A Novel Automated Guided Vehicle (AGV) Remote Path Planning Based on RLACA Algorithm in 5G Environment

Wangwang Yu, Jun Liu* and Jie Zhou

School of Electrical Engineering, Shanghai DianJi University. Shanghai 201306, China
E-mail: ywwsnowboy@outlook.com; liujun@sdju.edu.cn; zhoujie2811@163.com
**Corresponding Author*

Received 27 July 2021; Accepted 12 August 2021;
Publication 06 November 2021

Abstract

Remote control and monitoring will become the future trend. High-quality automated guided vehicle (AGV) path planning through web pages or clients can reduce network data transmission capacity and server resource occupation. Many Remote path planning algorithms in AGV navigation still have blind search, path redundancy, and long calculation time. This paper proposed an RLACA algorithm based on 5G network environment through remote control of AGV. The distribution of pheromone in each iteration of the ant colony algorithm had an impact on the follow-up. RLACA algorithm changed the transfer rules and pheromone distribution of the ant colony algorithm to improve the efficiency of path search and then modify the path to reduce path redundancy. Considering that there may be unknown obstacles in the virtual environment, the path obtained by the improved ant colony algorithm is used as the training data of reinforcement learning to obtain the Q-table. During the movement, the action of each step is selected by the Q-table until the target point is reached. Through experimental simulation, it can be concluded that the enhanced ant colony algorithm can quickly obtain a

Journal of Web Engineering, Vol. 20_8, 2491–2520.
doi: 10.13052/jwe1540-9589.20813

reasonable and adequate path in a complex environment and effectively avoid unknown obstacles in the environment.

Keywords: 5G, path planning, ant colony algorithm, reinforcement learning, path correction.

1 Introduction

With the development of 5G networks, planning a more intelligent, faster and more efficient path in the logistics field is a hot issue of continuous research. Automated guided vehicle (AGV) gradually plays an essential role in medical treatment, transportation and life. One of the most critical functions of AGV is path planning. Path planning has been widely used in various aspects, such as map software path planning, the movement of game characters, and the logistics and transportation fields. The quality of online path planning through web pages or software will affect network transmission speed, server resource usage and data transmission capacity. So far, many scholars have proposed many algorithms, such as Dijkstra algorithm [1], JPS algorithm [2], A* algorithm [3], artificial potential field method [4]. Many intelligent algorithms have emerged in the later period, such as ant colony algorithm [5], particle swarm algorithm [6], genetic algorithm [7, 8]. Many scholars apply these intelligent algorithms [9–11] to path planning and have achieved good results.

Ant colony algorithm [12] has strong robustness. It can be calculated in parallel and is easy to combine with other algorithms. In terms of path planning, ant colony algorithm has good results, but there are also premature, self-locking, and path redundancy. In response to these problems, many scholars have made improvements [13–17]. Luo [18] rewarded the optimal solution in each iteration, penalized the worst solution, and penalized the last two steps of the deadlock path for avoiding ant deadlock as much as possible. Wu [19] applied the rollback and death strategy to the maximum and minimum ant system. Yao [20] proposed an HFACO algorithm, which defines two different types of ants: exploration Ant A and exploitation Ant B, by dynamically adjusting the two to solve the premature problem of ant colony algorithm. Jiao [15] proposed the polymorphic ant colony algorithm, in which each ant is assigned a different work, and the deadlock is penalized for making the ant move in the direction without obstacles. Liu [21] combined pheromone diffusion and local geometric optimization, spread the path pheromone along the direction of the potential field to improve the higher fitness subspace, optimized the path using ant colony algorithm, and optimized the geometric algorithm to avoid cross path. Zhao [22] improved the initial pheromone distribution's uneven and directional selection strategy and introduced the pheromone coverage and update strategy to avoid repeated searches. Xia [23] applies the improved quantum ant colony algorithm to path planning. Local and global update rules update the quantum ant colony algorithm. K. Akka and F. Khaber [24] improved the transition probability function to increase the probability of ants moving to places with few obstacles. Improve the dynamic volatilization coefficient to prevent the ant colony algorithm from falling into local convergence. Wang [25] combined Bayesian decision-making with ant colony algorithm and introduced the environmentally affected side of the successor node into the decision of successor node selection, limiting the algorithm's optimization in the poor solution area and making the algorithm more optimal in the superior solution area. Sufficient while avoiding falling into the optimal local solution. Dai [26] combined the ant colony algorithm and the A* algorithm to improve the convergence speed of the path and roll back deadlocked ants until a movable path was found, which avoided the deadlock of the ants. Li [27] improved the environment model and improved the safety and smoothness of path search. Local path planning and global pheromone update can improve the real-time performance of pheromone and improve global searchability. Introduce the idea of cat colony, adopt dynamic grouping, redistribute ant colony, and improve algorithm diversity.

Many algorithms can improve the ability of the ant colony algorithm and increase the convergence speed. The improved path still has path redundancy and low search efficiency. In the AGV path planning system, although the network calculation speed will be increased, the path will not be the shortest, which will affect the efficiency of logistics and transportation.

Reinforcement learning is a type of machine learning. Reinforcement learning influences the subsequent decision-making of the agent through the interaction between the agent and the environment. Reinforcement learning has been widely used in various fields. Noguchi [28] combined the artificial potential field method and reinforcement learning for I-AUV to plan a safe path to capture sea urchins. Liu [29] combines reinforcement learning and robot path planning to solve practical problems. Qu [30] combines reinforcement learning and grey wolf algorithm for UAV path planning, and the planned path is shorter and smoother than the path planned by traditional algorithms. Leiva [31] proposed a robust approach to train map-free navigation strategies to improve the flexibility of dynamic sensor observation.

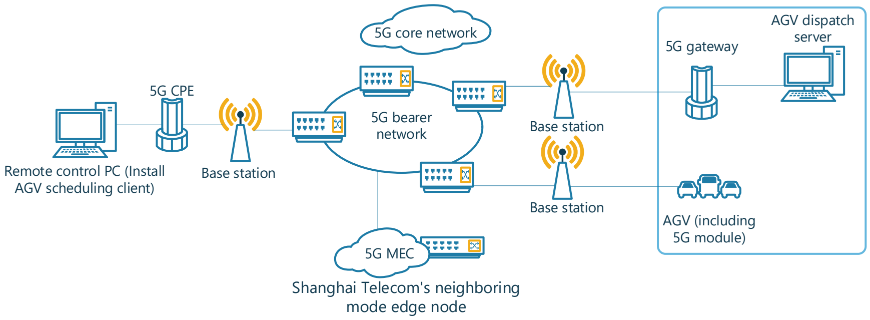

5G core network
5G CPE
5G bearer network
Base station
Base station
Base station
Remote control PC (Install AGV scheduling client)
5G MEC
Shanghai Telecom's neighboring mode edge node
5G gateway
AGV dispatch server
AGV (including 5G module)

Figure 1 AGV remote control diagram.

