, \text{A3}, \text{A3})$ and α_{Ti} , β_{Ti} represent the attack capability and ammunition reserve required to perform task T_i. If both equations are satisfied, then $\tau i = 1$, it means that the combination can successfully execute the task, otherwise $\tau i = 0$, it means that the combination cannot successfully execute the task. So $\tau i = 1$ is one of the necessary conditions for the task to be successfully executed.

(3) In order to prevent waste of resources and achieve load balancing when executing a mission, it is necessary to adapt the mission to the assigned heterogeneous UAV combinations as much as possible while ensuring that the mission can be executed. We express the fitness of the assigned UAV combination to the task in terms of the mean squared deviation of its resource vector, that is:

$$\varphi_i = \sqrt{(\sum_n \alpha_n - \alpha_{Ti})^2 + (\sum_n \beta_n - \beta_{Ti})^2} \leq \Delta X \quad (3)$$

where ΔX denotes the maximum mismatch of resources that the allocation scheme can accept. A smaller φ_i indicates a better allocation scheme, and when $\varphi_i > \Delta X$, it indicates that the task is not reasonably executed.

(4) Based on the practical constraints, we also need to consider the size of the number of UAVs that can be assigned to each type for the same task, and in this paper, we set the upper limit of the number of UAVs that can be assigned to the same task for each type of drone as H, i.e.,

$$\sum_j n_{ij} \leq H \quad (4)$$

$\sum_j n_{ij}$ denotes the sum of the number of drones of all types assigned to task T_i.

Finally, all the necessary conditions for successful task execution are shown in the following equation:

$$s.t. \begin{cases} \Delta\theta_{ij} \geq 0 \\ \tau_i = 1 \\ \varphi_i \leq \Delta X \\ \sum_j n_{ij} \leq H \end{cases} \quad (5)$$

3.3. Objective Function

For a certain task, it can be successfully executed only under the premise of satisfying all the above constraints. If the task is successfully executed, the indicators to evaluate the task assignment plan shall include the profit for completing the task, the cost of the UAV,

and the adaptability of the task, etc. So, we set the returns as a linear combination of these indicators and their weights, and our goal is to find the allocation scheme that maximizes the returns, which is

$$\max P_i = \psi_{Ti} - \sum \lambda_{Aj} - \omega_1 \cdot \varphi_i - \omega_2 \cdot n \tag{6}$$

where P_i represents the profit from executing task T_i, which is related to the reward, cost, and fitness etc., and is the result of their weighting, in this paper it is a representative value to measure the merit of the allocation scheme, the aim is to get the maximum profit P_i. $\sum \lambda_{Aj}$ represents the sum of the costs of the assigned heterogeneous UAV combinations, whose costs are not weighted as the actual consumption of resources. φ_i denotes the fitness between the assigned heterogeneous combinations and the task, which indicates the degree of resource wastage. n denotes the number of drones in the assigned combination, and since each UAV dispatched will cause the related loss of human and material resources, it also denotes one of the loss costs. ω_1 and ω_2 are the weights of the fitness and the number of the assigned indicators, and their values are set based on the practical significance.

The model in the literature [31] is based on the setting of constraints and reward values to measure whether satisfactory allocation results are obtained by judging the magnitude of the total reward value. Although the total reward value is finally proven to meet the objective requirements, there is no guarantee that every task assigned has been successfully performed, and does not take benefits and costs into account and does not give a specific allocation plan. In this paper, based on the model constructed in this literature, profit and costs are added and the final solution is evaluated with fitness as one of the elements, this setting makes the model more realistic. We can get the revenue and losses of different distribution schemes, and finally calculate the scheme with the largest profit from all distribution schemes that meet the requirements as the optimal scheme, and give the specific scheme and the corresponding profit value.

4. HR Q-Learning Algorithm

Reinforcement learning (RL) is a method that is not based on environmental solution models, this means that the state of the environment does not need to be known before solving the model. Model-free RL is a powerful and general tool for learning complex behaviors [34]. It is a method by which agents interact with the environment, get feedback from the environment, and then learn the optimal strategy through trial and error [35], the interaction process is shown in Figure 2. Its learning process can be described as a Markov decision process and it is an unsupervised learning method [36]. Reinforcement learning has five elements: the set of states S, the set of actions A, the state transfer probability P, the immediate reward value R, and the policy Π. The five elements specifically include:

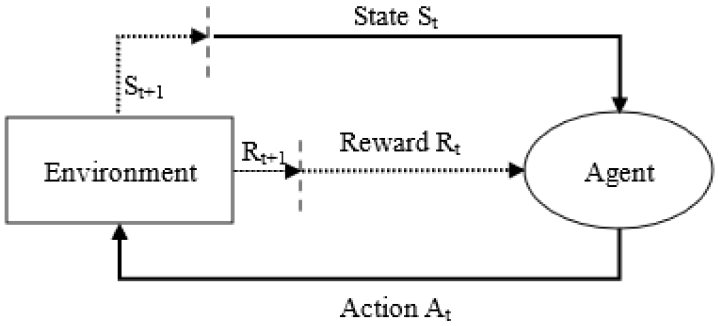


Figure 2. Action and environment interaction process.

State (S): represents the set of states, which refers to the current state of the agent and is a constantly changing quantity.

Action (A): from the action space, which is a description of the behavior of the agent.

State transfer probability $P(s'|s,a)$: denotes the probability of taking action a in state s, and then the environment changes to state s' at the next moment.

Immediate reward $r(s,a,s')$: refers to the reward value that the environment will give back to the agent after the agent makes an action a in state s. This reward value is related to the state s' at the next moment.

Policy: refers to how the agent decides the next action a based on the environment state s. There are generally two types of policies, deterministic policy and stochastic policy. Deterministic policy is a function of mapping from state space to action space, and stochastic policy represents the probability distribution of an agent to choose a certain action given an environmental state. General reinforcement learning uses stochastic policies.

4.1. Q-Learning Algorithm

Q-learning algorithm is a kind of reinforcement learning method based on the updating of value function to constantly update and adjust the strategy [37]. Reinforcement learning has five elements: state set S, action set A, immediate reward value R, decay factor γ, and exploration rate ε, and its goal is to solve the optimal policy $\pi$and the optimal action value function q [38].

The general idea of the Q-learning algorithm is to constantly try in an unknown environment, adjust the strategy according to the feedback information obtained from the attempt, and finally generate a better strategy. According to this strategy, the machine can know what action should be performed in what state. The main part of the algorithm is to build a Q-table with row as state set S and column as actions set A before solving. First initialize the Q-table. Let the initialized states s be the current state, based on the greedy method(ϵ-greedy strategy) of selecting action a on the basis of the current state and acting with the environment. Then get the updated state s' and the reward R from the environment feedback, update the Q-table with the obtained reward values, the updated state s' is used as the current state of the next loop, and judged whether the termination condition is reached, and if not, the loop continues. The goal is to find the expectation of the strategy with the largest cumulative reward.

If the reward sequence $R_{t+1}, R_{t+2}, R_{t+3} \cdots$ is obtained after moment t, in general, we look for the maximum value of the expected reward, and we want the agent to choose a series of actions that maximize the sum of the values of the discounted rewards obtained in the future, that means maximizing the expectation of the discounted reward G_t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \lambda^2 R_{t+3} + \cdots + R_{t+n} = \sum_{k=0}^{n} \lambda^k R_{t+k+1} \tag{7}$$

$$\max_\pi E[\sum_{t=0}^{h} \gamma^t R(S_t, A_t, S_{t+1})|\pi] = \max_\pi E[G_t|\pi) \tag{8}$$

where γ is the discount rate, which caters for $(0 \leq \gamma \leq 1)$, and is the decay factor of future reward value. The Q-learning algorithm uses the time difference method (TD) for learn offline, using the Bellman equation to solve the optimal process, and derives its Q-table to update the process as:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{9}$$

where α is the learning rate, generally taken as a number between $(0,1)$, the larger the learning rate the faster the convergence will be, but it may lead to problems of overfitting.

In the Q-learning algorithm, greedy ε is another important parameter, i.e., when selecting actions based on strategies, the actions are selected with greedy strategies in most cases, while the actions are selected randomly with a certain probability, which is to prevent the algorithm from falling into local optima and increase the exploration degree of the algorithm. A greedy rate that is too small will make the algorithm easily fall into a local optimum, and a greedy rate that is too large will make the algorithm too noisy, causing too much invalid exploration and not easy to converge. The specific process of the Q-learning

algorithm is showed in Figure 3, where E represents the number of iterations, and r stands for reward.

