electronics

MDPI

Article

Cooperative Coverage Path Planning for Multi-Mobile Robots Based on Improved K-Means Clustering and Deep Reinforcement Learning

Jianjun Ni [1,2,*], Yu Gu [1], Guangyi Tang [1], Chunyan Ke [2] and Yang Gu [1]

[1] College of Artificial Intelligence and Automation, Hohai University, Changzhou 213200, China; gyguyu@hhu.edu.cn (Y.G.); tang_gy@hhu.edu.cn (G.T.); 20231153@hhu.edu.cn (Y.G.)
[2] College of Information Science and Engineering, Hohai University, Changzhou 213200, China; chunyanke@hhu.edu.cn
* Correspondence: jianjun_ni@hhu.edu.cn

Abstract: With the increasing complexity of patrol tasks, the use of deep reinforcement learning for collaborative coverage path planning (CPP) of multi-mobile robots has become a new hotspot. Taking into account the complexity of environmental factors and operational limitations, such as terrain obstacles and the scope of the task area, in order to complete the CPP task better, this paper proposes an improved K-Means clustering algorithm to divide the multi-robot task area. The improved K-Means clustering algorithm improves the selection of the first initial clustering point, which makes the clustering process more reasonable and helps to distribute tasks more evenly. Simultaneously, it introduces deep reinforcement learning with a dueling network structure to better deal with terrain obstacles and improves the reward function to guide the coverage process. The simulation experiments have confirmed the advantages of this method in terms of balanced task assignment, improvement in strategy quality, and enhancement of coverage efficiency. It can reduce path duplication and omission while ensuring coverage quality.

Keywords: coverage path planning; deep reinforcement learning; dueling network; improved K-Means clustering algorithm; multi-mobile robots

Citation: Ni, J.; Gu, Y.; Tang, G.; Ke, C.; Gu, Y. Cooperative Coverage Path Planning for Multi-Mobile Robots Based on Improved K-Means Clustering and Deep Reinforcement Learning. *Electronics* **2024**, *13*, 944. https://doi.org/10.3390/electronics13050944

Academic Editor: Fernando De la Prieta Pintado

Received: 24 January 2024
Revised: 21 February 2024
Accepted: 28 February 2024
Published: 29 February 2024

1. Introduction

The core purpose of coverage path planning (CPP) is to formulate an algorithmic strategy that enables mobile agents to efficiently traverse every point in a specific area. The strategy aims to achieve maximum coverage of the designated space through precise calculation and planning while minimizing path duplication and omissions [1–3]. Currently, coverage path planning has shown its importance in multiple practical applications, including precision agriculture [4], automated cleaning [5,6], disaster response and search and rescue [7], patrol monitoring [8,9] and so on.

With the development of the social economy, people pay more attention to economic benefits and the optimal allocation of resources in all aspects. In patrol monitoring, if traditional manual patrols are still employed, not only can cost reduction objectives not be achieved but also, due to the complexity of the patrol environments, it may reduce resource allocation efficiency and increase dependence on human resources. With the continuous advancement of mobile robot technology, especially innovations in autonomous navigation and remote operation, the application of coverage path planning in patrol monitoring has been significantly promoted. The use of mobile robots can reduce reliance on a large number of manpower and achieve wider monitoring coverage under conditions of limited resources. In addition, the coverage path planning of mobile robots can provide real-time monitoring and assessment, speeding up response speed and decision-making efficiency. Therefore, using mobile robots for patrol monitoring can not only effectively reduce costs

and reduce dependence on human resources, but also improve monitoring coverage and efficiency, speed up response speed, and improve decision-making efficiency. It is also in this context, that this paper uses an improved algorithm for coverage path planning to carry out patrol monitoring.

In the process of mobile robot path planning, there are many different methods, such as graph-theoretic methods, heuristic methods, machine learning-based methods, and so on. Methods based on graph theory include approaches based on Voronoi diagrams [10] and edge probability heat graph [11]. Heuristic methods include the genetic algorithm [12], ant colony algorithm [13], particle swarm optimization algorithm [14], virus-evolutionary genetic algorithm [15], Dragonfly Algorithm [16], etc. In machine learning-based methods, many methods such as deep learning and reinforcement learning are used at present, such as [17–19]. The application of deep reinforcement learning (DRL) in coverage path planning mainly benefits from its effective learning and decision-making in complex environments. Compared with traditional algorithms, deep reinforcement learning has the following advantages: (1) It can process high-dimensional input data so that the model can learn directly from the raw perceptual data; (2) DRL learns strategies through interaction with the environment and can be used in environments that are difficult to accurately describe by the model. (3) In the case of environmental changes, DRL's adaptive ability can continuously optimize the strategy, which is very important for CPP in unknown environments. Therefore, the coverage path planning in this paper also uses a deep reinforcement learning algorithm and introduces a dueling network.

At present, both single-mobile robots and multi-mobile robots are applied in CPP [20,21]. To some extent, a single mobile robot can perform coverage path planning tasks effectively. Theile et al. [18], Shen et al. [22], and Xing et al. [23] used the improved DQN algorithm to achieve good coverage results and can basically achieve the goal of full coverage. However, the target environmental areas in these papers are relatively small. In Theile et al. [24], due to the expansion of the target coverage area and the increased complexity of the entire map area, the final coverage performance is much worse than that in the small map area. At this moment, utilizing multiple agents for coverage path planning would be a great choice. Through collaboration among multiple agents, coverage can be achieved in broader and more complex areas. At present, many methods are to allow multiple robots to cooperate throughout the entire map space. Ruan et al. [25] used a rolling optimization and dispersed predator–prey model to perform coverage tasks on the entire map and achieved good results; Zhang et al. [3] proposed a multi-mobile robot coverage path planning based on Monte Carlo tree search and finally completed the coverage task successfully.

However, since the above-mentioned collaboration between multiple robots is carried out in the entire map area, there will inevitably be problems such as high coverage repetition and excessively long total coverage paths. Therefore, it would be a good choice to divide the entire map area into small areas, and then use multiple robots to cover the small areas. Luo et al. [26] first used the fuzzy C-means clustering method to divide the entire area, and then performed the coverage task; Li et al. [27] first used the regional growth strategy to divide the area into K sub-areas and then used a single robot in each sub-area to complete the task. Finally, these methods not only complete the coverage task well but also reduce the path repetition ratio and path length. Therefore, in this paper, the strategy of dividing the whole area into small areas first, and then using multi-robots for cooperative coverage is also adopted.

After rasterizing the map, there are many methods to further divide it into small areas, and then use multi-robots to perform coverage path planning, such as Trapezoidal decomposition [28], Boustrophedon decomposition [29,30], Morse decomposition [31,32], and clustering algorithms [26,33]. These methods can effectively reduce environmental complexity and improve coverage efficiency. In this paper, the K-Means clustering algorithm is selected and improved to better divide the map. Compared with other algorithms, it has the advantages of simplicity, efficiency and wide adaptability [34]. Mobile robots have

the advantages of long running time, continuous completion of tasks, and more economical and practical advantages [35].

In summary, this paper uses an improved K-Means clustering algorithm to divide the map, introduces dueling networks and improved reward functions in deep reinforcement learning, and uses multi-mobile robots for coverage path planning, which brings the following benefits:

1. Using the improved K-Means clustering method, the location of the initial value of k is arranged more reasonably, and a better map division effect is obtained, thus making the tasks of each robot more balanced;
2. The dueling network is introduced and the reward function of deep reinforcement learning is improved, which improves the strategy quality and learning efficiency;
3. Using the cooperation of multi-robots, the overall coverage ratio is effectively improved, and the repeated coverage and coverage paths are reduced.

The rest of this paper is organized as follows. The second section is a description of the CPP problem; the third section is the introduction of the improved K-Means clustering algorithm and deep reinforcement learning algorithm; the fourth section is the experiment and result analysis, and the fifth section is the summary of the article and the future prospects.