Jian [32] proposed an ART2 neural network and its learning algorithm based on reinforcement learning, reducing the number of collisions between robots and obstacles and effectively avoiding obstacles. Multiple experiments have shown that reinforcement learning has been widely used in robotics.

This paper is aimed at the problem of remote control AGV path planning in 5G network environment. As shown in Figure 1, the AGV dispatch client is installed in the terminal, and the instruction data flow is uploaded through a 5G gateway (5G CPE) and diverted to the AGV's dispatch server through the edge node. The scheduling server calculates the globally optimal path and uploads the relevant instruction data flow through the 5G gateway (5G CPE), diverts to the AGV through the edge node, and the AGV selects the current optimal path according to the actual environment and the planned global optimal path, thus reaching the target point.

In order to improve the shortcomings of ant colony algorithm in remote path planning, this paper proposed an AGV path planning based on the RLACA algorithm. Reinforced ant colony algorithm combines the advantages of reinforcement learning and improved ant colony algorithm. Compared with other ant colony algorithms, the difference of the improved ant colony algorithm is that reinforcement learning is added to integrate global and local path planning to realize the unification of global path planning and local path planning. At the same time, considering the shortcomings of blindly searching for the optimal path, changing the initial pheromone distribution and changing the pheromone update rules avoids the locally optimal path. In order to ensure the stability of the path, increase the smoothing coefficient and the target point trend coefficient, and modify the path to reduce the redundancy of the path. Considering that unknown obstacles and collisions need to be avoided in the real environment, all the paths to the target point are learned through reinforcement learning, and each movement action is selected through the Q value obtained through learning. The main contributions of this paper are as follows:

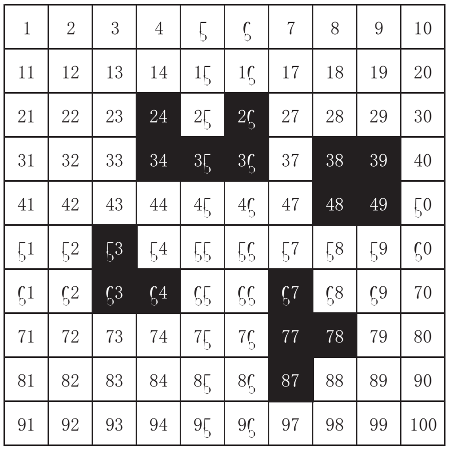
(1) A new type of reinforced ant colony algorithm is proposed, which combines the characteristics of reinforcement learning and ant colony algorithm to solve the problems of AGV system in path planning.
(2) The initial pheromone distribution is improved in the RLACA algorithm, reducing the blindness of the ant colony algorithm search. The randomness of path selection in ant pathfinding will lead to redundancy in the path. By introducing the smoothing coefficient and the target point trend coefficient, the stability of the path is ensured, and the path is kept moving towards the target point. Improve the pheromone update rules for the ants that reach the target point so that the path pheromone distribution has a specific gradient, which improves the search diversity and avoids premature maturity. During the search process, the redundant path due to environmental factors is added to the global taboo table to prevent the ants from continuing to move to this path in the later stage, making the path search more efficient and intelligent.
(3) The modified path to the target point in the improved ant colony algorithm is used as the training number for updating the Q-table. During the movement of the AGV, the direction of the next move is selected by grid map and actual environment to avoid the existence of the environment Unknown obstacles.

The structure of this article is as follows. The second section is built the environment model by web or software, basic ant colony algorithm and reinforcement learning algorithm of AGV. The third section describes the specific implementation of the RLACA algorithm. The fourth section describes the implementation steps and flowchart of RLACA. The fifth section compares the RLACA algorithm with other algorithms. Section VI summarizes the RLACA algorithm.

2 Basic Knowledge and Ideas

2.1 Environment Modelling

The establishment of the model needs to be consistent with the actual situation to help solve the actual problem. In the environment where the AGV is operating, the grid map is more conducive to realizing the path planning algorithm, and the grid graph is also more conducive to the deletion and



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Figure 2 Grid map.

modification of the environmental map. Therefore, the grid map is selected as the simulation map. The moving direction of the AGV is divided into eight directions: front, back, left, right, left front, right front, left back, and right back, corresponding to the eight grids around a grid in the grid map.

By converting the two-dimensional environment map into a grid map, the mobile robot moves in the $MM \times MM$ grid environment, and each square in the grid map is sorted from left to right and top to bottom. In the map, black represents obstacles, and white represents areas that can move freely. A grid is represented by (x, y) its coordinates, and is represented by Equation (1):

$$\begin{cases} x = mod(i, MM) - 0.5 \\ y = MM + 0.5 - ceil\left(\dfrac{i}{MM}\right) \end{cases} \tag{1}$$

Where x is the abscissa, y is the ordinate, mod is the remainder function, and i is the serial number, which is shown in Figure 2. MM is the length of the grid map, and $ceil$ is rounded to infinity.

2.2 The Basic Principle of Ant Colony Algorithm

The ant colony algorithm is a swarm intelligence algorithm that simulates the pheromone left in the pathfinding process of ants and selects the next node through the generated pseudo-random number combined with the state transition probability until the target point is reached or there is no node to move. After a single ant release is over, the path is updated with pheromone, and as the ants continue to search, an optimal path is finally converged.

2.2.1 Probability transfer function

In the basic ant colony algorithm, the pheromone and distance on the path determine the probability transition function. In the kth iteration, the current transition probability $p_{ij}^{k}(t+1)$ of ant from node i to node j is calculated as follows:

$$P_{ij}^{k}(t+1) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}{\sum_{J=A_k(i)}[\tau_{ij}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, & j \in J_k(i) \\ 0, & \text{others} \end{cases} \tag{2}$$

Among them, τ_{ij} is the pheromone concentration between node i and node j. η_{is} is the visibility factor, usually the reciprocal of the distance between node i and target point j, and α and β respectively reflect the importance of pheromone concentration and visibility factor η_{ij}.

The transition of the ant colony algorithm path is random, without considering the smoothness of the path, and at the same time, the degree of approaching the target point is relatively low. These shortcomings will affect the distribution of pheromone and the speed of convergence.

2.2.2 Pheromone update

When the ant completes a single iteration, the pheromone is updated once, and the pheromone update equation is calculated as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \triangle\tau_{ij} \tag{3}$$

$$\triangle\tau_{ij} = \sum_{k=1}^{m} \triangle\tau_{ij}^{k} \tag{4}$$

$$\triangle\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k}, & \text{when ant k passes edge ij in this move} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Among them, $\tau_{ij}(t+1)$ represents the pheromone concentration, ρ represents the pheromone volatilization coefficient, $\triangle\tau_{ij}^{k}$ represents the pheromone increment in the kth iteration from node i to node j, and m represents the number of ants released each time.