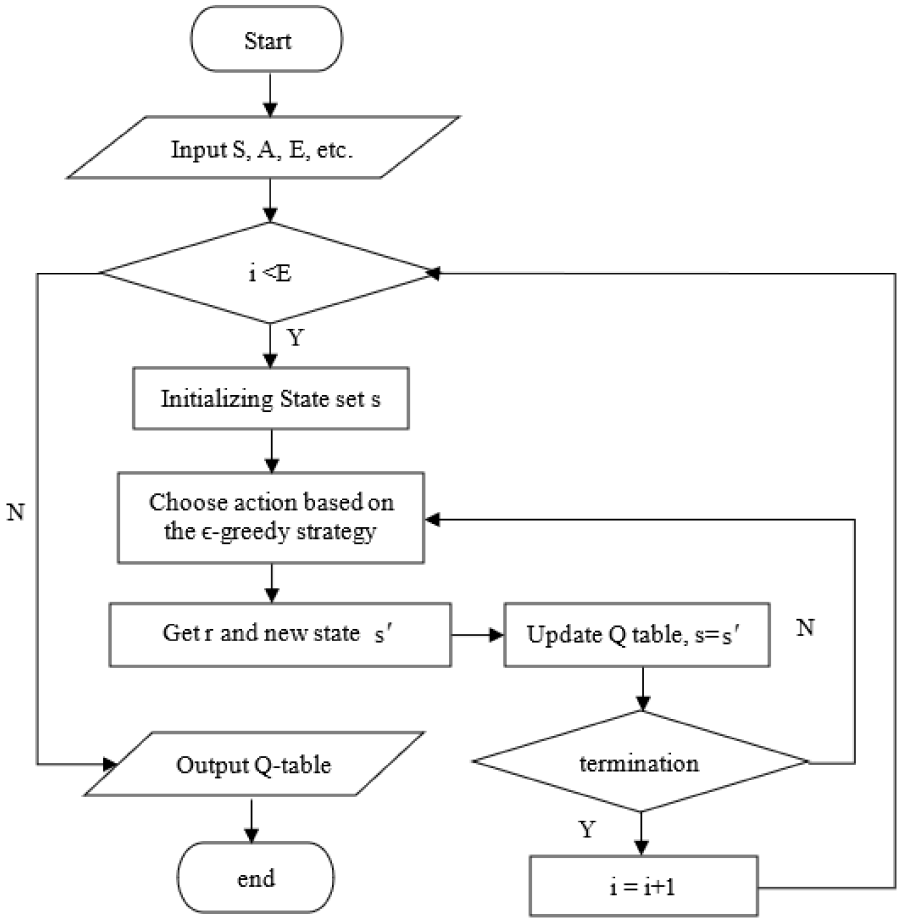


Figure 3. Flow chart of Q-learning algorithm.

4.2. Improved Q-Learning Algorithm–HR Q-Learning

In Section 4.1 we mentioned that the Q-learning algorithm selects actions based on a greedy strategy and there is a greedy rate ε. The meaning of the greedy rate is so that there is a certain probability that the action will be chosen randomly when the agent chooses a strategy based on a greedy strategy, which is to add some noise to the action randomly so that the agent can explore the possibility of achieving a greater reward in the future by choosing a non-optimal action. The equation is as follows:

$$a = \begin{cases} \text{argmax}_a Q(s,a), & \text{with probability } 1-\varepsilon \\ \text{random from A}, & \text{otherwise} \end{cases} \tag{10}$$

Random selection of actions under a certain probability instead of selecting the optimal strategy based on the greedy method may result in poor returns in the current iteration, but in the long run, a better solution may be explored.

According to the model constructed in Section 3, we can find that the results obtained are always poor when the UAV's interference immunity is less than the task's interference capability. So, this paper proposes an improved evolutionary Q-learning algorithm with a half-random selection strategy based on standard Q-learning, and combined with the task assignment model constructed in Section 3. In order to reduce the invalid exploration and make the results converge faster, the HR Q-learning algorithm makes improvements in the process of randomly selecting actions. In the randomly selected strategy, the action is selected according to the greedy strategy, the reward value of the action is obtained, and

it is judged whether it is an action that is always poor in the future, and if so, the action is removed from the randomly selected action in the future. This means that when an agent randomly chooses an action with a certain probability, the result of the exploration will not be the action that has the worst effect and will not achieve a good reward in the future. Improved departments such as Algorithm 1, and the other parts are basically the same as the Q-learning algorithm.

Algorithm 1: Half-Random Exploration

Input: Current State S, exploration rate ε, Action set A
Output: action a
a) Check if status S is in the q-table, if not then add
b) If k $\geq 1 - \varepsilon$ (k is a random number generated between 0 and 1)
Randomly choose a in the action set A
Else:
Greedy strategy choose *a* based on Q-table
c) Execute a, get R and s'
d) If a is worst action
A = A remove a
e) Update action set A
f) Determine if termination is reached, and if not, execute a)

Since the improvement of this algorithm makes the Q-learning algorithm less completely random in the process of random exploration, which is to eliminate the actions with very poor results from the action set in advance by judging them in advance, so we call this algorithm a Half-Random Q-learning (HR Q-learning) algorithm. In terms of the performance of the algorithm this improvement reduces the noise, reduces the invalid random exploration, and allows the algorithm to perform the random strategy efficiently while avoiding getting trapped in a local optimum. For the collaborative task assignment model constructed in this paper, an invalid random exploration leads to the termination of the loop for that round, leaving the round without a satisfactory solution and moving to the next round, reducing the likelihood of the algorithm achieving more reasonable solutions for the same number of iterations, and improving the algorithm increases the likelihood of the algorithm achieving a reasonable solution in each round, which also corresponds to an increase in the probability of obtaining a better drone assignment solution.

4.3. Combining Algorithms with Models

In the model in Section 3, we give the parameters of the task, the parameters of the UAV, the constraints, and the objective function. In order for the algorithm to solve the model, we view the process of solving the model as a Markov decision process according to what was mentioned earlier, and set the states, actions, and rewards in the model.

The initial parameters of the task are the initial set of states, the dimension of the set of states is 4, the drone types are action sets, and if there are N types of drones, the action set dimension is N and each action $\mathrm{a}(\mathrm{a} \in A)$ also has four parameter indices. For each action selection, first judge whether the jamming resistance of the selected drone is greater than the jamming capability of the task, judge whether the task can be executed, and if it is not satisfied, the current round ends, otherwise continue to judge whether other constraints are satisfied. Reinforcement learning is an algorithm that updates the selection policy in a single step, so the next state s' is the parameter set of the current state set minus the corresponding parameter set of the selection action, until the cumulative assigned UAV combinations reach the condition that the task can be executed, and judge whether the constraint of fitness is satisfied, if it is satisfied, it means that the task is successfully executed and the round is over, and the round assignment scheme is recorded in the list. The judgment process is shown in Figure 4.

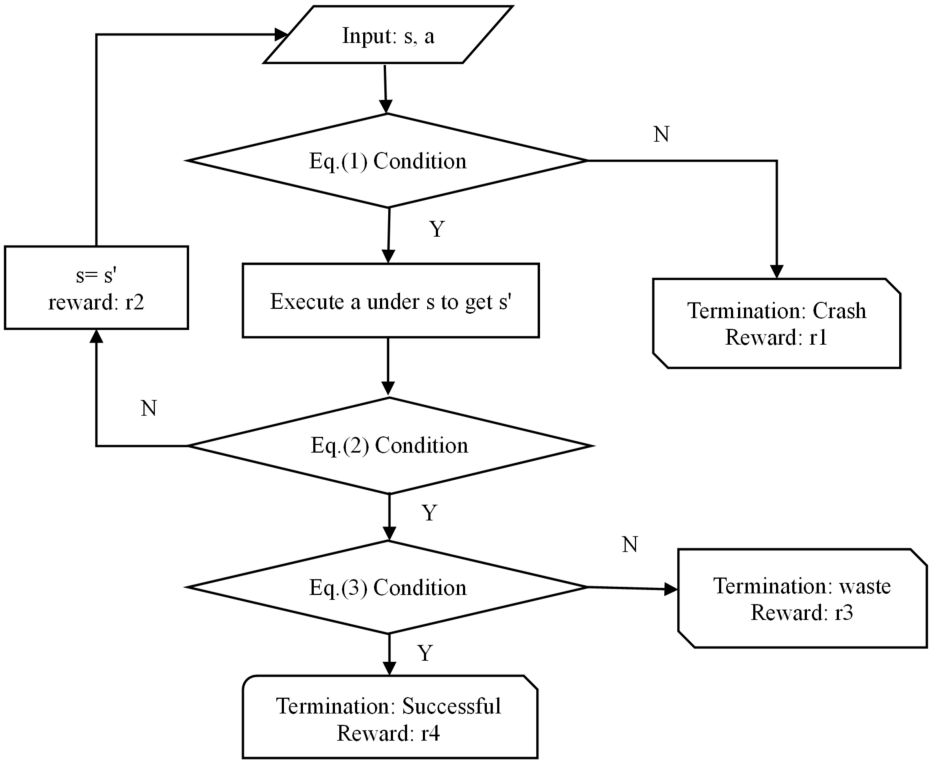


Figure 4. Termination condition judgment.