2. Problem Description

In this paper, when performing coverage path planning, the environmental map is divided, and each divided small map is assigned to the robot that performs the coverage task. Multiple robots use deep reinforcement learning to complete their respective coverage tasks, ultimately achieving coverage of the entire map. The framework of the proposed method is shown in Figure 1. This approach builds upon existing methodologies in the field of multi-robot systems and deep reinforcement learning, as discussed in [3,18]. Next, this paper will define CPP from two aspects: the environment and the objectives.

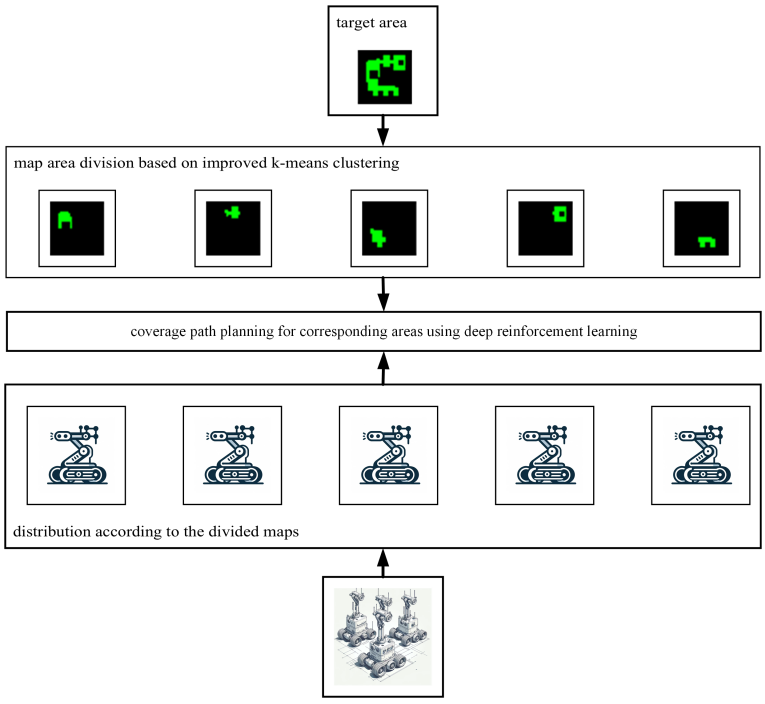


Figure 1. The overall framework of the method proposed in this paper.

2.1. Environment and Multi-Mobile Robot Model

In order to simplify the map space, this paper converts the real environment into a two-dimensional grid map. The size of this map is $M \times M \in \mathbb{N}^2$, where $\mathbb{N}^2$ is the set of natural numbers. In the built grid map, the environment of the mobile robot can be divided

into departure and return areas, prohibited areas and obstacles. The entire map can be represented by the tensor $\mathbf{M} \in \mathbb{B}^{M\times M\times 3}$, where $\mathbb{B} = \{0,1\}$. $M \times M \times 3$ indicates that the map contains three layers, namely: (1) departure and return area; (2) combination of prohibited areas and obstacles; (3) obstacles alone.

In a grid map, a mobile robot occupies exactly one grid cell. The current position of the mobile robot can be defined by $\phi_t \in \mathbb{N}^2$. According to the division of the task area, the departure and return area of the mobile robot is set nearby. Therefore, in the process of executing the task, first of all, the multi-mobile robots will start from their respective set departure and return areas and go to different divided small map areas (that is, the target coverage area). Then, each mobile robot must cover the designated target coverage area as completely as possible without entering prohibited areas or encountering obstacles. Finally, the mobile robots return to their respective set departure and return areas.

2.2. Task Objective

The task objective of CPP is to ensure that the robot comes to its designated target coverage area and observes the target coverage area through the camera sensor. The entire target coverage area can be described as $\mathbf{T}_t \in \mathbb{B}^{M\times M}$, and each element in $\mathbf{T}_t$ represents whether the grid cell corresponding to this element needs to be covered. For each mobile robot, $\mathbf{T}_t^i \in \mathbb{B}^{M\times M}$ represents the i-th small target area to be covered after the entire target area is divided; the corresponding i-th mobile robot is required to perform the coverage task. $\mathbf{V}_t^i \in \mathbb{B}^{M\times M}$ represents the current field of vision of the mobile robot, which is a square area of 1×1, that is to say, the mobile robot can just cover the current position.

Therefore, every time the mobile robot takes an action, the corresponding remaining target coverage area is:

$$\mathbf{T}_{t+1}^i = \mathbf{T}_t^i \wedge \neg \mathbf{V}_t^i, \tag{1}$$

where $\wedge$ and $\neg$ are logical unit symbols AND and NOT, respectively. In addition, the departure and return areas can be used as target coverage areas, while obstacles and prohibited areas cannot be used as target coverage areas. Finally, the goal of this paper is to maximize the coverage area and return to the departure and return area in the cooperation of multiple mobile robots under safe conditions.

3. Proposed Method

3.1. Improved K-Means Clustering Algorithm

In the process of using multi-robots for coverage path planning, in order to avoid collisions between robots, which will damage the robots and reduce the completion ratio of tasks, this paper divides the target coverage area in the entire map into several small areas, and each area is covered by a robot. An approach like this not only avoids collisions between robots but also reduces the complexity of the task, which is helpful in improving the coverage ratio. In order to divide the target coverage area more reasonably, the improved K-Means clustering algorithm is applied.

The K-Means clustering algorithm is a classic unsupervised learning algorithm that is widely used in data mining and statistical data analysis [36]. The main purpose of this algorithm is to divide a data set containing n data points into k clusters so that data points in the same cluster are as similar as possible and data points in different clusters are as different as possible. The core of the K-Means clustering algorithm is to minimize the intra-cluster variance, which is the sum of squares of the distance from each point to the cluster center. This can be expressed as follows:

$$J = \sum_{i=1}^{k} \sum_{x\in S_i} \|x - c_i\|^2, \tag{2}$$

where J is the sum of variance within the cluster; k is the number of clusters; S_i is the set of data in the point of the i-th cluster; c_i is the center of the i-th cluster, and $\|x - c_i\|$ is the

Euclidean distance from point x to cluster center c_i. According to the obtained data in the cluster center, the new cluster center is obtained by using the following equation:

$$c_i = \frac{1}{S_i} \sum_{x \in S_i} x, \tag{3}$$

Using the K-Means clustering algorithm to divide the target area is helpful in simplifying complex problems and obtaining a more balanced and consistent area division. However, in practice, the number of clusters (i.e., k value) and the selection of initial cluster centers will greatly affect the final practical effect [37,38].

In the choice of the number of clusters, if the k value is too small, the clustering results will be over-generalized, and ultimately lead to inappropriate merging of data; while if the k value is too large, the clustering will be overfitted, increasing the running time of the algorithm and the consumption of computing resources. In order to obtain the appropriate k value, the elbow rule is applied in this paper to help determine the appropriate k value. The core idea of the elbow rule is to find the "elbow point" where the gain decreases as the k value increases, that is, after this point, adding more clusters will not significantly improve the clustering performance.

After choosing the value of k, it is necessary to initialize the cluster center. Random initialization is a common practice, but it may lead to unstable results and convergence to suboptimal solutions [39]. The K-Means++ algorithm can optimize the selection of the initial cluster center. It will first randomly select a point as the first cluster center:

$$c_1 = random(X), \tag{4}$$

where $X = \{x_1, x_2, \cdots, x_N\}$ is the set of data, and $c_1 \in C$, C is the current cluster center set. Then, calculate the distance from each point to the nearest cluster center, which is usually the Euclidean distance. The calculation equation is:

$$D(x_i) = \min_{c \in C} \|x_i - c\|^2, \tag{5}$$

Then, the probability of the data selected as the next cluster center is proportional to $D(x_i)$. It can be expressed as:

$$P(x_i) = \frac{D(x_i)}{\sum_{j=1}^{N} D(x_j)}, \tag{6}$$

where $\sum_{j=1}^{N} D(x_j)$ is the sum of the distances from each point to its nearest cluster center, which is used to normalize the probability distribution. Then, according to the above probability distribution, the remaining initial cluster centers are selected in sequence until the predetermined k value is reached.