The pheromone update of the traditional ant colony algorithm is affected by the number of moving ants on a specific path ij. The more the number of moves, the larger the $\triangle\tau_{ij}$, and more ants will follow the path, which is prone to local convergence.

2.3 Reinforcement Learning Algorithm

Reinforcement learning is a kind of machine learning and has been widely used in various fields. Reinforcement learning can be divided into learning methods based on strategy and value. The Q-learning algorithm is a value-based learning algorithm. In the learning process, the agent chooses the corresponding action by the current state to estimate the optimal strategy. The Q-score table is dynamically updated according to the reward. The update equation is as follows:

$$Q_{t+1}(s_t, a_t) \leftarrow Q(s_t, a_t) + \delta \cdot [r_{t+1} + \gamma \cdot max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{6}$$

Where s_t represents the current state, s_{t+1} represents the next state, a_t represents the current action taken, and r_{t+1} represents the reward corresponding to the next state based on the corresponding action a_t. δ is the learning rate, and γ is the discount factor. $max_a Q(s_{t+1}, a)$ is the next state s_{t+1}. Select action a to obtain the maximum Q value. The pseudocode of the Q-learning algorithm is as follows:

The Q-learning algorithm pseudocode

```
Initialize $Q(s, a)$
Randomly choose an initial state $s_t$
while the terminal is not reached do
    Choose $a_t$ from $s_t$ using policy derived from Q-table
    Tack action $a$, observe reward
    Find the new state $s_{t+1}$
    Observe the maximum Q-table of $s_{t+1}$
    Update the Q-table by Equation (6)
    Update the state $s_t \leftarrow s_{t+1}$
end while
```

Reinforcement learning is a blind search of the environment. This search method does not reinforce the optimal solution in history, and the optimal historical path is not necessarily the global optimal path. The ant colony algorithm will strengthen the optimal solution in the search path and can make up for the shortcomings in reinforcement learning. Therefore, the RLACA algorithm is proposed, which combines the ant colony algorithm and the reinforcement learning algorithm to achieve the path planning algorithm's global and local path planning performance.

3 RLACA Algorithm

3.1 Improve Initial Pheromone Distribution

In the traditional ant colony algorithm, the initial path pheromone concentration in every grid is equal, and path selection is blind. In a more complex environment, the number of ants will be lost. If the pheromone is updated and the number of ants in each iteration is unreasonable, It is easy to make the ant colony algorithm premature and can not get the optimal path under actual conditions. Given the shortcomings in this part, a non-uniform initial pheromone distribution method is proposed. The pheromone distribution of each part is as follows:

$$\tau_{ij}(0) = \mu\left(\frac{a * \tau_{0\_point} + b * \tau_{0\_line}}{a * \tau_{0\_point}(max) + b * \tau_{0\_line}(max)}\right) \tag{7}$$

$$\tau_{0\_point} = l_{t\_se} - l_{t\_ie} \tag{8}$$

$$\tau_{0\_line} = 1/(l_{line} + c) \tag{9}$$

Among them: μ is the pheromone distribution coefficient, which is a constant, l_{t_se} is the Euclidean distance between the starting point and the target point in the pheromone matrix, and l_{t_ije} is the Euclidean distance from point i to the target point in the pheromone matrix Distance, l_{line} is the distance between the current point in the pheromone matrix and the line between the starting point and the target point, a, b, and μ are the proportional coefficients. c is the offset, which is a constant.

Figure 3 shows the initial pheromone distribution in a 30 × 30 environment. In Figure 3, the coordinate (1,1) is the starting point, and (30,30) is the target point. It can be seen that the pheromone concentration of the pheromone to the target point is higher, which makes the ant's pathfinding process affected by uneven pheromone. Move continuously to the target point. The pheromone distribution method can reduce the blind search in the basic ant colony algorithm and improve search efficiency.

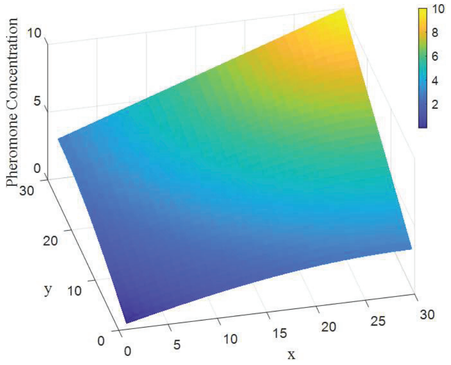


Figure 3 Schematic diagram of pheromone distribution.

3.2 Improve Transfer Rules

In the basic ant colony algorithm, the introduction of heuristic factors and pheromone can make the ants have a greater probability of moving to where the pheromone concentration is high and the target point. Since the pheromone concentration of adjacent nodes has little difference, the path tortuosity rate is high. It can continuously move to the target point while improving the stability of the path. Based on the traditional transfer function, the stability coefficient φ_{sta} and the target point trend coefficient φ_{ori} are added.

$$p_{ij}^{k}(t) = \begin{cases} {\varphi_{sta}}^{\xi}{\varphi_{ori}}^{\epsilon}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}{\sum_{S=J_k(i)}[\tau_{ij}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, & j \in J_k(i) \\ 0, & \text{others} \end{cases} \tag{10}$$

$$\varphi_{sta} = \begin{cases} \dfrac{1+\cos(180^\circ - \varepsilon)}{2}, & \text{current point} = \text{start point} \\ 1, & \text{current point} \neq \text{start point} \end{cases} \tag{11}$$

$$\varphi_{ori} = \frac{1+\cos(\omega)+0.01}{2} \tag{12}$$

Among them: ξ is the smoothing factor, and ϵ is the offset factor, which reflects the smoothness of the movement and the relative importance of the target point. 0.01 avoids setting the trend coefficient of the movable direction away from the target point to 0.

In Figure 4, point P is the previous node, Q is the current node, R is the next node that can be selected, and point T is the target node.

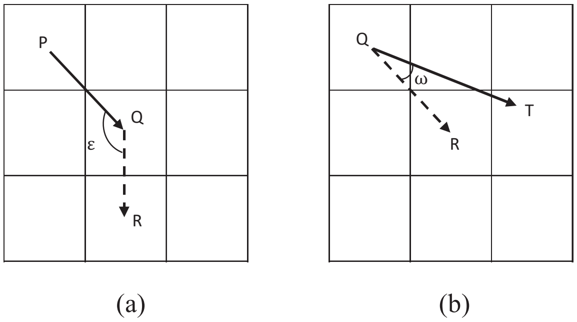


Figure 4 Schematic diagram of transfer rules.