Depending on the state s' the agent is given a different reward value r. This reward value is used to update the Q-table and to guide the selection of the agent's action for the next round. The process is the part of reward value acquisition and termination condition determination, which is also the core part of applying the model constructed in Section 3 to the reinforcement learning algorithm. Algorithm 2 shows the specific process.

Algorithm 2: HR Q-Learning Algorithm Implementation of Task Assignment

Input: Action set A, initial State S, learning rate α, exploration rate ε, etc.
Output: Allocation scheme, Profit value
1) Initialize Q-table, State S
2) For j from 1 to T
   S = S (T_j)
   Initialize A
3) For i in range E:
   a) Initialization state s
   b) Choose the a from the A based on the ε-greedy strategy
   c) Execute a, get R and S′
   d) If a is worst actio
      A = A remove a
   e) Update action set A
   f) Equation (9) update the value function Q (S, A)
      $Q_{New}(S, A) = Q_{Old}(S, A) + \alpha(R + \gamma max_{A'} Q(S', A') - Q_{Old}(S, A))$
   g) s = s′
   h) Determine if termination is reached, and if not, execute b)
   i) if s′ = successful,
      Storage
   j) If the iteration completes, start the next task from 2)
4) When all tasks are assigned and completed
   Calculate the revenue according to Equation (6)
   Collation results

First, the parameters α_{Ti}, β_{Ti} and θ_{Ti} of the task are used as the parameter vectors of the state S, $s = (\alpha_{Ti}, \beta_{Ti}, \theta_{Ti})$ is the initial state; the action set is the type of heterogeneous UAV, which is $A = (A_1, A_2, \cdots A_m)$; the reward r and the next state s' can be obtained by performing action a in state s. The reward value is determined by the state obtained, and since the reward value in turn determines the choice of action, so different rewards should be set for different states. The state vector of the next step is equal to the parameter vector of the current state minus the parameter vector of the selected drone type, where the interference value θ is only compared to determine whether it can be executed, and is not subtracted. When the assignment task completes or fails, it is determined to be terminated, and if s' is terminated, the next loop is started, otherwise the next state s' is used as the current state to continue iteration. The flow chart is shown in Figure 5.

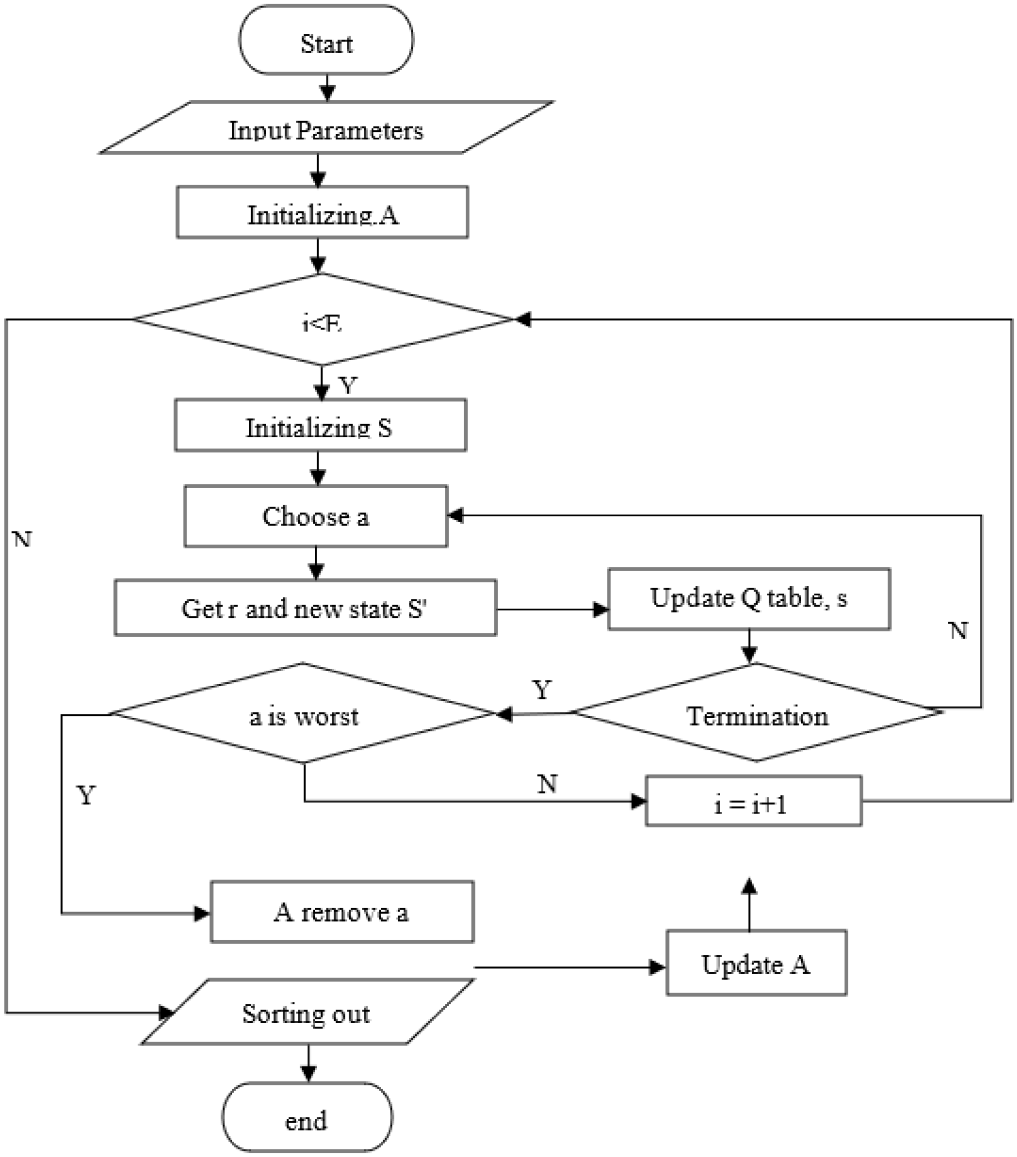


Figure 5. HR Q-learning flow chart.

5. Simulation and Discussion

In order to verify the effectiveness of the algorithm, in this section we will conduct multiple sets of experimental analysis to compare by setting different number of tasks and number of UAV types. Finally, the effectiveness of the algorithm is measured by analyzing the magnitude of the profit value of the resulting allocation scheme and the stability of the

results. The experimental system configuration was AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GH,16.0 GB RAM, the experimental environment is python3.

5.1. Parameter Setting

Now let there be N types of UAVs on the battlefield with different mission execution capabilities and cost values, and now specify 10 types of UAVs whose parameters are shown in Table 1.

Table 1. Resource parameters for UAVs.

UAV Type	Resource Capacity			Cost
	α_A	β_A	θ_A	λ_A
A1	25	30	25	1.5
A2	27	42	15	2.1
A3	18	20	30	1.0
A4	45	25	13	2.0
A5	20	16	30	0.8
A6	18	40	25	1.8
A7	35	20	15	1.2
A8	20	45	20	2.5
A9	18	25	28	0.9
A10	23	15	23	0.8

There are M tasks to be executed, each task requires different combat capabilities to be executed and can be successfully executed with different rewards. The parameter values and reward values are randomly generated in a certain interval, to ensure the universality of the algorithm, so the parameter values of each comparison task are re-generated randomly, but the task parameters are the same in one comparison. We set the ranges of attack capability, defense capability, and jamming capability required for each task as (130, 170), (130,170), (10,30), respectively, the interval of the value of the returns for completing the task is (25,40).

Since reinforcement learning algorithm is based on the value function to solve the model, different reward values should be given to different states to complete the final allocation. The value of reward value R_i for each step of task T_i is determined by the following formula:

$$R_i = \begin{cases} -9, & \text{if } \Delta\theta_{ij} < 0 \\ 0, & \text{if } \Delta\theta_{ij} \leq 0 \text{ and } \tau_i = 1 \text{ and } \varphi_i > \Delta X \\ \psi_{Ti} - \omega_1 \cdot \varphi_i - \omega_2 \cdot 1, & \text{if } \Delta\theta_{ij} \leq 0 \text{ and } \tau_i = 1 \text{ and } \varphi_i \leq \Delta X \\ -1 \cdot \omega_2, & \text{otherwise} \end{cases} \tag{11}$$

where let $\Delta X = 30$, it represents an allocation of the fitness size to waste no more than one drone. The meaning of the above reward value is that when the UAV cannot satisfy the task execution condition, a large negative reward value of -9 is given; when the UAV satisfies the task execution condition but is not reasonable, that is, the fitness is small and a reward of 0 is given. When the task is reasonably performed, the reward value is equal to the value of the revenue of performing the task minus the fitness multiplied by the weight ω_1 and the value of the depletion caused by the assignment is 1 multiplied by the weight ω_2. Based on the actual suitability importance and the actual loss per drone dispatched, we set $\omega_1 = 0.7, \omega_2 = 0.6$.