However, in the above process, although the selection of the subsequent initial cluster center becomes reasonable under the optimization of the algorithm, the selection of the first cluster center is still random, which may affect the selection process of the subsequent cluster center and the final clustering quality. In order to select the first initial cluster center more reasonably, this paper introduces the method of Minimum Spanning Tree (MST). Minimum spanning tree is a concept in graph theory, which connects all vertices in a graph without forming a loop, and ensures that the total edge weight connecting all vertices is the minimum [40], as shown in Figure 2. There are 7 points A-G. The numbers on the connection line represent the weight of the edges. After the MST process, a graph is generated that does not form a loop but connects all vertices according to the weights. In this paper, the distance between two vertices is used to represent the weight of edges, and the distance equation is:

$$d_{ij} = \sqrt{\sum_{k=1}^{m} \left(x_{ik} - x_{jk} \right)^2}, \tag{7}$$

Among them, x_{ik} and x_{jk} represent the coordinates of x_i and x_j in the k-th dimension, and m represents the total dimension. Then, randomly start from a vertex and store it in the empty set L. Use a greedy algorithm to select the unconnected point with the smallest weight as the next vertex from the weight set of the points in L to the remaining points. The equation is:

$$N = \{x | x \in Xx \notin L\}, \tag{8}$$

$$D_{ij} = [d_{i1}, d_{i2}, \ldots, d_{im}], \tag{9}$$

$$x_{next} = \arg\min(D_{ij}), \tag{10}$$

where the set N is to remove the connected vertices in the data set X, the data in D_{ij} are the weight between each vertex in the set L and each vertex in the set N, and x_{next} is to obtain the point with the smallest weight from D_{ij} as the next connection point through the greedy algorithm. By repeating Equations (7)–(9) continuously, the minimum spanning tree graph can be obtained. In this case, a longer edge usually means connecting vertices with a larger weight, which is reflected in the distribution of points as a larger spatial spacing. In K-Means++, the ideal initial cluster centers should be far away from each other to cover different areas of the data space as much as possible. Therefore, applying the minimum spanning tree algorithm to the selection of the first initial cluster center in K-Means++ and randomly selecting the vertex connected with the largest weight in the minimum spanning tree as the first cluster center is more conducive to improving the quality of clustering and improve the effect of final coverage path planning. The corresponding pseudocode of the improved k-clustering algorithm is shown in Algorithm 1.

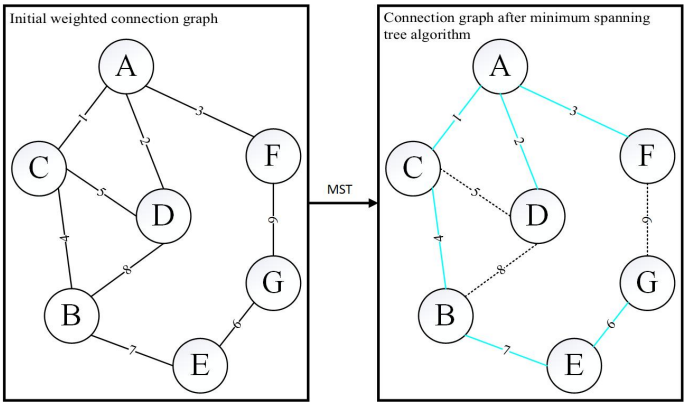


Figure 2. Schematic graph of the results of the minimum spanning tree algorithm. Among them, letters represent points, numbers represent the weights between points, and blue lines represent the results of the minimum spanning tree algorithm.

Algorithm 1 Improved K-Means clustering algorithm

Require: Input the data set X to be clustered
Ensure: Output Cluster center k and data points in each cluster

1: According to the elbow rule, obtain the optimal number of k values
2: The MST method is introduced, and the minimum spanning tree graph is obtained according to Equations (7)–(10). The vertices connected with the largest weights in the minimum spanning tree are randomly used as the first initial cluster center in the K-Means++ algorithm
3: Use the K-Means++ algorithm to obtain the remaining $k-1$ initial clustering center points according to Equations (5) and (6)
4: After obtaining k initial clustering center points, select new clustering center points according to Equation (3), and perform clustering again based on new clustering center points
5: Repeat the operation of Step 4 until the cluster center points no longer changes

3.2. Environment Settings

Deep reinforcement learning (DRL) combines the principles of deep learning (DL) and reinforcement learning (RL) to solve complex decision-making and control problems. Deep learning enables agents to learn abstract expressions from large amounts of data through neural networks to extract features. Reinforcement learning focuses on maximizing rewards and optimizing the decision-making process through interactions with the environment. The process of DRL is schematically illustrated in Figure 3.

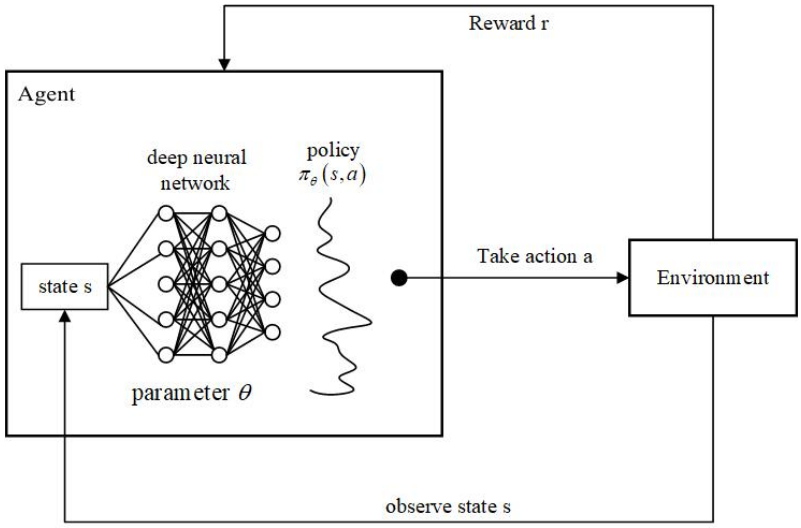


Figure 3. Concept graph of deep reinforcement learning.

In this paper, the entire map has been divided by the improved K-Means clustering algorithm in Section 3.1. However, when using mobile robots for coverage path planning, there are still problems that need feedback on the observed information in real-time, and the map features are complex, so it will be difficult to model and solve the CPP problem. Therefore, in this paper, deep reinforcement learning is applied and converted into partially observable Markov decision processes (POMDP), which are defined by tuples $(\mathcal{S}, \mathcal{A}, \mathrm{P}, \mathrm{R}, \gamma)$. Among them, $\mathcal{S}$ represents the state space, $\mathcal{A}$ represents the action space, $\mathrm{P} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the probability transfer function, $\mathrm{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is the reward function that maps the current state, action, and the next state into real-valued rewards, and $\gamma \in (0,1)$ is the discount factor. Therefore, the definition of state s_t in the environment of this article is:

$$s_t = (\mathbf{M}, \mathbf{T}_t, \phi_t), \tag{11}$$

State s_t consists of three parts: (1) $\mathbf{M}$ represents the current environment map, including departure and return areas, prohibited areas and obstacles; (2) $\mathbf{T}_t$ represents the target map that still needs to be covered at time t; (3) ϕ_t is the position of the mobile robot position at time t. $a_t \in \mathcal{A}$ represents the possible actions of the mobile robot at time t. Use $\mathcal{B}_{behevior}$ to represent the set of actions in action space $\mathcal{A}$:

$$\mathcal{B}_{behevior} = \{forward, backward, left, right, reach\}, \tag{12}$$

Among them, the first four elements represent the actions of the forward, backward, left and right, and the last element represents whether the mobile robot returns to the designated area. The reward function during the entire task is:

$$r = r_1 + r_2, \tag{13}$$

where r_1 is some conventional action reward values, which can be expressed as:

$$r_1 = \begin{cases} -1, & boundary \quad penalty \\ -0.2, & movement \quad penalty \\ 0.4, & coverage \quad reward \end{cases}, \tag{14}$$

r_2 is a newly introduced reward value in this paper. Since the process of deep reinforcement learning is unsupervised, it is easier to produce repeated coverage paths, while the introduction of r_2 can reduce the repetition ratio and improve coverage efficiency. The position of the mobile robot at time t is ϕ_t, so at time t, eight grid points around the position of the agent can be expressed as ϕ'_t. r_2 can be expressed by the following equation:

$$r_2 = \begin{cases} 0.4, & if \quad \phi_t \notin \mathbf{T}_t \quad and \quad \exists \phi'_t \in \mathbf{T}_t \quad and \quad \forall \phi'_{t-1} \notin \mathbf{T}_{t-1} \\ 0.2, & if \quad \phi_t \notin \mathbf{T}_t \quad and \quad \exists \phi'_{t-1} \in \mathbf{T}_t \\ 0, & if \quad \phi_t \notin \mathbf{T}_t \quad and \quad \forall \phi'_t \notin \mathbf{T}_t \quad and \quad \forall \phi'_{t-1} \notin \mathbf{T}_{t-1} \\ -0.4, & if \quad \phi_t \notin \mathbf{T}_t \quad and \quad \forall \phi'_t \notin \mathbf{T}_t \quad and \quad \exists \phi'_{t-1} \in \mathbf{T}_{t-1} \end{cases}, \tag{15}$$