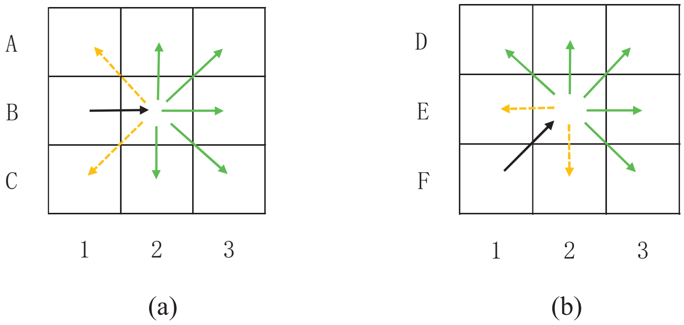


Figure 5 Schematic diagram of the moving direction.

Changing the proportion of the transfer function in the path selection makes the path both stable and at the same time able to move faster to the target point. When the current point in Equation (11) is at the origin, there is no previous node, so the stability coefficient is 1, which will not affect the choice of the path. The 0.01 in Equation (12) is to prevent the value of φ_{ori} from being 0 and make it effective, The node loses the transfer opportunity.

By analyzing the path of the ant colony algorithm, the path of the ant colony algorithm has fallbacks and deadlocks because some nodes will cause the fallback. As shown in Figure 5, B2 and E2 are the current points, and B1 and F1 are In the previous node, the black arrow indicates the direction that has been moved. All the movable directions are divided into the forward direction and the escape direction. The green arrow indicates the forward direction, and the yellow arrow indicates the escape direction. Set A1, C1, E1, and F2 as the escape point. When encountering obstacles, all the movement in the forward direction is invalid, then you can move to the escape point.

By improving the transition probability, the movement of the ant can be affected by the target point, moving direction, and the smoothness of the path can be taken into account while moving toward the target point.

3.3 Change Pheromone Update Rule

In the process of finding a path, the ant deviates from the start-end line (the line between the start point and the target point). The longer the path is. The ant colony algorithm is sorted according to the length of the path. After sorting, the pheromone is updated according to the length of the path so that the distribution of pheromone in the map has a specific gradient, which can effectively avoid the premature ant colony algorithm and fall into the local optimum. Improve the diversity of paths, and enhance the ability to search for the optimal solution globally. The equation for the improved pheromone update rule is as follows:

$$\tau_{ij}(t) = (1-\rho)\tau_{ij}(t-1) + \triangle\tau_{ij} \tag{13}$$

$$\triangle\tau_{ij}^{k} = \begin{cases} \dfrac{N_{eff} - m + 1}{N_{eff}} \cdot \dfrac{Q}{L_k}, & \text{when ant k passes edge ij in this move} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

Among them: N_{eff} is the number of ants reaching the target point in this iteration, and m is the sequence number of the shortest path of the ant in this iteration. L_k is the length of the path.

3.4 Improve the Taboo List

The function of the traditional taboo list is to prevent the ants from walking again and reduce the efficiency of pathfinding. Since the transition probability of ants is random, the path may be self-locked, which will affect the update of pheromone and reduce the number of effective ants. To solve this problem, increase the global taboo table and combine it with the local taboo table. When there is no alternative path other than the escape point during the movement, or even the escape point cannot be transferred, the transfer path is recorded into the global taboo table to prevent the ants in the later period once again entered the misunderstanding.

As shown in Figure 6(a), K1 is the previous node, K2 is the current node, and J1 and L1 are the next nodes that can be selected. Since K1→K2→J1/L1

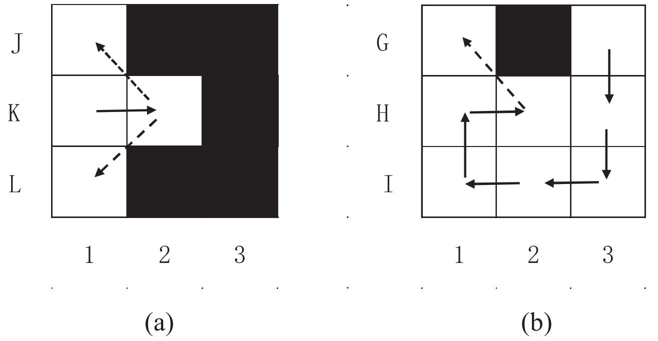


Figure 6 Path selection example diagram.

is a redundant path, this should not be considered at the K1 node. By updating the taboo table, set the path K1→K2 as an immovable path. Subsequent ants will not continue to walk this path. If the following equation is satisfied in the path selection, the path from the previous node to the current node is added to the global taboo table.

$$T = m + 1 \tag{15}$$

T: The number of directions in which the point in the taboo table can move.

m: Actual number of directions that can be moved.

The number 1 in Equation (14) represents the path from the previous node to the current node. The global taboo table is updated in real-time with the path of the lost ant and not updated with the update of the pheromone, which can prevent the ants from repeatedly entering the wrong zone in the later period. The role of the local taboo table is to avoid the ants from repeating the path that has been taken. Combining the two can effectively reduce the number of ants lost in the complex environment, and at the same time, it can adaptively adjust the environment to make the ants more intelligent in their pathfinding. The situation is shown in Figure 6(b). conforms to the equation, K1→K2 is added to the global taboo table, and the path H1→H2 in Figure 6. does not meet the conditions for joining the global taboo table.

3.5 Path Correction

In the traditional ant colony algorithm, since the probability of each transition is generated by pseudo-random numbers, it has strong randomness. The path selected in this way will have back-off, redundancy and even self-locking

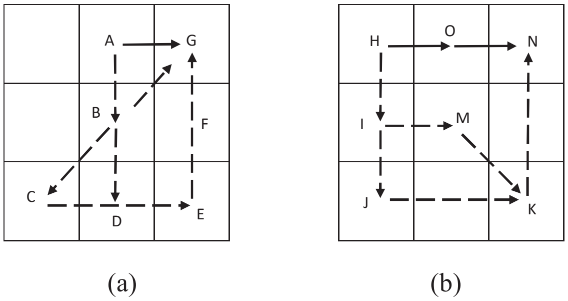


Figure 7 Path correction diagram.

phenomena. The redundant path is also It will interfere with the update of pheromone and even cause local convergence. The method of path correction is adopted to correct the redundant path to the target point. The path correction method is divided into the following two cases.

1. There is a redundant path between two adjacent nodes, as shown in Figure 7(a). That is, if Equations (16) or (17) is satisfied between any two non-adjacent nodes in the path that has moved, it can be judged that there is redundancy between these two points.

$$|x_1 - x_2| + |y_1 - y_2| = 1 \tag{16}$$

$$|x_1 - x_2| + |y_1 - y_2| = \sqrt{2} \tag{17}$$

   In Figure 7, node A and node G are adjacent paths, and there are multiple paths from node A to node G, such as A→B→G; A→B→C→D→E→F→G; →→A→B→ D→E→F→G. These paths have many redundant nodes, and these paths are directly replaced with A→G.