5.2. Contrast Analysis

5.2.1. Parameter Comparison

The setting of parameters has some influence on the results of reinforcement learning, especially the setting of learning rate and exploration rate. According to the existing experience, the learning rate and exploration rate can be set as constants, and in addition,

they can be set as quantities that vary with the running process in order to make the algorithm converge better. In general, the exploration rate can be set to $1/t$, where t is the number of iterations, implying that at the beginning the algorithm tends to choose actions more randomly, it tends to choose better strategies as the number of iterations increases; the learning rate can be set to $\alpha = 1/(1 + visits(s, a))$, where $visits(s, a)$ is the number of times that we have sampled the state/action pair (s, a), indicating that the learning rate gradually decreases with the number of samples of a state/action pair. We set epsilon equal to 0.15, 0.1,0.05,1/t, respectively, and set multiple sets of experiments to compare the profit value curves of HR Q-learning algorithm and Q-learning algorithm. As shown in Figure 6, t denotes the number of iterations.

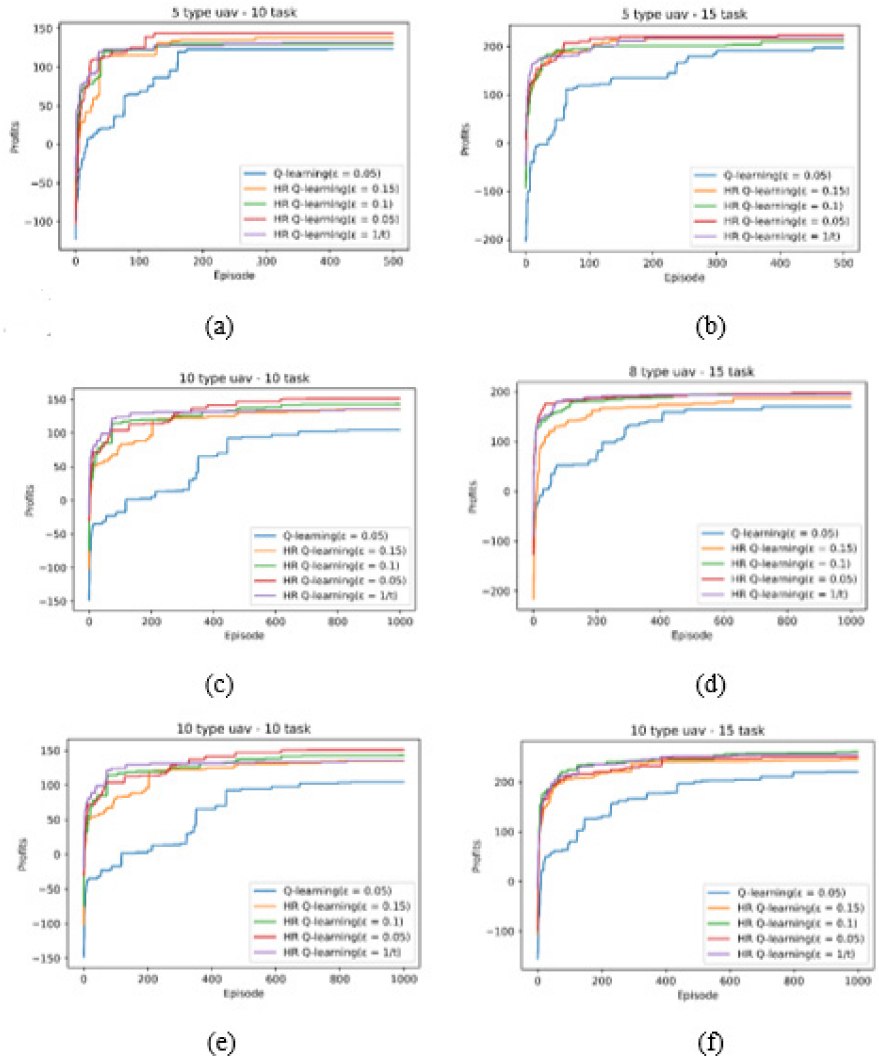


Figure 6. Comparison of different exploration rates (**a**–**f**).

By comparing the curves for different number of UAV types performing different number of tasks with different epsilon, it can be seen that the HR Q-learning algorithm is better than the Q-learning algorithm. For the HR Q-learning algorithm, the gain value at $\varepsilon = 0.05$ is slightly higher than the other curves. The algorithm can converge better, but the profits value does not show an advantage when $\varepsilon = 1/t$.

We continue to compare the performance of the algorithm under different learning rates. Since in the above experiment we found that the algorithm has the best performance when $\varepsilon = 0.05$, we set $\varepsilon = 0.05$, the number of iterations is 1000, and the learning rate is set to be 0.1, 0.05, 0.01, and $1/(1 + visits(s, a))$, respectively, as shown in Figure 7.

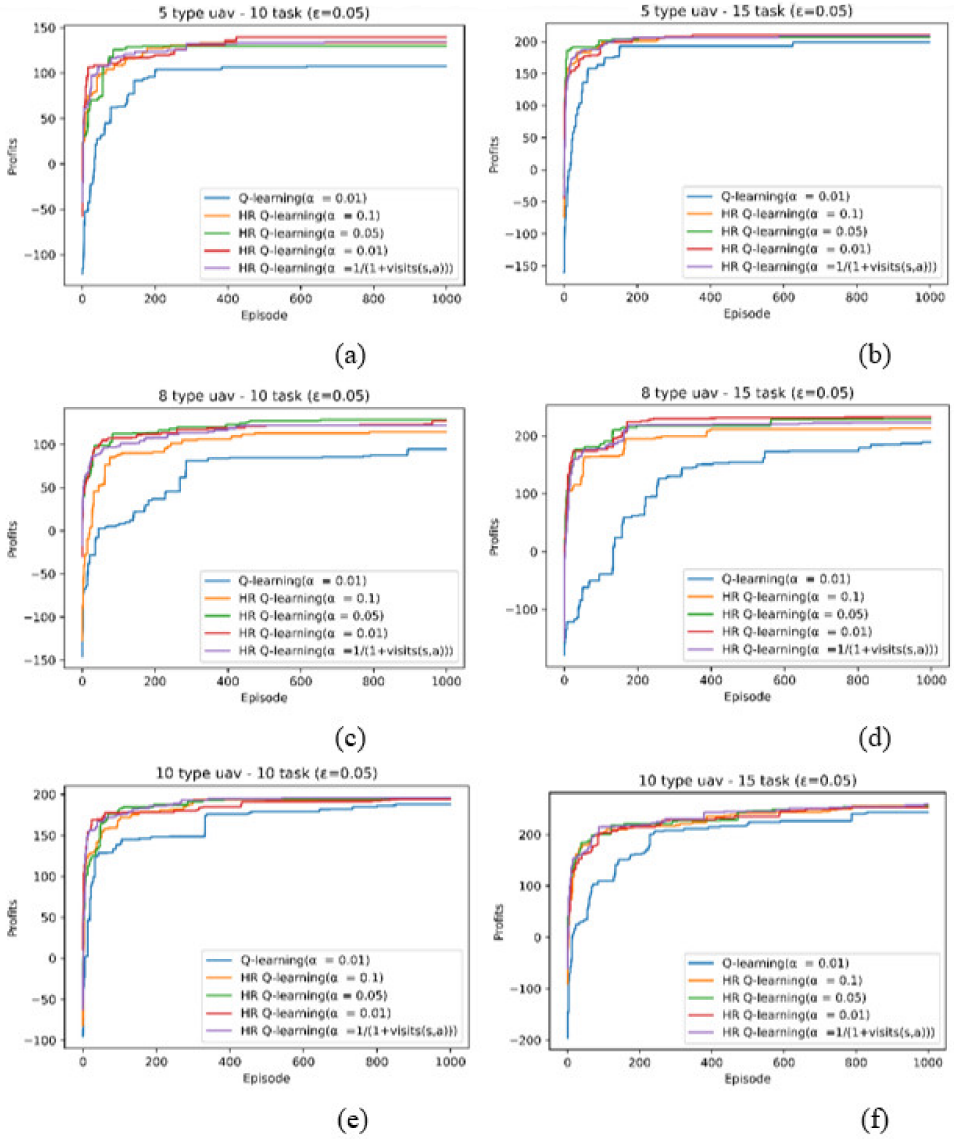


Figure 7. Comparison of different learning rate (**a**–**f**).

It can be seen from Figure 7 that the effect of different learning rates on the return profit curve of the HR Q-learning algorithm is not obvious. When $\alpha = 0.1$, the effect is poor, and the effect is slightly better when $\alpha = 0.01$. Therefore, in the subsequent algorithm experiments, we set $\alpha = 0.01$ and $\varepsilon = 0.05$ for achieving better effectiveness of the algorithm.