For instance, if there is no unit to be covered around the agent at time $t-1$, at time t, the location of the agent is not the cell that needs to be covered, and there are cells that need to be covered in the eight grids around the location; then $r_2 = 0.4$, and the rest of the reward values are obtained, and so on. By introducing r_2, the mobile robot can not only be guided to avoid those covered areas but also be guided to uncovered areas.

3.3. Deep Reinforcement Learning Using Dueling Network Structure

In the process of cooperative coverage path planning by using multi-mobile robots, due to the narrow field of view of the mobile robot, it can only cover one grid cell in the grid map at a time. Therefore, in a complex environment, in order to improve the coverage, mobile robots will inevitably enter those areas that are prone to collision. The deep reinforcement learning process is unsupervised. Entering these areas will not only increase the difficulty of training but also greatly reduce the safety in the coverage path planning process. Value-based deep reinforcement learning, such as DQN and DDQN algorithms, will directly output the Q-value function, which reduces the adaptability to complex environments, increases computing costs, and reduces learning efficiency [41]. Therefore, this paper introduces the dueling network on the basis of DDQN and optimizes the algorithm by optimizing the structure of the neural network to more effectively focus on obstacles in action [42].

The schematic structure of the dueling network is shown in Figure 4. It divides the Q-value network into two parts. The first part is only related to the state s and has nothing to do with the action a to be taken. This part is the value function, denoted as $V(s;\omega,\alpha)$. The second part is related to both the state a and the action a. This part is the advantage function, denoted as $A(s,a;\omega,\beta)$, then the Q-value function can be expressed as:

$$Q(s,a;\omega,\alpha,\beta) = V(s;\omega,\alpha) + A(s,a;\omega,\beta), \tag{16}$$

where ω refers to the network parameters of the public part, α is the parameter of the value function part, and β is the parameter of the advantage function part. V and A here are not unique. If the two networks fluctuate up and down with the same amplitude and in opposite directions, then the output of the networks is the same, but since both networks are fluctuating up and down, neither network is stable. To solve this problem, we force the advantage function estimator to have zero advantage over the chosen action. That is, we let the last module of the network implement forward mapping:

$$Q(s,a;\omega,\alpha,\beta) = V(s;\omega,\alpha) + \left(A(s,a;\omega,\beta) - \frac{1}{|\mathcal{B}_{behevior}|} \sum_{a^*} A(s,a^*;\omega,\beta) \right), \tag{17}$$

where $\frac{1}{|\mathcal{B}_{behevior}|}\sum_{a^*} A(s, a^*; \omega, \beta)$ is the average of all possible advantage functions, and a^* is used here to represent a general iteration for all actions. In this way, the stability of the network can be improved.

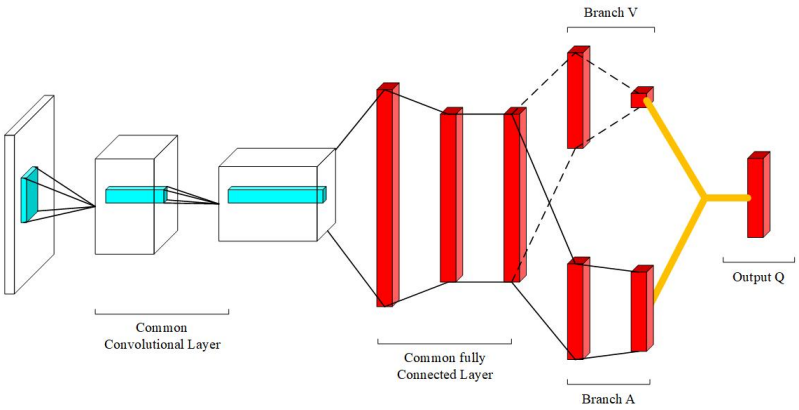


Figure 4. Structure of the dueling network. Blue represents the convolutional layer operations, and red represents the fully connected layer operations related to the dueling network.

On the basis of DDQN, the dueling network is introduced, and there is still an estimation network and a target network. The parameters are θ and $\overline{\theta}$, and θ, $\overline{\theta}$ are the set of $\{\omega, \alpha, \beta\}$. Then, the loss function of DDQN with the dueling network is:

$$L(\theta) = \mathbb{E}_{s_t, a_t, s_{t+1} \sim D}\left[\left(Q_\theta(s_t, a_t) - Y_t\right)^2\right], \tag{18}$$

The target value is:

$$Y_t = r(s_t, a_t) + \gamma Q_{\bar{\theta}}\left(s_{t+1}, \arg\max_{a'} Q_\theta\left(s_{t+1}, a'\right)\right), \tag{19}$$

The Q value iteration equation is:

$$Q_\theta(s_t, a_t) = Q_\theta(s_t, a_t) + \xi(Y_t - Q_\theta(s_t, a_t)), \tag{20}$$

where ξ is the learning rate. In deep reinforcement learning, a neural network is used to approximate the Q value. In this paper, the Adam optimizer is used to train and estimate the parameter θ of the network, and the iterative Q value is updated by updating the parameters. The updating formula of parameter $\overline{\theta}$ of the target network is:

$$\bar{\theta} \leftarrow (1 - \tau)\bar{\theta} + \tau\theta, \tag{21}$$

As shown in Figure 5, it is a schematic diagram of the whole network structure. After inputting the whole grid map into the convolutional neural network after a convolution operation, the output data are flattened into a one-dimensional array that is output to the fully connected layer; the value function V and advantage function A are output; the final Q value is obtained. Then, based on the obtained Q value, the sampling softmax strategy is used to obtain the next action. The equation can be expressed as:

$$\pi(a_t|s_t) = \frac{e^{Q_\theta(s_t, a_t)/\eta}}{\sum_{\forall a \in \mathcal{A}} e^{Q_\theta(s_t, a)/\eta}}, \tag{22}$$

where the parameter $\eta \in \mathbb{R}^+$ is used to balance the relationship between exploration and utilization during training.

Finally, the improved K-Means clustering algorithm and DDQN algorithm with a dueling network are combined and applied to the coverage path planning of multi-mobile robots. The pseudocode of the final algorithm is shown in Algorithm 2.

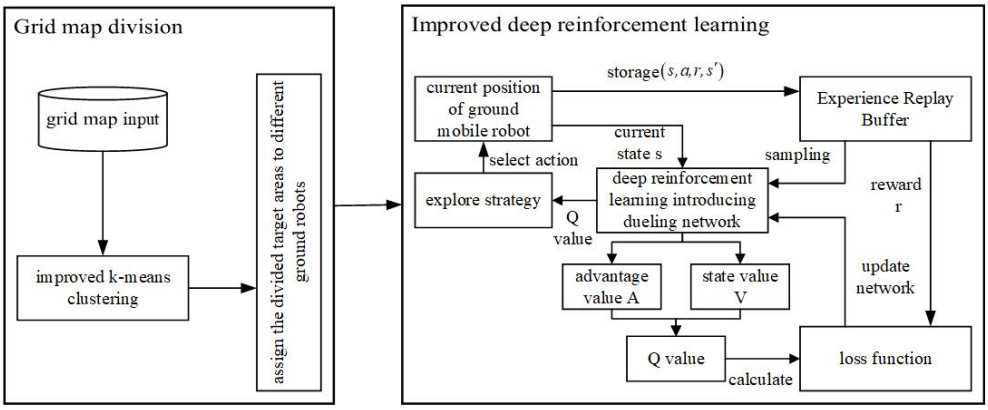


Figure 5. The overall process of this algorithm.