2. There is a movable node between the two nodes, as shown in Figure 7(b). Any two non-adjacent nodes (x_3, y_3) and (x_4, y_4) of path point must satisfy Equation (18, 19)

$$|x_3 - x_4| + |y_3 - y_4| = 2 \tag{18}$$

   And the middle point (x_5, y_5) satisfies

$$MAP(x_5, y_5) = 1 \tag{19}$$

3.6 Improve Reinforcement Learning

Considering that there may be unknown obstacles in the environment where the AGV moves, the AGV will collide with the obstacles during the movement, which will damage the equipment. In the traditional Q-learning algorithm, the path of each movement in path planning may not be the shortest, and the path of each movement can reach the target point, but it cannot achieve global sub-optimal and local optimal path planning. In order to avoid this situation happens by using the characteristics of the ant colony algorithm: the shorter the path, the more ants move on this path. Each path to the target point in the ant colony algorithm is used as the training data for reinforcement learning, and the Q value corresponding to the shortest path is continuously strengthened.

In the RLACA algorithm, the current position is taken as the current state, the direction of movement is selected as the action, and the reward value r is calculated by the Equations (20–22):

$$discur = \sqrt{(x_{cur} - x_{goal})^2} - \sqrt{(y_{cur} - y_{goal})^2} \tag{20}$$

$$disnex = \sqrt{(x_{nex} - x_{goal})^2} - \sqrt{(y_{nex} - y_{goal})^2} \tag{21}$$

$$r = discur - disnex \tag{22}$$

Where *discur* represents the Euclidean distance from the current position to the target point, and *disnex* represents the Euclidean distance from the next position to the target point. x_{cur}, y_{cur} represent the abscissa and ordinate of the current position, respectively. x_{nex}, y_{nex} respectively represent the abscissa and ordinate of the next position. In the beginning, all actions that movable positions have in Q-table are set to 1, and the actions that non-movable positions have are set to -INF to avoid interference with path selection in local path planning after training.

The pseudocode to improve the Q-learning algorithm is as follows:

The Improved Q-learning algorithm pseudocode

```
for k=1 to n do
    Determine current location $s_t$ and next location $s_{t+1}$
    Calculate action $a_t$ and reward $r$
    Observe the maximum Q-value of $s_{t+1}$
    Update the Q-table by Equation (6)
    Update the state $s_t \leftarrow s_{t+1}$
end for
```

4 RLACA Implementation Steps and Flowchart

Step 1: Create a map model, initialization parameters, initial pheromone, heuristic information, global taboo table, Q-table.

Step 2: Initializes the minimum path in this iteration.

Step 3: Initialize the ant state parameters and initialize the local taboo table at the same time to calculate the path and the number of paths that can be moved in the next step.

Step 4: Determine whether the current node is not the target point and whether the number of movable paths in the next step is greater than 1. If it is, go to step 5; otherwise, go to step 8.

Step 5: Determine whether the current point is the initial point. If it is, go to step 7; otherwise, go to step 6.

Step 6: Divides the nodes that can be moved in the next step into escape points and non-escape points and judge whether there is a non-escape point. If there is, use the non-escape point as a new movable node. If not, use the escape point as a new movable node. Move the node and add the path to the global taboo table.

Step 7: Calculate the stationary coefficient, target point trend coefficient and transition probability, select the next node by pseudo-random, add the node to the local taboo table, and use it as the current node, calculate the nodes that can be moved in the next step and the number of nodes, and execute the steps 4.

Step 8: Determine whether the path reaches the target point. If it reaches the target point, correct the path, calculate the length of the path, calculate the reward value r, update the Q-table, and record the path. If the target point is not reached, the length of the path is set to 0.

Step 9: Determines whether the upper limit of the number of ants released this time has been reached. If it is reached, go to step 10; if not, go to step 3.

Step 10: Sorts the path to the target point from small to large and updates the pheromone according to the sequence number.

Step 11: Determine whether the number of iterations reaches the upper limit. If not, perform step 2; otherwise, end the search and save the optimal path.

Step 12: According to the current position and the trained Q-table, select a path to avoid unknown obstacles until reaching the target point.

The algorithm flow chart is shown in Figure 8:

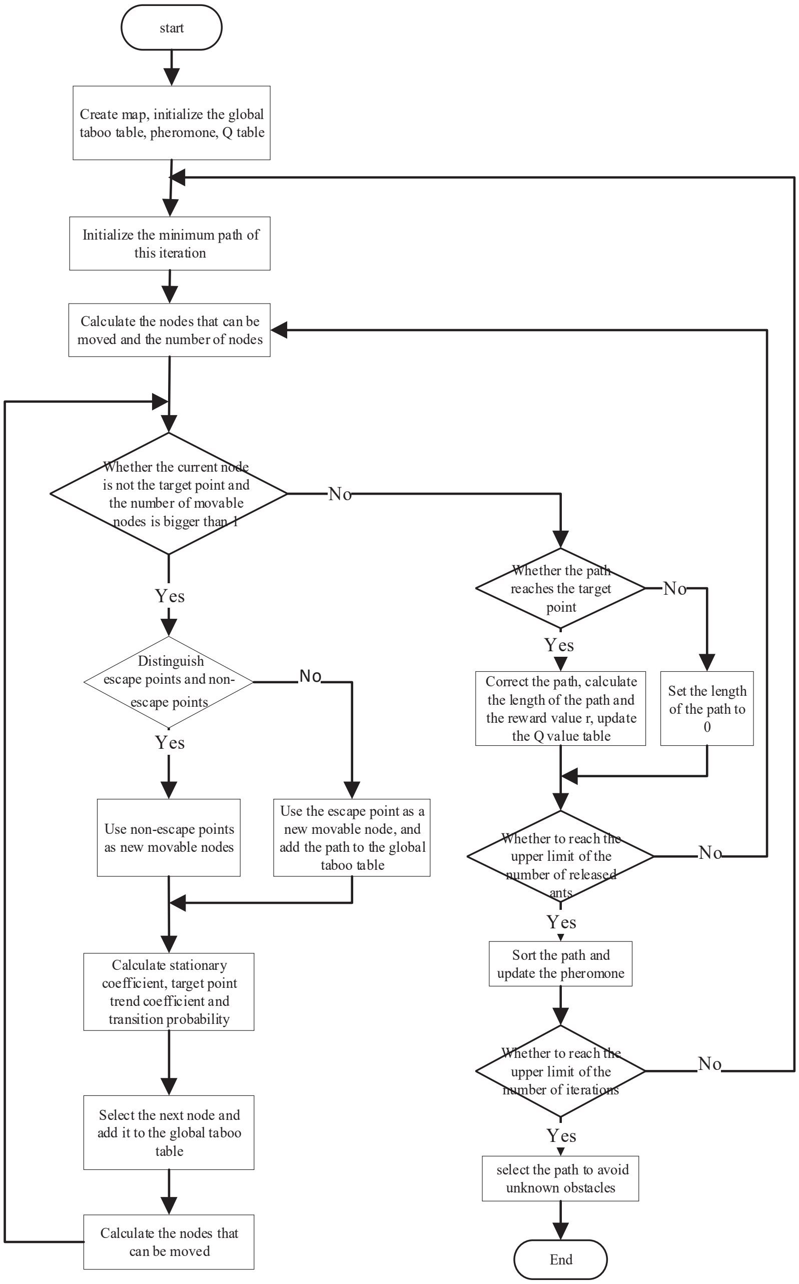

start
Create map, initialize the global taboo table, pheromone, Q table
Initialize the minimum path of this iteration
Calculate the nodes that can be moved and the number of nodes
Whether the current node is not the target point and the number of movable nodes is bigger than 1
No
Yes
Distinguish escape points and non-escape points
No
Yes
Use non-escape points as new movable nodes
Use the escape point as a new movable node, and add the path to the global taboo table
Calculate stationary coefficient, target point trend coefficient and transition probability
Select the next node and add it to the global taboo table
Calculate the nodes that can be moved
Whether the path reaches the target point
No
Yes
Correct the path, calculate the length of the path and the reward value r, update the Q value table
Set the length of the path to 0
Whether to reach the upper limit of the number of released ants
No
Yes
Sort the path and update the pheromone
Whether to reach the upper limit of the number of iterations
No
Yes
select the path to avoid unknown obstacles
End

Figure 8 Flow chart of RLACA algorithm.

5 Experimental Simulation and Result Analysis

In order to verify the feasibility of the improved ant colony algorithm, a large number of experimental simulations were carried out. The algorithm's performance was analyzed in both global and local path planning by comparing the traditional ant colony algorithm and other documents' improved ant colony algorithm. The running environment of this algorithm is: CPU is Intel(R) Xeon(R) E5-2450H @ 2.10GHz; graphics card is GTX 1050 Ti; Windows 10 64bit; 32G memory; simulation software: Matlab2020b.

5.1 Global Path Planning Aspects

In order to verify the feasibility of the algorithm, reduce contingency factors, and avoid errors in the judgment of the algorithm performance by the grid map, three different grid maps of 30 × 30 (Environment 1), 50 × 50 and 60 × 60 were routed 20 times. Planning and statistical analysis of operating results. Set the RLACA parameter settings as shown in the table through multiple debugging options:

Table 1 Parameter selection of RLACA

Parameter	Symbol	Value
Smoothing factor	ξ	1.4
Offset factor	ϵ	1.6
Volatility coefficient	ρ	0.9
Heuristic factor	α	1
Expected heuristics	β	7
Number of ants released each time	M	100
Learning rate	δ	0.9
Discount factor	γ	0.7

Where IACA 1, IACA 2 are the improved ant colony algorithm in paper [33] and paper [18]. RLACA strengthens the ant colony algorithm for this article.

Through Figures 9, 11, and 13, compared with other algorithms, RLACA has a lower convergence value and requires fewer iterations than other algorithms. The convergence value of the RLACA algorithm in high-complexity scenarios is similar to the ASrank, and RLACA has a shorter convergence time. Other algorithms have strong randomness in their movement, resulting in the convergence path is not the shortest. After the first iteration, the RLACA algorithm can get a shorter path due to path modification, which reduces path redundancy.

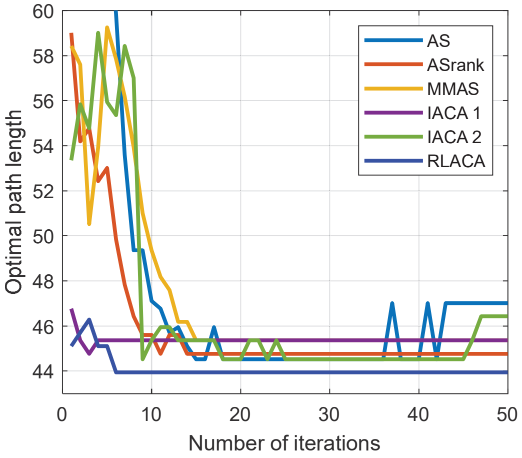


Figure 9 The change of the optimal path in the 30×30 environment with the number of iterations.

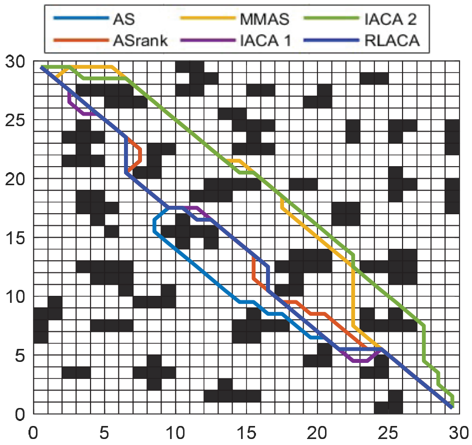


Figure 10 Road map of 30×30 environmental planning.

From Table 2, it can be seen that in a low-complexity environment, the path length planned by RLACA and other algorithms is not much different, but the IACA 1 algorithm tends to open safe space, which leads to its fast convergence speed, and it is difficult to find the shortest path. In a 50 × 50 environment, the difference between the RLACA convergence path and the ASrank convergence path is 1.85% of its own, but the ASrank convergence time is 2.12 times that of RLACA.

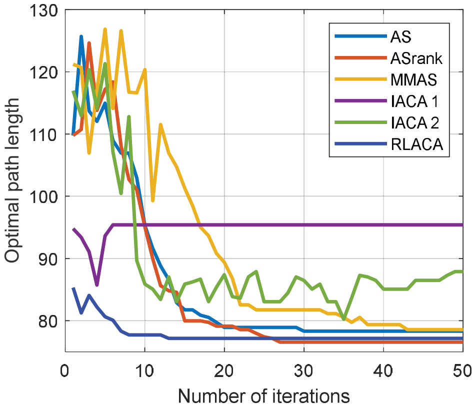


Figure 11 The change of the optimal path in the 50×50 environment with the number of iterations.

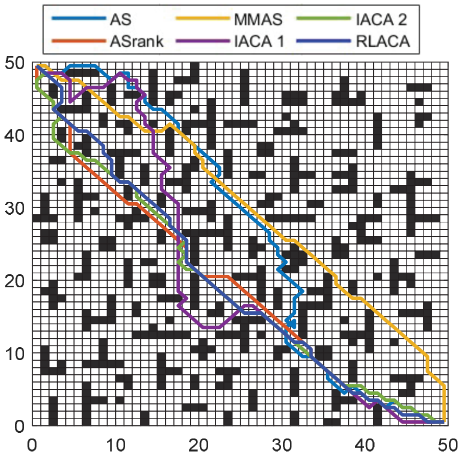


Figure 12 Road map of 50 × 50 environmental planning.

In summary, the RLACA algorithm has the shortest optimization path and shorter convergence time. Other improved algorithms have different degrees of precocity, the number of lost ants is large, and the convergence time is extended.

5.2 Partial Path Planning

In the RLACA algorithm, the path that can reach the target point obtained by the improved ant colony algorithm is used as the Q table training data.