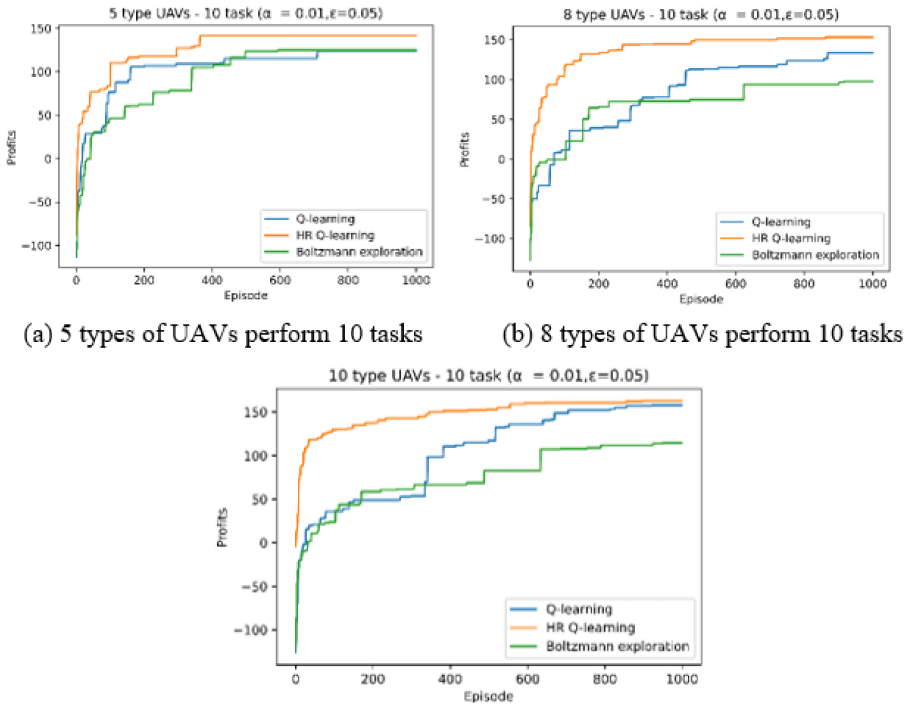
5.2.2. Comparison with Existing Reinforcement Learning Approaches

For the model proposed in the paper, there already exists well-known exploration strategies in reinforcement learning that have been known to show good performance in convergence, and we focus on comparing the proposed HR Q-learning algorithm with the original Q-learning algorithm and Boltzmann exploration method in this section. Boltzmann exploration strategy is more complicated than the greedy strategy. Its general form is

$$P(a_n|s) = \frac{e^{\frac{Q(s,a_n)}{T}}}{\sum_i e^{\frac{Q(s,a_i)}{T}}} \tag{12}$$

In which $P(a_n|s)$ is the probability of selecting action a_n in state s, and T is the temperature parameter, we set $T = 1$.

First, 10 tasks are randomly generated and performed by 5, 8, and 10 of the 10 types of UAVs in Table 1, respectively, and the number of iterations is set to 1000. The profit curve by running the results to obtain the maximum profit value curve of the algorithms is shown in Figure 8.



(a) 5 types of UAVs perform 10 tasks (b) 8 types of UAVs perform 10 tasks

(c) 10 types of UAVs perform 10 tasks

Figure 8. Comparison with Boltzmann exploration strategy (**a**–**c**).

As can be seen from Figure 8, the Boltzmann exploration method does not show good performance in the model proposed in this paper. The HR Q-learning algorithm outperforms both the Q-learning algorithm and the Boltzmann exploration method in terms of profits values and convergence speed.

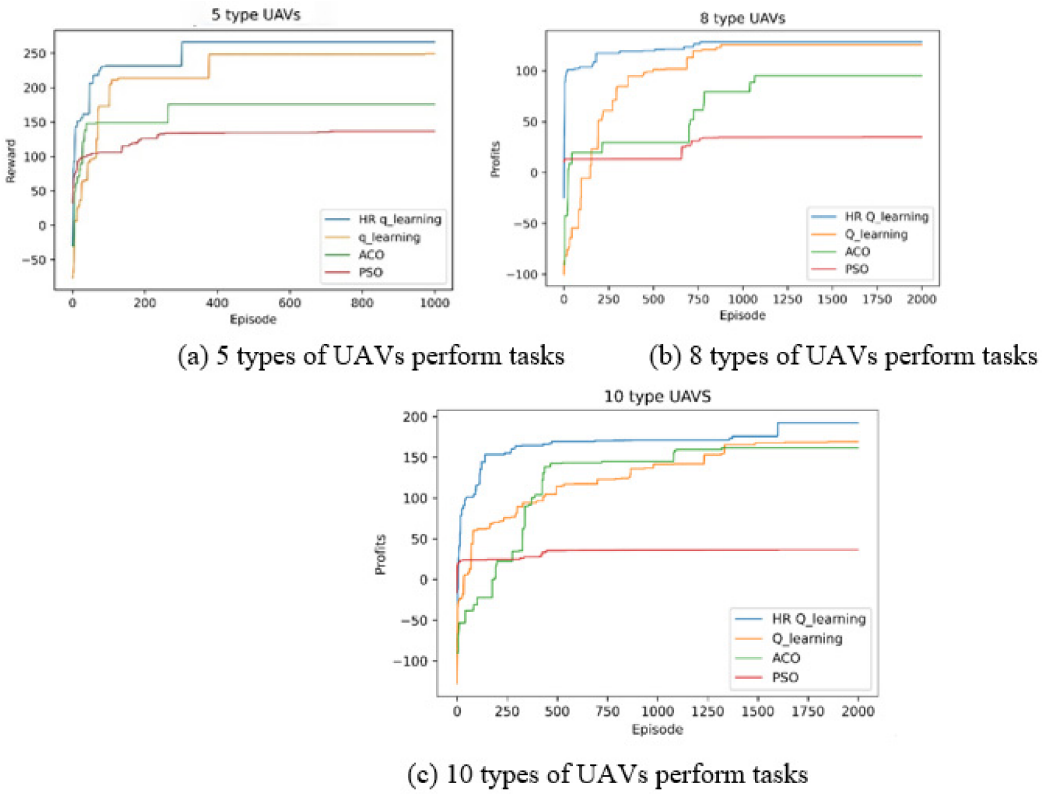
5.2.3. Comparison with Other Algorithms

For the UAV task assignment problem, many scholars also use heuristic algorithms such as ant colony algorithm (ACO) and particle swarm algorithm (PSO) for solving. To better illustrate the effectiveness of the algorithm proposed in this paper, we compare the HR Q-learning algorithm with several unimproved heuristic algorithms, where the parameters of the heuristic algorithm are set to those commonly used in the literature. The parameter settings are shown in Table 2.

Table 2. Algorithm parameter setting.

Parameter	Value(s)
ACO	
Information Heuristic Factor α	2
Expectation Heuristic Factor β	2
Information volatility factor ρ	0.4
Population number m	50
PSO	
Inertia weight ω	0.8
Learning factor C_1, C_2	2
Population number n	1000

Similarly, 10 tasks are randomly generated in the given interval, and given different number of types of UAVs to perform the tasks. Since the four algorithms can be fully converged by 1000 iterations when 5 types of UAVs perform the task, the number of iterations is set to 1000; when 8 and 10 types of UAVs perform the task, the number of iterations is set to 2000 due to the high complexity of the model. The gain value curves of the run results are shown in Figure 9.

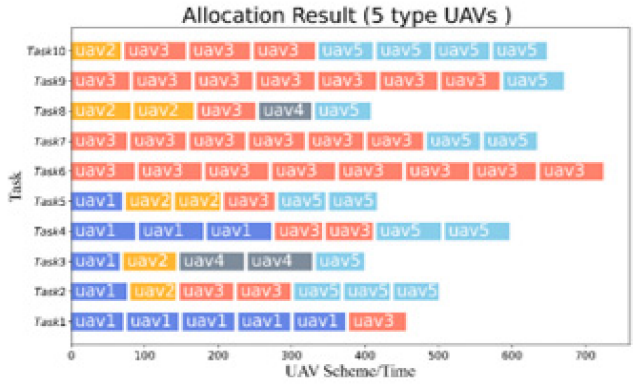


(a) 5 types of UAVs perform tasks (b) 8 types of UAVs perform tasks

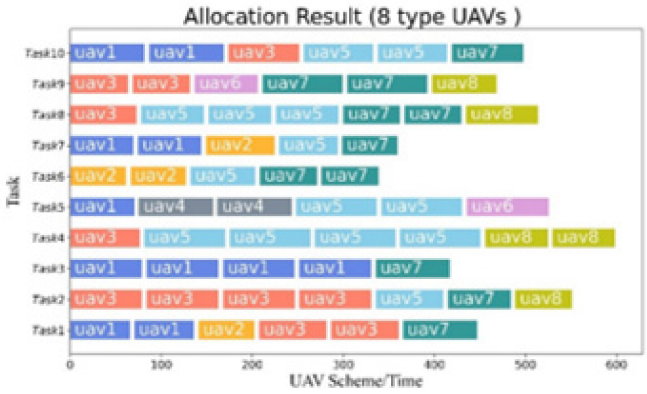
(c) 10 types of UAVs perform tasks

Figure 9. Comparison with heuristic algorithms (**a**–**c**).