Algorithm 2 The overall process of the algorithm

Require: Input the grid map to be covered, set of target coverage points X
Ensure: Output the parameters θ, $\overline{\theta}$ of the model

1: According to Algorithm 1, the data in X are reasonably divided to obtain small target coverage areas
2: Input the grid map into the convolutional neural network, introduce the dueling network, obtain the Q value according to Equations (16) and (17), guide the next action of the agent through Equation (22), obtain the next state, and store the experience data in the experience pool
3: Obtain a small batch of experience data from the experience pool, obtain the loss function according to Equation (18), and then use the gradient descent method to update θ
4: Update $\overline{\theta}$ according to Formula (21)
5: Repeat Step 2–Step 4 until the specified training period is reached, and the best θ and $\overline{\theta}$ are output

Therefore, the algorithm proposed in this paper first uses the minimum spanning tree to improve the selection of the first initial clustering point in the clustering algorithm, thereby improving the final clustering effect and thus better dividing the coverage of each mobile robot. A complex task is divided into simple tasks to be completed by multiple mobile robots, which reduces the complexity of tasks and optimization strategies. Then, the reward function in deep reinforcement learning is improved to reduce randomness in the training process, guide the agent's exploration, and improve the exploration efficiency. Finally, a dueling network is introduced to improve sample efficiency. Through the above process, the coverage path planning problem in this paper can be better solved.

4. Experiment and Discussion

4.1. Simulation Settings

This paper cites the map "Manhattan 32" in the literature [24], which is a 32 × 32 grip map. In order to adapt to the coverage path planning of multi-mobile robots in this paper, based on the "Manhattan 32" map, more departure and return areas are set up according to the effect of improved K-Means clustering. In terms of experimental hardware configuration, this study uses an NVIDIA RTX 2080 graphics card. In terms of software environment, the experiment is based on the Ubuntu 20.04 operating system, uses Python 3.7 as the programming language, and uses TensorFlow 2.4 as the main deep learning framework for development and testing.

In this paper, clustering is performed first, followed by coverage path planning. In order to evaluate the effect of the improved K-Means clustering algorithm, the average silhouette coefficient, Calinski–Harabasz Index, Davies–Bouldin Index and Iteration are used. The higher the average silhouette coefficient, the better the clustering quality; the larger the Calinski–Harabasz Index, the more tightness within clusters and more separation between clusters; the smaller the Davies–Bouldin Index the better the distribution of the clusters; the Iteration is an index to measure the speed of obtaining the final cluster center. When conducting coverage path planning, the evaluation metrics include coverage ratio (CR), repetition ratio (RR), return to specified area Reached, and coverage ratio and reached (CRAR). CR represents the ratio of the target area covered at the end of the task to the initial target area, RR represents the ratio of the repeated parts when covering the target area, Reached refers to the ratio of returning to the designated area, and CRAR is the comprehensive index of covering and being able to return to the designated area. In the simulation process, the target area is randomly generated, and we select the average of 1000 test results as our experimental results. The parameter settings of the algorithm in this paper are shown in Table 1, and the legend of the "Manhattan 32" map is shown in Table 2.

Table 1. Related parameters.

Parameter	32×32	Description
$\|\theta\|$	1,175,302	Trainable parameters
n_c	2	Number of conv. layers
n_k	16	Number of conv. kernels
s_k	5	Conv. kernel size
D	50,000	Experience pool size
m	128	Small batch sample size

Table 2. Some legends in the "Manhattan 32" map.

Symbol	Description
	Departure and return area
	Prohibited area
	Need to cover area
	Obstacle area
	Remaining need to be covered area

4.2. Experimental Results

In the cooperative coverage path planning of multi-mobile robots in this paper, the map in the literature [24] is cited. The improved K-Means clustering algorithm is first applied, and the elbow rule is used to select the appropriate k value, as shown in Figure 6. It can be seen from the figure that as the number of clusters k increases, the sum of squared errors (SSE) gradually decreases. At $k = 4$ and $k = 5$, the decline rate of SSE begins to slow down, which is the appropriate number of clusters. In this paper, in order to reduce the complexity of the environment and have better coverage, $k = 5$ is selected as our number of clusters.

Then, the introduced minimum spanning tree algorithm is used to select the first initial cluster center in K-Means++ to obtain better clustering results, as shown in Figure 7 and Table 3. The comparison algorithms used in Table 3 include K-Means [36], K-Means++ [37] and K-MeansII [43].

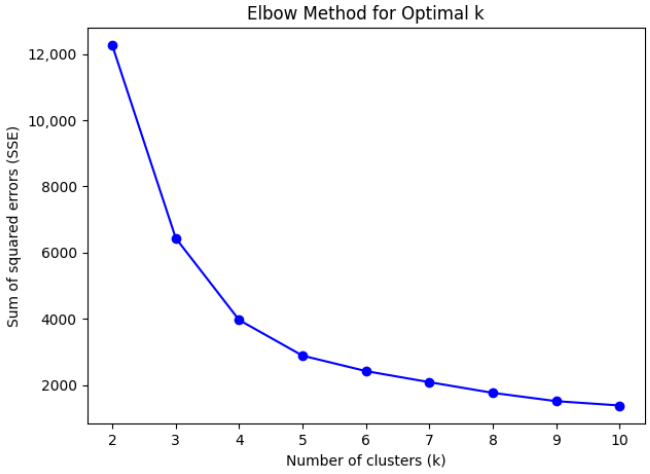


Figure 6. Selection of the number of clusters k in the clustering algorithm.