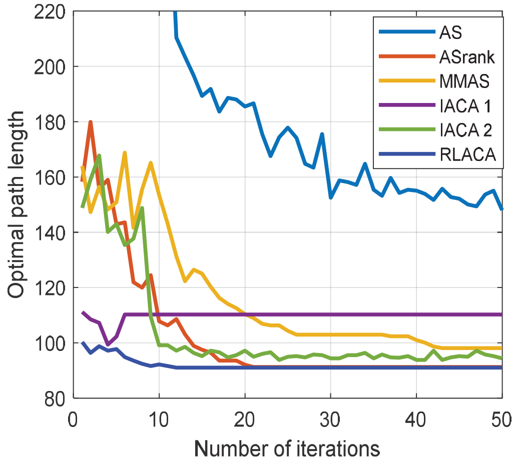


Figure 13 The change of the optimal path in the 60×60 environment with the number of iterations.

Table 2 Comparison of various algorithms in different environments

	Algorithms	Average Path Length	Best Path Length	Worst Path Length	Average Number of Ants Lost	Average Convergence Time (s)
30 × 30 Grid Environment	AS	45.15	44.53	46.43	381.9	11.43
	ASrank	45.99	43.94	77.77	122.9	17.45
	MMAS	44.85	43.94	45.94	247.8	20.81
	IACA 1	45.36	45.36	45.36	1.65	6.34
	IACA 2	45.27	44.53	45.94	171.9	15.61
	RLACA	**43.94**	**43.94**	**43.94**	**1.75**	**5.11**
50 × 50 Grid Environment	AS	90.08	84.2254	102.23	910.65	93.66
	ASrank	76.52	75.15	78.33	622.3	125.01
	MMAS	79.44	76.57	83.4	1090.45	271.11
	IACA 1	107.32	95.4	111.3	48.45	80.7
	IACA 2	83.98	78.57	87.05	610.85	93.93
	RLACA	**77.32**	**76.57**	**78.33**	**96.15**	**58.86**
60 × 60 Grid Environment	AS	163.59	121.2	241.18	1093.85	267.36
	ASrank	90.62	90.47	92.12	831	245.31
	MMAS	97.28	94.23	98.12	1319.7	539.32
	IACA 1	110.42	109.05	111.88	41.55	105.95
	IACA 2	97.72	94.71	99.64	654.2	110.68
	RLACA	**90.47**	**90.47**	**90.47**	**103.65**	**109.32**

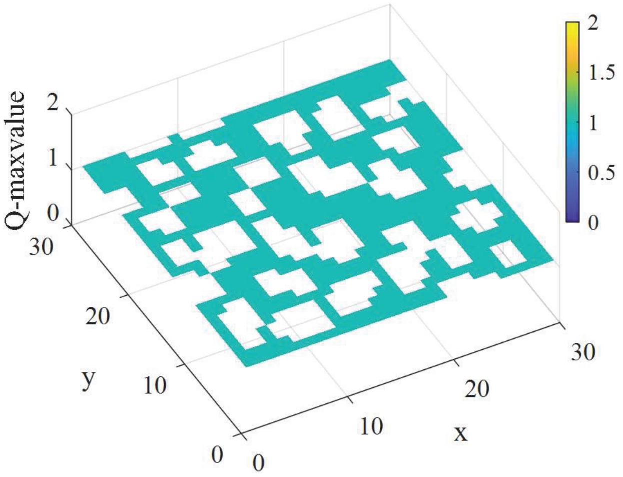


Figure 14 The Q value distribution corresponding to the maximum action of the Q table in Environment 1.

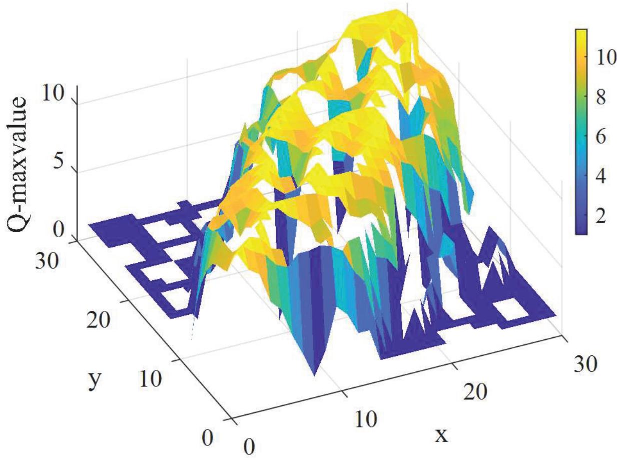


Figure 15 The Q value distribution corresponding to the maximum action of the Q table after the training in Environment 1.

The Q value changes are shown in Figures 14 and 15.

In Figures 14 and 15, the starting point is (1,1), the endpoint is (30,30), and the blank part corresponds to the obstacle's position. It can be seen that the RLACA algorithm for local path selection is distributed around the start and end line, and this distribution mode meets the needs of local path

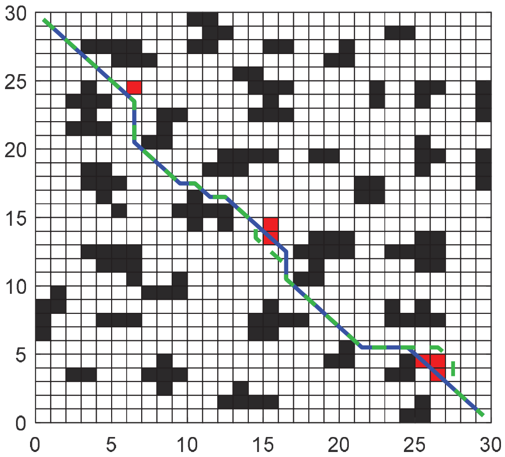


Figure 16 Comparison of global path planning and local path planning in a 30 × 30 grid environment.

planning in the virtual environment. The actual path movement in the local path planning needs to be moved in conjunction with the learned Q-table. When there are no obstacles on the globally optimal path, the path moved by the AGV according to the Q-table entirely coincides with the globally optimal path. When there are unknown obstacles on the global path that are not in the environment map, the action corresponding to the maximum Q value in the current state will be selected according to the Q-table.

In Figures 16, 17, and 18, the green is the moving path that simulates the actual environment, the blue is the globally planned path, and the red grid is the unknown obstacle. Through simulation, it can be seen that the RLACA algorithm can avoid unknown obstacles to reach the target point.

In summary, the advantage of RLACA over other algorithms is that it can converge to the optimal path in as short a time as possible. Improve the transfer rules, increase the stability coefficient and the target point guidance coefficient, improve the ant's guidance to the target point, and reduce the blind search path. Through path correction, redundant path pheromone update is avoided, the pheromone update efficiency is improved, and it is beneficial to find the optimal path. Increasing the taboo table reduces the number of lost ants and speeds up the path search efficiency. The RLACA algorithm integrates global path planning and local path planning, and the path selection in obstacle avoidance considers the global optimal while also considering the local optimal, which has higher practicability.