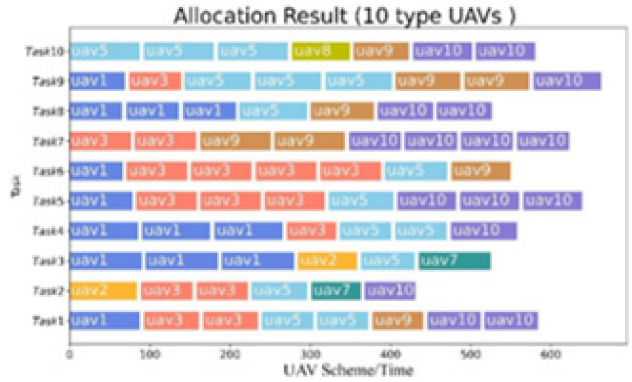
As can be seen from Figure 9, the HR Q-learning algorithm can obtain greater gain values than the original Q-learning, ACO and PSO. The worst results are obtained by the PSO. Through analysis, it is found that the PSO is easy to fall into local optimum in the operation process, and the model constructed in this paper is a discrete model, and the algorithm is not suitable for solving. The ant colony algorithm gives better results when the population size is large, but the system takes longer time, and does not give better solutions than the two reinforcement learning algorithms when the initial population is 50. The Q-learning algorithm has a similar curve to the HR Q-learning algorithm, but the HR Q-learning algorithm converges faster and the profit value is stable than the Q-learning algorithm. This also indicates that the HR Q-learning algorithm reduces the noise after removing the invalid exploration and obtains an increase in the probability of the fitting solution. As the type of UAV increases, the complexity of the algorithm processing increases, but the HR Q-learning algorithm can still obtain better gain values and converge relatively fast compared to the heuristic algorithm and Q-Learning algorithm. The results of the HR Q-learning algorithm for the assignment of ten tasks under different types of number of UAVs are shown in Figure 10.



(a) Results of 5 types of UAV allocation



(b) Results of 8 types of UAV allocation



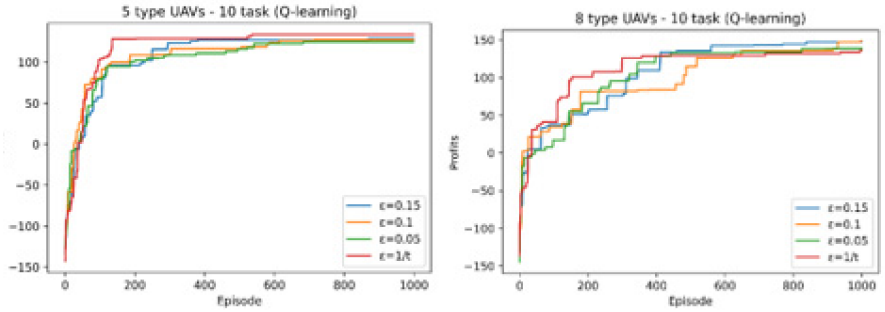
(c) Results of 10 types of UAV allocation

Figure 10. HR Q-learning algorithm distribution results (**a**–**c**).

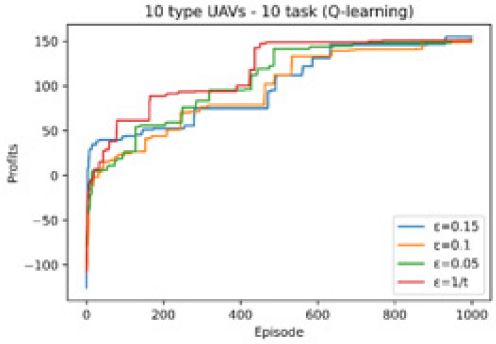
5.3. Multiple Test Comparison

Since reinforcement learning has a probabilistic process, in order to illustrate the effectiveness of the algorithm, we compare the HR Q-learning algorithm with Q-learning by running several repetitions and prove that the HR Q-learning algorithm with can obtain a better allocation scheme by statistical significance.

To ensure the fairness of the experiment, we first set different parameters for the Q-learning algorithm for comparison. Similarly, we set epsilon and alpha as some variables and constants, respectively, and set different number of UAV types to perform the task, and the results are shown in the Figures 11 and 12.

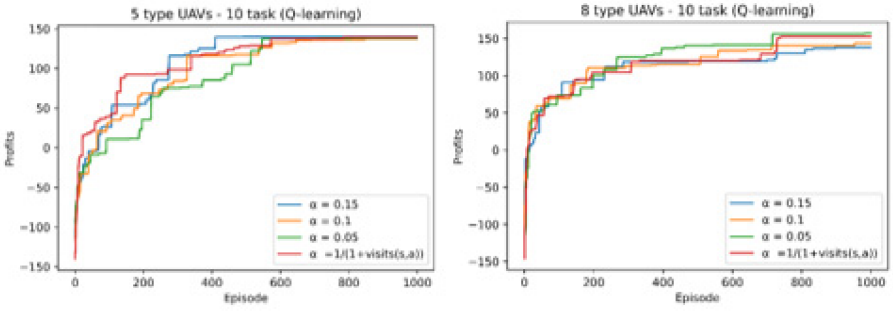


(a) 5 types of UAVs perform tasks (b) 8 types of UAVs perform tasks

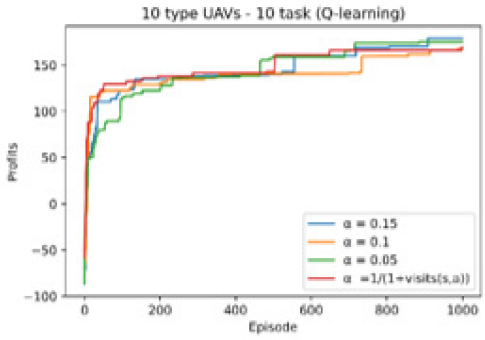


(c) 10 types of UAVs perform tasks

Figure 11. Q-learning algorithm comparison with different epsilon (**a**–**c**).



(a) 5 types of UAVs perform tasks (b) 8 types of UAVs perform tasks



(c) 10 types of UAVs perform tasks

Figure 12. Q-learning algorithm comparison with different alpha (**a**–**c**).

Through experimental comparison, we found that setting different parameters has less effect on the results of Q-learning algorithm. The algorithm converges faster when epsilon 1/t, but the profit value is less different from the other parameters, and the algorithm runs with little difference in results for different alpha. We set the same parameters of Q-

learning algorithm and HR Q-learning algorithm in the experimental, and set the number of repetitive trials to 20.

We ran 6 sets of experiments with 5 types of UAVs performing 10 missions, 5 types of UAVs performing 15 missions, 8 types of UAVs performing 10 missions, 8 types of UAVs performing 15 missions, 10 types of UAVs performing 10 missions, and 10 types of UAVs performing 15 missions, and obtained a comparison of the results of 20 runs of the 6 sets of experiments as shown in Figure 13, the result is retained as an integer.