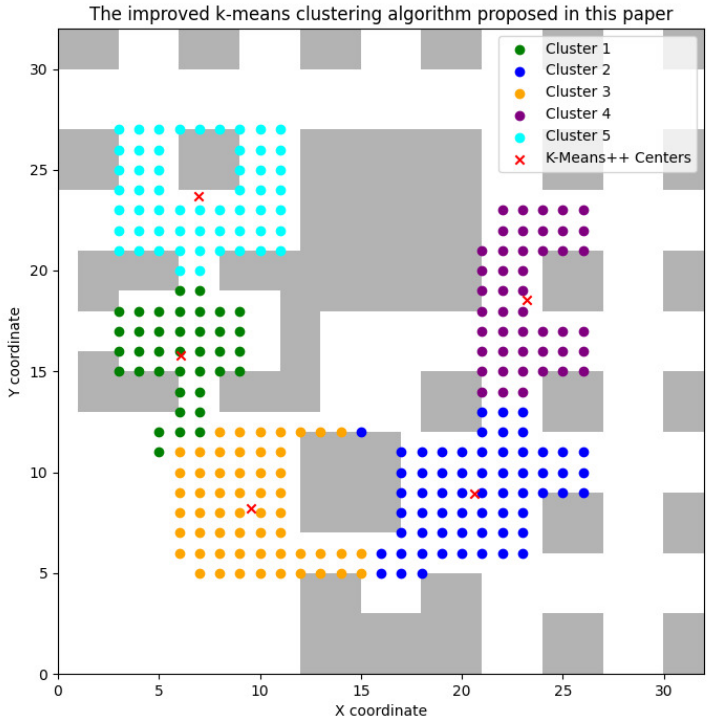


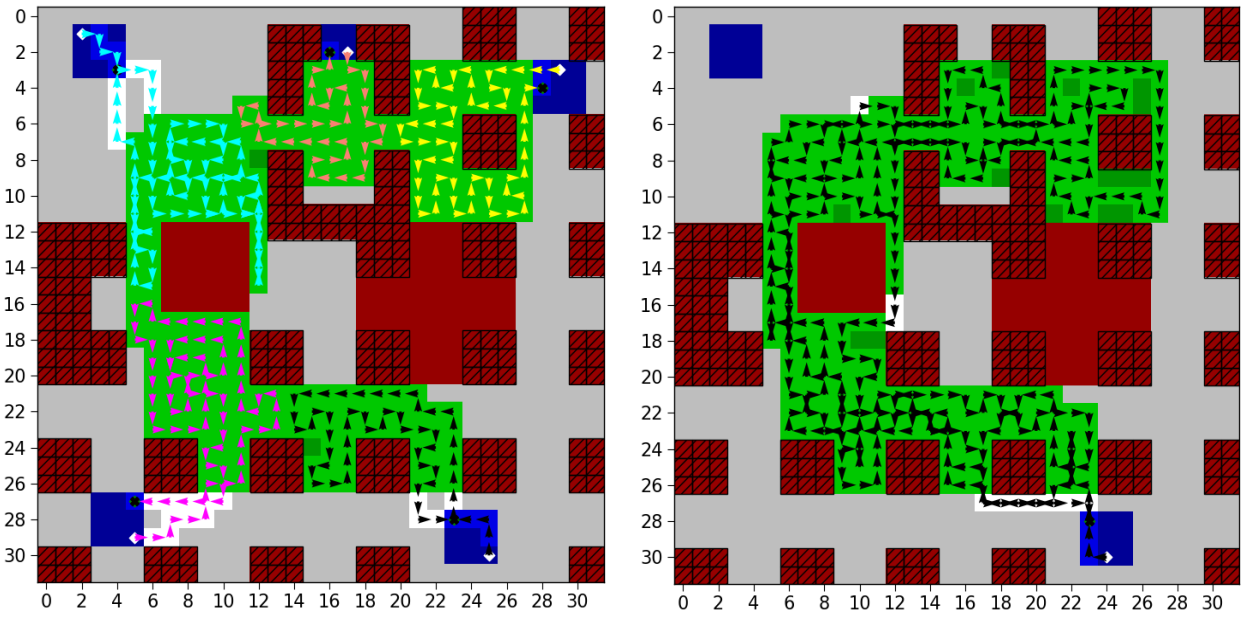
Figure 7. Effect graph using improved K-Means clustering.

Table 3. Average clustering effect after 100 experiments.

Metrics	Average Silhouette Coefficient	Calinski–Harabasz Index	Davies–Bouldin Index	Iteration
K-Means	0.375	91.59	0.843	7.16
K-Means++	0.396	93.78	0.792	4.66
K-MeansII	0.402	95.73	0.778	4.61
Improved K-Means	**0.412**	**101.790**	**0.770**	**4.58**

From the results of three indexes: average silhouette coefficient, Calinski–Harabasz Index and Davies–Bouldin Index, it is obvious that the improved algorithm proposed in this paper has higher clustering quality and better clustering distribution than the general clustering algorithm. At the same time, clusters are tighter internally and more separated from each other.

Finally, in the divided map, the improved algorithm in this paper is used for multi-robot coverage path planning. Each robot is responsible for a small divided area. The final renderings are shown in Figure 8 and Table 4. Figure 8a shows the effect of cooperative coverage of multiple mobile robots in this paper, and Figure 8b shows the effect of the algorithm in [24] using a single mobile robot for coverage.



(**a**) Multi-robots' coverage effect in "Manhattan 32". (**b**) Single-robot' coverage effect in "Manhattan 32".

Figure 8. Comparison of coverage effects between multi-mobile robots and single-mobile robot. Figure 8a is the result of multi-robot cooperation of the algorithm proposed in this article, and Figure 8b is the result of the algorithm in [24].

Table 4. Comparison of average results of 1000 experiments with single robot and multi-robot collaboration.

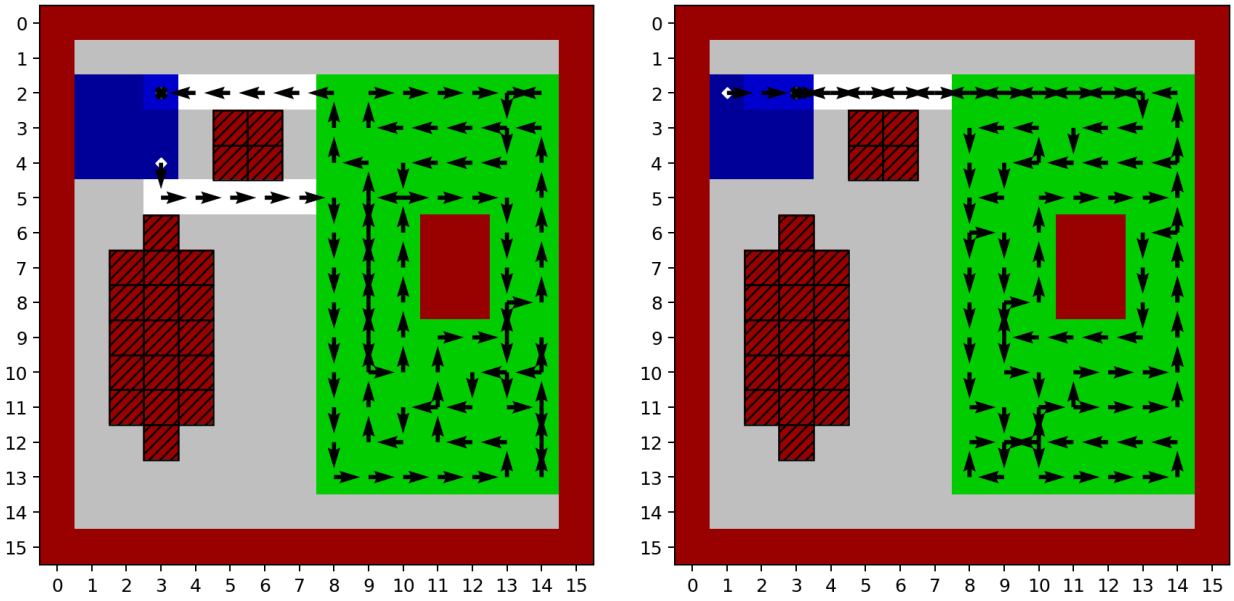
Metrics	CR	RR	Reached	CRAC
Multi-mobile robots	**0.990**	**1.180**	**0.989**	**0.98**
Single-mobile robot	0.690	1.460	0.985	0.682

The simulation results show that the coverage effect of multi-mobile robots is better than that of a single-mobile robot, the map is better divided and the coverage repetition ratio is reduced. In this paper:

1. The minimum spanning tree algorithm is introduced, and the improved K-Means clustering is used for map division in this paper, which makes the map division effect better, helps reduce the complexity of the environment, and improves the coverage effect;
2. The introduction of reward r_2 can prevent the mobile robot from entering a position that has been covered and has no uncovered area around it. At the same time, there will be guidance to guide the mobile robot to take action towards the uncovered area;
3. The dueling network is introduced to separate the state value function and the action advantage value function, which helps the mobile robot pay attention to the obstacles ahead and take actions to avoid them;
4. Using multiple mobile robots to complete the coverage task will help reduce the task complexity of each mobile robot and better complete the coverage task.

4.3. Discussion

In order to further verify that the reward r_2 proposed in this paper, that is, Equation (15), can effectively reduce the repetition ratio of robots in the coverage process. This paper selects the map in reference [18]. It adds and does not add reward r_2 to the algorithm proposed in this paper, it obtains the coverage path planning graph as shown in Figure 9, and it takes the average value after 1000 experiments to obtain the repetition ratio as shown in Table 5.



(**a**) Introducing the coverage map of r_2. (**b**) Coverage map without introducing r_2.

Figure 9. The impact of reward r_2 on coverage results. Figure (**a**,**b**) test the impact of r_2 on the RR index based on the main algorithm of this article.

Table 5. Comparison of the average results of 1000 experiments with and without r_2.