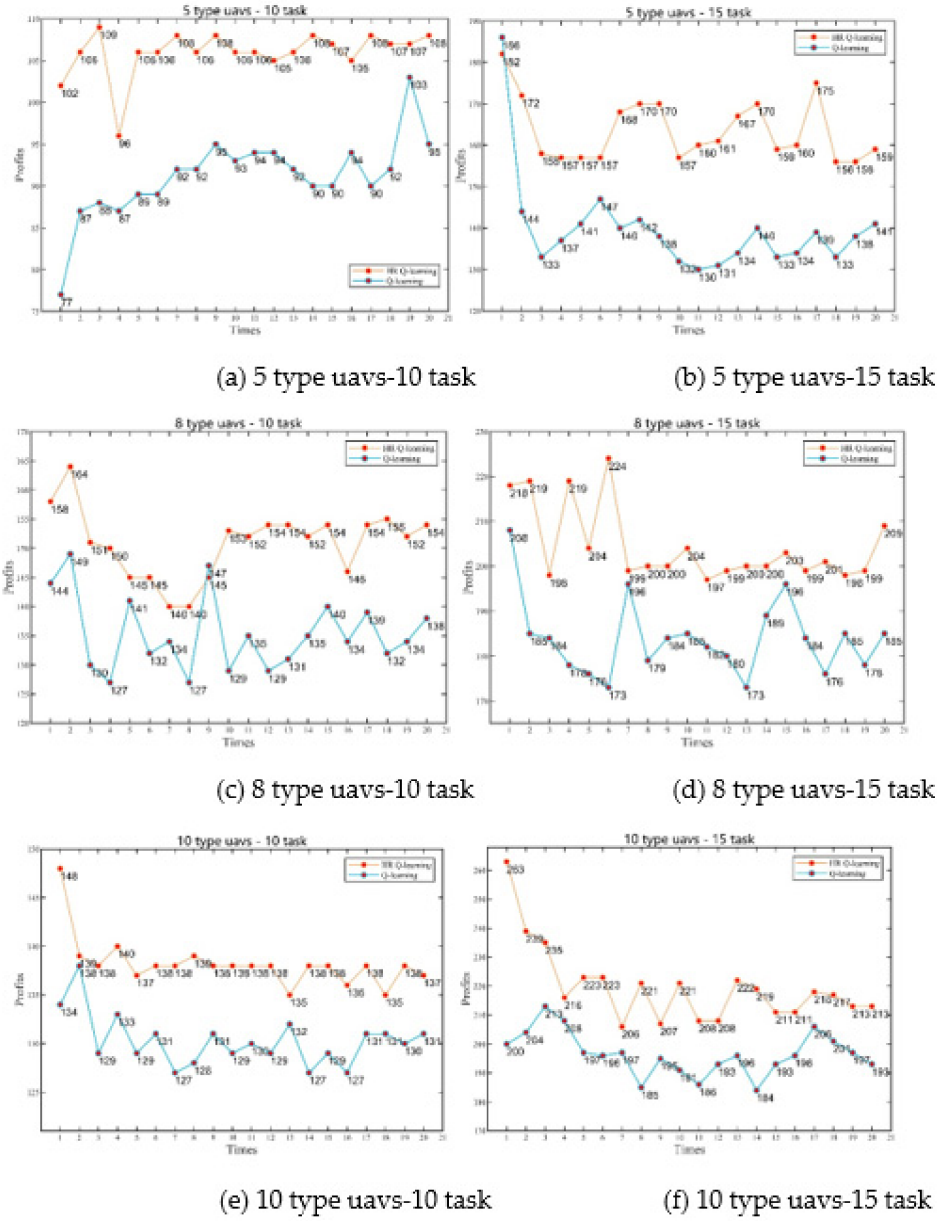


Figure 13. Twenty times profit comparison (**a**–**f**).

From the above figure, it can be seen that the gain value of HR Q-learning algorithm is significantly higher than that of Q-learning algorithm in 20 times, but its minimum value is not necessarily greater than the maximum value of Q-learning algorithm, due to certain random exploration, Q-learning will achieve higher gain value than HR Q-learning algorithm in certain probability. In general, the HR Q-learning algorithm is more advantageous. The mean, maximum, minimum, standard deviation, and coefficient of variation of the results runs for the above six task sets are shown in Table 3, and the statistical significance test *p*-value was calculated for the six data sets. The results are

retained to two decimal places, and Type refers to the type of UAVs and task combination (e.g., 8–10 refers to 8 types of UAVs performing 10 task, corresponding to in Figure 11a).

Table 3. Descriptive analysis of profit values.

Task Group	Type	Algorithm	Best	Worst	Mean	Std	Dispersion Coefficient	Mean Increase Rate
1	5–10	Q-learning	102.66	77.36	91.08	4.75	0.052	16.43%
		HR Q-learning	108.73	95.92	106.04	2.71	0.026	
2	5–15	Q-learning	185.65	129.91	139.56	11.49	0.082	17.19%
		HR Q-learning	182.19	156.37	163.55	7.35	0.045	
3	8–10	Q-learning	148.74	126.76	135.27	6.17	0.046	11.55%
		HR Q-learning	164.01	139.64	150.89	5.80	0.038	
4	8–15	Q-learning	208.36	172.65	183.93	8.71	0.047	11.11%
		HR Q-learning	223.94	195.97	204.37	8.40	0.041	
5	10–10	Q-learning	138.82	127.36	130.25	2.62	0.020	6.04%
		HR Q-learning	147.61	135.43	138.12	2.47	0.018	
6	10–15	Q-learning	212.79	184.15	196.47	7.15	0.036	11.87%
		HR Q-learning	262.70	206.36	219.80	12.84	0.058	
t-test: $p = 0.00087$								

We performed a t-test on the means of the six comparison groups and obtained $p = 0.00087 < 0.05$, indicating that the means of the HR Q-learning algorithm are significantly greater than that of Q-learning algorithm. The above table shows that the maximum, minimum, and mean values of the HR Q-learning algorithm are greater than those of the Q-learning algorithm in the six task sets, except for the maximum value in the second group, and the increment of the average value increases as the number of tasks increases. The standard deviation and coefficient of variation of the HR Q-learning algorithm are smaller than those of the Q-learning algorithm, indicating that the HR Q-learning algorithm can not only achieve a better allocation scheme, but also obtain more stable benefit values. The cost of each type of UAV ranges from [0.8–2.5], so in terms of mean increase, the HR Q-learning algorithm can reduce about ten or so drone wastes than the Q-learning algorithm at ten task volumes, at fifteen task volumes, the HR Q-learning algorithm can reduce about 15 or so UAV wastes than the Q-learning algorithm, i.e., on average, the HR Q-learning algorithm saves about one UAV consumption per task than the Q-learning algorithm. This can significantly reduce the cost loss of drones in practical problems. Since the standard deviation is small, the fluctuation of the benefit values of the two algorithms is relatively small, so the comparison of the benefit values obtained by the two algorithms can be viewed from the mean size alone, and the mean bar chart is shown in Figure 14.

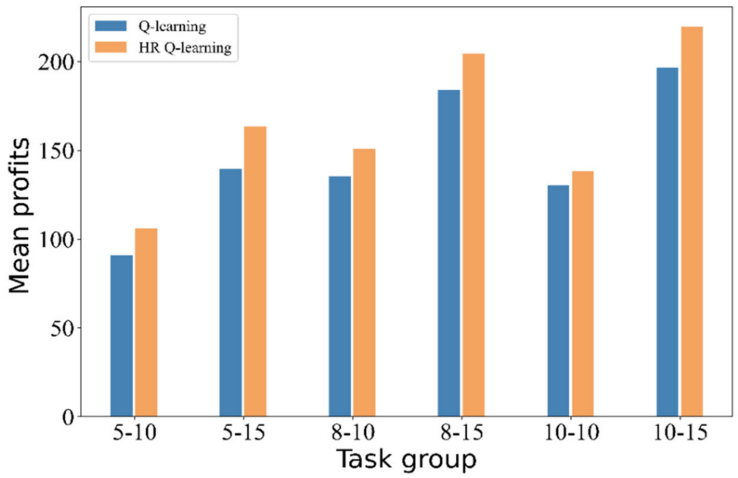


Figure 14. Bar Chart of Means.

From the histogram, the average gain value of the HR Q-learning algorithm is significantly higher than that of the Q-learning algorithm, which indicates that this improved algorithm is to a certain extent improving the effectiveness of task execution and saving resources, which has some practical significance for the task assignment of UAVs in operations.

From the above analysis, we can conclude that HR Q-learning has a higher probability of successful completion of the task, higher gain value and greater stability than the Q-learning algorithm. In addition, HR Q-learning takes less time than the two heuristic algorithms.

6. Conclusions

Based on the characteristics of large-scale missions and the constraints of combat capabilities, this paper establishes a model for cooperative multi-UAV operations and proposes an improved algorithm under reinforcement learning to solve the UAV task assignment model constructed. In this paper, the algorithm is improved according to the characteristics of the model and combined with the design ideas of the q-leaning algorithm. The improved reinforcement learning algorithm is made to reduce the invalid exploration of the algorithm in the operation, while maintaining the exploration ability of the algorithm to avoid falling into local optimum. Finally, by comparing the revenue values of the several algorithms for different number of tasks many times through simulation experiments, it is proved that the improved HR Q-learning algorithm outperforms the original Q-learning algorithm. The specific conclusions are as follows:

(1) The improved HR Q-learning algorithm can increase the possibility of effective exploration, so that more possible allocation schemes can be obtained within a limited number of iterations, so there is a greater possibility of achieving larger gain values. Experiments show that for the overall, the improved HR Q-learning algorithm yields a better allocation scheme.

(2) By increasing the number of UAV types and tasks, it can be found that the HR Q-learning algorithm can consistently obtain satisfactory assignment results with different task sizes and more complex UAV combinations, and maintain stable high yield values compared to other heuristic algorithms and Q-learning algorithms.

The model constructed in this paper is based on a task assignment model in a stable state, but in the actual battlefield, it may encounter some uncertain situations, such as attack and crash, new tasks, and the failure of UAV. Therefore, in the following research, we will continue to consider the task assignment modeling under uncertain state, and design a more appropriate algorithm to solve the model based on the complexity of the actual battlefield and the continuity of the battle process.

Author Contributions: Conceptualization, P.Z.; methodology, P.Z.; software, P.Z.; validation, P.Z.; writing—original draft, P.Z.; visualization, P.Z.; conceptualization, X.F.; validation, X.F.; writing—review and editing, X.F.; supervision, X.F.; project administration, X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Equipment Pre-Research Ministry of Education Joint Fund [grant numbers 6141A02033703].

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. The authors declare that they have no known competing interests.

References

1. Coutinho, W.P.; Battarra, M.; Fliege, J. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Comput. Ind. Eng.* **2018**, *120*, 116–128. [CrossRef]
2. Chu, E.; Kim, J.M.; Jung, B.C. Interference modeling and analysis in 3-dimensional directional UAV networks based on stochastic geometry. *ICT Express* **2019**, *5*, 235–239. [CrossRef]
3. Chamola, V.; Kotesh, P.; Agarwal, A.; Gupta, N.; Guizani, M. A Comprehensive Review of Unmanned Aerial Vehicle Attacks and Neutralization Techniques. *Ad Hoc Netw.* **2020**, *111*, 102324. [CrossRef] [PubMed]
4. Wang, Z.; Liu, L.; Long, T.; Wen, Y. Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding. *Chin. J. Aeronaut.* **2018**, *31*, 339–350. [CrossRef]
5. Fan, J.R.; Li, D.G.; Li, R.P.; Wang, Y. Analysis on MAV/UAV cooperative combat based on complex network. *Def. Technol.* **2020**, *16*, 154–161. [CrossRef]
6. Alotaibi, K.A.; Rosenberger, J.M.; Mattingly, S.; Punugu, R.K.; Visoldilokpun, S. Unmanned aerial vehicle routing in the presence of threats. *Comput. Ind. Eng.* **2018**, *115*, 190–205. [CrossRef]
7. Wu, H.; Shang, H. Potential game for dynamic task allocation in multi-agent system. *ISA Trans.* **2020**, *102*, 208–220. [CrossRef] [PubMed]
8. Jzab, C.; Jxa, B. Cooperative task assignment of multi-UAV system. *Chin. J. Aeronaut.* **2020**, *33*, 2825–2827.
9. Hua, X.; Wang, Z.; Yao, H.; Li, B.; Shi, C.; Zuo, J. Research on many-to-many target assignment for unmanned aerial vehicle swarm in three-dimensional scenarios. *Comput. Electr. Eng.* **2021**, *91*, 107067. [CrossRef]
10. Page, A.J.; Keane, T.M.; Naughton, T.J. Multi-heuristic dynamic task allocation using genetic algorithms in a heterogeneous distributed system-sciencedirect. *J. Parallel Distrib. Comput.* **2010**, *70*, 758–766. [CrossRef]
11. Shao, S.; Peng, Y.; He, C.; Du, Y. Efficient path planning for uav formation via comprehensively improved particle swarm optimization. *ISA Trans.* **2020**, *97*, 415–430. [CrossRef]
12. Zhen, Z.; Chen, Y.; Wen, L.; Han, B. An intelligent cooperative mission planning scheme of uav swarm in uncertain dynamic environment. *Aerosp. Sci. Technol.* **2020**, *100*, 105826. [CrossRef]
13. Shu, L.; Zhang, L.; Fan, Y. Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. *Comput. Ind. Eng.* **2021**, *159*, 107489.
14. Sutton, R.S.; Barto, A.G. Reinforcement learning: An introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [CrossRef]
15. Liu, R.; Cui, J.; Song, Y. Forward Greedy Heuristic Algorithm for N-Vehicle Exploration Problem (NVEP). In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2015.
16. Tan, Z.; Karakose, M. Optimized Deep Reinforcement Learning Approach for Dynamic System. In Proceedings of the 2020 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 12 October–12 November 2020.
17. Zhou, J.; Zhao, X.; Zhang, X.; Zhao, D.; Li, H. Task Allocation for Multi-Agent Systems Based on Distributed Many-Objective Evolutionary Algorithm and Greedy Algorithm. *IEEE Access* **2020**, *8*, 19306–19318. [CrossRef]
18. Maoudj, A.; Hentout, A. Optimal path planning approach based on Q-learning algorithm for mobile robots. *Appl. Soft Comput.* **2020**, *97*, 106796. Available online: https://authors.elsevier.com/c/1bxn25aecShE~{}D (accessed on 9 October 2021). [CrossRef]
19. Baró, G.B.; Martínez-Trinidad, J.F.; Rosas, R.M.V.; Ochoa, J.A.C.; González, A.Y.R.; Cortés, M.S.L. A PSO-based algorithm for mining association rules using a guided exploration strategy. *Pattern Recognit. Lett.* **2020**, *138*, 8–15. [CrossRef]
20. Kurdi, H.; Aldaood, M.F.; Al-Megren, S.; Aloboud, E.; Youcef-Toumi, K. Adaptive task allocation for multi-uav systems based on bacteria foraging behaviour. *Appl. Soft Comput.* **2019**, *83*, 105643. [CrossRef]
21. Gao, S.; Wu, J.; Ai, J. Multi-UAV reconnaissance task allocation for heterogeneous targets using grouping ant colony optimization algorithm. *Soft Comput.* **2021**, *25*, 7155–7167. [CrossRef]
22. Bong-Kyun, K.; Yeong-Dae, K. Heuristic algorithms for assigning and scheduling flight missions in a military aviation unit. *Comput. Ind. Eng.* **2011**, *61*, 1309–1317.
23. Ye, F.; Chen, J.; Sun, Q.; Tian, Y.; Jiang, T. Decentralized task allocation for heterogeneous multi-UAV system with task coupling constraints. *J. Supercomput.* **2020**, *77*, 111–132. [CrossRef]
24. Huang, L.; Qu, H.; Zuo, L. Multi-Type UAVs Cooperative Task Allocation Under Resource Constraints. *IEEE Access* **2020**, *6*, 17841–17850. [CrossRef]
25. Zhou, Y.; Zhao, H.; Chen, J.; Jia, Y. A novel mission planning method for UAVs' course of action. *Comput. Commun.* **2020**, *152*, 345–356. [CrossRef]
26. Wu, H.; Li, H.; Xiao, R.; Liu, J. Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm. *Phys. A Stat. Mech. Its Appl.* **2017**, *491*, 127–141. [CrossRef]
27. Wei, Y.; Blake, M.B.; Madey, G.R. An Operation-Time Simulation Framework for UAV Swarm Configuration and Mission Planning. *Procedia Comput. Sci.* **2013**, *18*, 1949–1958. [CrossRef]
28. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science* **2018**, *362*, 1140–1144. [CrossRef]
29. Aderberg, M.; Czarnecki, W.M.; Dunning, I.; Marris, L.; Lever, G.; Castañeda, A.G.; Beattie, C.; Rabinowitz, N.C.; Morcos, A.S.; Ruderman, A.; et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* **2019**, *364*, 859–865. [CrossRef] [PubMed]

30. Li, X.; Lv, Z.; Wang, S.; Wei, Z.; Zhang, X.; Wu, L. A Middle Game Search Algorithm Applicable to Low-Cost Personal Computer for Go. *IEEE Access* **2019**, *7*, 121719–121727. [CrossRef]
31. Zhao, X.; Zong, Q.; Tian, B.; Zhang, B.; You, M. Fast task allocation for heterogeneous unmanned aerial vehicles through reinforcement learning. *Aerosp. Sci. Technol.* **2019**, *92*, 588–594. [CrossRef]
32. Hu, Z.; Gao, X.; Wan, K.; Zhai, Y.; Wang, Q. Relevant experience learning: A deep reinforcement learning method for UAV autonomous motion planning in complex unknown environments. *Chin. J. Aeronaut.* **2021**, *34*, 187–204. [CrossRef]
33. Xu, J.; Guo, Q.; Xiao, L.; Li, Z.; Zhang, G. Autonomous Decision-Making Method for Combat Mission of UAV based on Deep Reinforcement Learning. In Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 20–22 December 2019.
34. Shin, J.; Badgwell, T.A.; Liu, K.-H.; Lee, J.H. Reinforcement Learning—Overview of recent progress and implications for process control. *Comput. Chem. Eng.* **2019**, *127*, 282–294. [CrossRef]
35. Singh, S.; Lewis, R.L.; Barto, A.G.; Sorg, J. Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective. *IEEE Trans. Auton. Ment. Dev.* **2010**, *2*, 70–82. [CrossRef]
36. John, P.D.; Mithra, N.M. A Finite Horizon Markov Decision Process Based Reinforcement Learning Control of a Rapid Thermal Processing system. *J. Process. Control.* **2018**, *68*, 218–225.
37. Shahrabi, J.; Adibi, M.A.; Mahootchi, M. A reinforcement learning approach to parameter estimation in dynamic job shop scheduling. *Comput. Ind. Eng.* **2017**, *110*, 75–82. [CrossRef]
38. Littman, M.L. Reinforcement learning improves behaviour from evaluative feedback. *Nature* **2015**, *521*, 445–451. [CrossRef] [PubMed]