Metrics	CR	RR
Add reward r_2	**0.987**	**1.21**
Without reward r_2	0.98	1.32

As can be seen from Figure 9 and Table 5, after applying the reward r_2, not only the coverage ratio can be improved, but also the path repetition ratio can be greatly reduced. Obviously, on the premise that the coverage ratio is slightly improved, i.e., 0.07, the repetition ratio is reduced by 0.11, which shows that the algorithm in this paper can further reduce the repetition ratio. It reflects the effect of r_2 proposed in this paper.

From the boxplot provided in Figure 10, we can observe the difference in repetition rates between the two sets of data. Having r_2 exhibits a lower median repetition rate and a smaller data distribution range, which indicates that the consistency and reliability of coverage path planning are improved after considering the r_2 factor. The absence of r_2 shows a higher median repetition rate and a wider data distribution, which means the results are more variable. Therefore, based on the analysis of the boxplot, we can conclude that the coverage path planning after introducing the r_2 factor shows better performance.

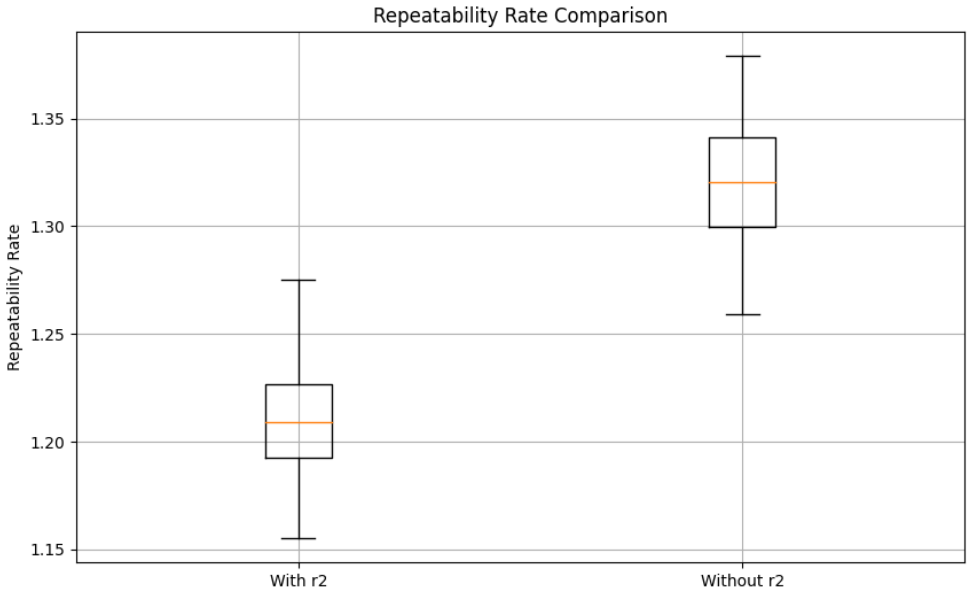


Figure 10. Repetition rate boxplot with and without r_2.

5. Conclusions and Future Work

In this paper, an improved K-Means clustering algorithm is proposed to re-divide the map areas that need to be covered. Then, the deep reinforcement learning and dueling network model is used and the reward function is improved so that the multi-mobile robots can cover the target area through cooperation. Through the above improvements and experimental results, we draw the following conclusions: (1) The clustering method used in this paper can further improve the clustering effect and provide a more reasonable basis for regional division for the collaborative coverage of multiple mobile robots; (2) the use of dueling networks and improved reward functions, combined with the collaboration of multiple mobile robots, can significantly improve coverage while reducing repeated coverage and coverage path length, thus improving the efficiency and quality of path planning. (3) Compared with a single robot, using multiple robots reduces the task complexity.

Of course, the use of mobile robots in this paper mainly focuses on their long running time, economy, practicality and safety. It gives up the advantage of the wide vision of the UAV. Therefore, the next part of the work is to combine the mobile robot in this paper with the UAV to further divide the coverage area in the map, use the mobile robot to load the UAV to the designated coverage area, and then use the UAV to patrol the designated area. This can take into account the advantages of the long running time of the mobile robot and the wide vision of the UAV. In addition, clustering and deep learning can be further combined in this process, for example, [44–46].

Author Contributions: Conceptualization, Y.G. (Yu Gu) and J.N.; methodology, Y.G. (Yu Gu); validation, J.N. and Y.G. (Yu Gu); writing—original draft preparation, Y.G. (Yu Gu); writing—review and editing, G.T., C.K. and Y.G. (Yang Gu). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (61873086), and the National Key R&D Program of China (2022YFB4703402).

Data Availability Statement: Data are contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

CPP	Coverage path Planning
DRL	Deep Reinforcement Learning
DQN	Deep Q-Network
DL	Deep Learning
RL	Reinforcement Learning
POMDP	Partially Observable Markov Decision Processes
DDQN	Double Deep Q-Network
CR	Coverage Ratio
RR	Repetition Ratio
CRAR	Coverage Ratio and Reached
SSE	Sum of Squared Errors

References

1. Fevgas, G.; Lagkas, T.; Argyriou, V.; Sarigiannidis, P. Coverage Path Planning Methods Focusing on Energy Efficient and Cooperative Strategies for Unmanned Aerial Vehicles. *Sensors* **2022**, *22*, 1235. [CrossRef]
2. Zhang, Q.; Li, C.; Lu, X.; Huang, S. Research on Complete Coverage Path Planning for Unmanned Surface Vessel. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Ordos, China, 27–28 April 2019; Volume 300, p. 022037. [CrossRef]
3. Zhang, C.; Yu, D. Research on complete coverage path planning for multi-mobile robots. In Proceedings of the 2022 China Automation Congress, Xiamen, China, 25–27 November 2022; Volume 6, pp. 291–296.
4. Hoeffmann, M.; Patel, S.; Bueskens, C. Optimal Coverage Path Planning for Agricultural Vehicles with Curvature Constraints. *Agriculture* **2023**, *13*, 2112. [CrossRef]
5. Yakoubi, M.A.; Laskri, M.T. The path planning of cleaner robot for coverage region using Genetic Algorithms. *J. Innov. Digit. Ecosyst.* **2016**, *3*, 37–43. [CrossRef]
6. Zhu, J.; Yang, Y.; Cheng, Y. SMURF: A Fully Autonomous Water Surface Cleaning Robot with A Novel Coverage Path Planning Method. *J. Mar. Sci. Eng.* **2022**, *10*, 1620. [CrossRef]
7. Ai, B.; Jia, M.; Xu, H.; Xu, J.; Wen, Z.; Li, B.; Zhang, D. Coverage path planning for maritime search and rescue using reinforcement learning. *Ocean. Eng.* **2021**, *241*, 110098. [CrossRef]
8. Peng, C.; Isler, V. Visual Coverage Path Planning for Urban Environments. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5961–5968. [CrossRef]
9. Xu, P.F.; Ding, Y.X.; Luo, J.C. Complete Coverage Path Planning of an Unmanned Surface Vehicle Based on a Complete Coverage Neural Network Algorithm. *J. Mar. Sci. Eng.* **2021**, *9*, 1163. [CrossRef]
10. Huang, K.C.; Lian, F.L.; Chen, C.T.; Wu, C.H.; Chen, C.C. A novel solution with rapid Voronoi-based coverage path planning in irregular environment for robotic mowing systems. *Int. J. Intell. Robot. Appl.* **2021**, *5*, 558–575. [CrossRef]
11. Shen, Z.; Agrawal, P.; Wilson, J.P.; Harvey, R.; Gupta, S. CPPNet: A Coverage Path Planning Network. In Proceedings of the OCEANS 2021: SAN DIEGO—PORTO, San Diego, CA, USA, 20–23 September 2021; pp. 1–5.
12. Schaefle, T.R.; Mohamed, S.; Uchiyama, N.; Sawodny, O. Coverage Path Planning for Mobile Robots Using Genetic Algorithm with Energy Optimization. In Proceedings of the 2016 International Electronics Symposium (IES), Denpasar, Indonesia, 29–30 September 2016; pp. 99–104.
13. Xu, N.; Zhou, W. Research on Global Coverage Path Planning of Picking Robot Based on Adaptive Ant Colony Algorithm. *J. Agric. Mech. Res.* **2023**, *45*, 213–216+221. [CrossRef]
14. Zhao, Y.; Shi, Y.; Wan, X. Path Planning of Multi-UAVs Area Coverage Based on Particle Swarm Optimization. *J. Agric. Mech. Res.* **2024**, *46*, 63–67. [CrossRef]
15. Kubota, N.; Fukuda, T.; Shimojima, K. Trajectory planning of cellular manipulator system using virus-evolutionary genetic algorithm. *Robot. Auton. Syst.* **1996**, *19*, 85–94. [CrossRef]
16. Ni, J.; Wang, X.; Tang, M.; Cao, W.; Shi, P.; Yang, S.X. An Improved Real-Time Path Planning Method Based on Dragonfly Algorithm for Heterogeneous Multi-Robot System. *IEEE Access* **2020**, *8*, 140558–140568. [CrossRef]
17. Kyaw, P.T.; Paing, A.; Thu, T.T.; Mohan, R.E.; Vu Le, A.; Veerajagadheswar, P. Coverage Path Planning for Decomposition Reconfigurable Grid-Maps Using Deep Reinforcement Learning Based Travelling Salesman Problem. *IEEE Access* **2020**, *8*, 225945–225956. [CrossRef]
18. Theile, M.; Bayerlein, H.; Nai, R.; Gesbert, D.; Caccamo, M. UAV Coverage Path Planning under Varying Power Constraints using Deep Reinforcement Learning. 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. 1444–1449. [CrossRef]
19. Ni, J.; Li, X.; Hua, M.; Yang, S.X. Bioinspired Neural Network-Based Q-Learning Approach for Robot Path Planning in Unknown Environments. *Int. J. Robot. Autom.* **2016**, *31*, 464–474. [CrossRef]
20. Zellner, A.; Dutta, A.; Kulbaka, I.; Sharma, G. Deep recurrent Q-learning for energy-constrained coverage with a mobile robot. *Neural Comput. Appl.* **2023**, *35*, 19087–19097. [CrossRef]

21. Almadhoun, R.; Taha, T.; Seneviratne, L.; Zweiri, Y. A survey on multi-robot coverage path planning for model reconstruction and mapping. *SN Appl. Sci.* **2019**, *1*, 847. [CrossRef]
22. Shen, X.; Zhao, T. UAV regional coverage path planning strategy based on DDQN. *Electron. Meas. Technol.* **2023**, *46*, 30–36. [CrossRef]
23. Xing, B.; Wang, X.; Yang, L.; Liu, Z.; Wu, Q. An Algorithm of Complete Coverage Path Planning for Unmanned Surface Vehicle Based on Reinforcement Learning. *J. Mar. Sci. Eng.* **2023**, *11*, 645. [CrossRef]
24. Theile, M.; Bayerlein, H.; Nai, R.; Gesbert, D.; Caccamo, M. UAV Path Planning using Global and Local Map Information with Deep Reinforcement Learning. In Proceedings of the 2021 20th International Conference on Advanced Robotics (ICAR), Ljubljana, Slovenia, 6–10 December 2021; pp. 539–546.
25. Ruan, G.; Chen, J.; Xu, F. Complete coverage path planning algorithm based on rolling optimization and decentralized predator-prey model. *Control. Decis.* **2023**, *38*, 2545–2553. [CrossRef]
26. Luo, Z.; Feng, S.; Liu, X.; Chen, J.; Wang, R. Method of area coverage path planning of multi-unmanned cleaning vehicles based on step by step genetic algorithm. *J. Electron. Meas. Instrum.* **2020**, *34*, 43–50. [CrossRef]
27. Li, L.; Shi, D.; Jin, S.; Yang, S.; Zhou, C.; Lian, Y.; Liu, H. Exact and Heuristic Multi-Robot Dubins Coverage Path Planning for Known Environments. *Sensors* **2023**, *23*, 2560. [CrossRef] [PubMed]
28. Latombe, J.C., Exact Cell Decomposition. In *Robot Motion Planning*; Springer: Boston, MA, USA, 1991; pp. 200–247. [CrossRef]
29. Choset, H. Coverage of known spaces: The boustrophedon cellular decomposition. *Auton. Robot.* **2000**, *9*, 247–253. [CrossRef]
30. Choset, H.; Pignon, P. Coverage Path Planning: The Boustrophedon Cellular Decomposition. In *Field and Service Robotics*; Zelinsky, A., Ed.; Springer: London, UK, 1998; pp. 203–209.
31. Acar, E.; Choset, H.; Rizzi, A.; Atkar, P.; Hull, D. Morse decompositions for coverage tasks. *Int. J. Robot. Res.* **2002**, *21*, 331–344. [CrossRef]
32. Han, Y.; Shao, M.; Wu, Y.; Zhang, X. An Improved Complete Coverage Path Planning Method for Intelligent Agricultural Machinery Based on Backtracking Method. *Information* **2022**, *13*, 313. [CrossRef]
33. Shi, W.; Huang, H.; Jiang, L. Multi-robot Path Planning for Collaborative Full- Coverage Search in Complex Environments. *Electron. Opt. Control.* **2022**, *29*, 106–111.
34. Bao, C. K-means clustering algorithm: A brief review. *Acad. J. Comput. Inf. Sci.* **2021**, *4*, 37–40. [CrossRef]
35. Muhammad, A.; Sebastian, K. Potential applications of unmanned ground and aerial vehicles to mitigate challenges of transport and logistics-related critical success factors in the humanitarian supply chain. *Asian J. Sustain. Soc. Responsib.* **2020**, *5*, 1–22. [CrossRef]
36. Bradley, P.S.; Fayyad, U.M. Refining Initial Points for K-Means Clustering. In Proceedings of the International Conference on Machine Learning, Madison, WI, USA, 24–27 July 1998; Volume 98, pp. 91–99.
37. Arthur, D.; Vassilvitskii, S. K-Means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
38. Salvador, S.; Chan, P. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In Proceedings of the ICTAI 2004: 16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, USA, 15–17 November 2004; Khoshgoftaar, T., Ed.; pp. 576–584.
39. Yang, J.; Wang, Y.K.; Yao, X.; Lin, C.T. Adaptive initialization method for K-means algorithm. *Front. Artif. Intell.* **2021**, *4*, 740817. [CrossRef]
40. Shi, F.; Neumann, F.; Wang, J. Time complexity analysis of evolutionary algorithms for 2-hop (1,2)-minimum spanning tree problem. *Theor. Comput. Sci.* **2021**, *893*, 159–175. [CrossRef]
41. He, Z.; Pang, H.; Bai, Z.; Zheng, L.; Liu, L. An Improved Dueling Double Deep Q Network Algorithm and Its Application to the Optimized Path Planning for Unmanned Ground Vehicle. In Proceedings of the SAE 2023 Intelligent and Connected Vehicles Symposium, Nanchang, China, 22–23 September 2023. [CrossRef]
42. Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; de Freitas, N. Dueling Network Architectures for Deep Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Balcan, M.; Weinberger, K., Eds.; Volume 48, pp. 1995–2003.
43. Bahmani, B.; Moseley, B.; Vattani, A.; Kumar, R.; Vassilvitskii, S. Scalable k-means++. *Proc. VLDB Endow.* **2012**, *5*, 622–633. [CrossRef]
44. Dornaika, F.; El Hajjar, S. Single phase multi-view clustering using unified graph learning and spectral representation. *Inf. Sci.* **2023**, *645*, 119366. [CrossRef]
45. Borlea, I.D.; Precup, R.E.; Borlea, A.B. Improvement of K-means Cluster Quality by Post Processing Resulted Clusters. *Procedia Comput. Sci.* **2022**, *199*, 63–70. [CrossRef]
46. Mihalache, S.; Burileanu, D. Speech Emotion Recognition Using Deep Neural Networks, Transfer Learning, and Ensemble Classification Techniques. *Rom. J. Inf. Sci. Technol.* **2023**, *26* , 375–387. [CrossRef]