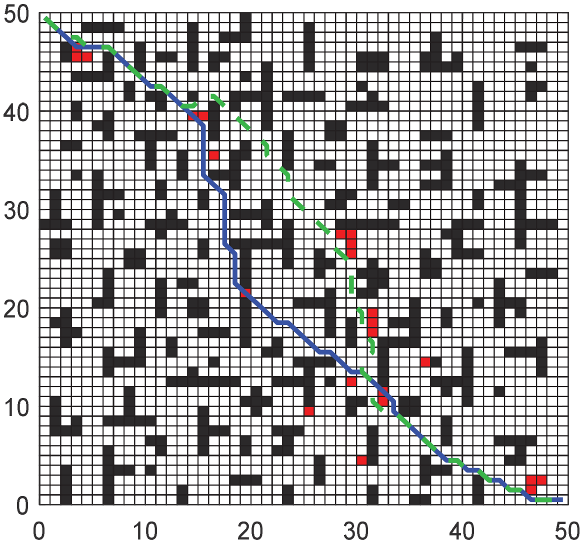


Figure 17 Comparison of global path planning and local path planning in a 50 × 50 grid environment.

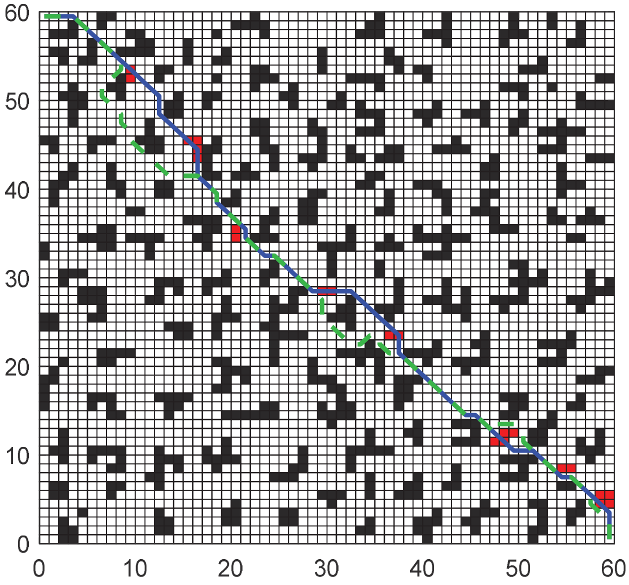


Figure 18 Comparison of global path planning and local path planning in a 60 × 60 grid environment.

6 Conclusion

An RLACA algorithm is proposed to solve the problems of premature and long calculation time in traditional path planning algorithms, increase the speed of AGV network calculation and improve the operational efficiency of logistics and transportation efficiency. The algorithm is summarized as follows:

1. By distributing the initial pheromone according to the distance from the start and end line and the target point, the blind search of the ant colony algorithm is reduced, and the probability of searching for the optimal path is maintained.
2. Change the transfer rules, increase the stability coefficient φ_{sta} and the target point trend coefficient φ_{ori} So that the path selection is smooth and moves to the target point at the same time so that the path planning of the robot is more in line with the actual situation, dividing the moving direction into the forward direction and the escape direction can significantly reduce retreat and self-locking, and improve the search efficiency.
3. Improve the pheromone update rules, and update the pheromone according to the order of the length of the path, which can strengthen the role of the optimal path while retaining the possibility of searching for other paths.
4. Combine the global taboo table with the local taboo table to prevent subsequent ants from repeatedly entering the escape direction, making the search more intelligent. The pheromone update of the redundant path avoids misleading the subsequent ants to find the optimal path.
5. By making redundant corrections to the path, the redundant path is corrected to avoid subsequent misleading of ants, affect the convergence speed of the path, and reduce the precocity of the ant colony algorithm.
6. The path of the improved ant colony algorithm is used as the learning data set of reinforcement learning, and the optimal action is selected according to the Q-table and the current environment.

Through simulation experiments, it can be concluded that the RLACA algorithm can improve the search efficiency of path planning, speed up the convergence speed, and solve the premature problem of the ant colony algorithm. Compared with other intelligent network calculation methods, this algorithm has stronger practicability and can complete dynamic path planning in a changing environment.

Acknowledgements

This work was partly supported by National Natural Science Foundation of China(NSFC, project number: 61973209) and Shanghai Municipal Information Development Special Fund-5G+ Intelligent Manufacturing Demonstration Line Innovation Laboratory Construction (Shanghai Municipal Commission of Economy and Informatization project number: 202001008.

References

[1] Y. Deng, Y. Chen, Y. Zhang, S. Mahadevan, 'Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment', Applied Soft Computing, pp. 1231–1237, Mar., 2012.

[2] X. Xu, X. Li, A. Zhan, 'An enhanced path planning method for unmanned surface vehicle based on JPS+ and goalbounding algorithm', IOP Conf. Ser.: Mater. Sci. Eng., p. 052079, Aug., 2019.

[3] A. K. Guruji, H. Agarwal, D. K. Parsediya, 'Time-efficient A* Algorithm for Robot Path Planning', Procedia Technology, pp. 144–149, 2016.

[4] B. F. Zhang, Y. C. Wang, X. L. Zhang, 'Mobile Robot Path Planning Based on Artificial Potential Field Method', AMM, pp. 350–353, Jul. 2014.

[5] M. Dorigo, L. M. Gambardella, 'Ant colonies for the travelling salesman problem', Biosystems, pp. 73–81, Jul., 1997.

[6] P. K. Das, P. K. Jena, 'Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators', Applied Soft Computing, p. 106312, Jul., 2020.

[7] H.-Y. Lee, H. Shin, J. Chae, 'Path Planning for Mobile Agents Using a Genetic Algorithm with a Direction Guided Factor', Electronics, p. 212, Sep., 2018.

[8] A. H. Karami, M. Hasanzadeh, 'An adaptive genetic algorithm for robot motion planning in 2D complex environments', Computers & Electrical Engineering, pp. 317–329, Apr., 2015.

[9] W. Zhang, X. Gong, G. Han, Y. Zhao, 'An Improved Ant Colony Algorithm for Path Planning in One Scenic Area With Many Spots', IEEE Access, pp. 13260–13269, 2017.

[10] Q. Song, Q. Zhao, S. Wang, Q. Liu, X. Chen, 'Dynamic Path Planning for Unmanned Vehicles Based on Fuzzy Logic and Improved Ant Colony Optimization', IEEE Access, pp. 62107–62115, 2020.

[11] P. Wang, S. Gao, L. Li, B. Sun, S. Cheng, 'Obstacle Avoidance Path Planning Design for Autonomous Driving Vehicles Based on an Improved Artificial Potential Field Algorithm', Energies, p. 2342, Jun., 2019.

[12] M. Dorigo, V. Maniezzo, A. Colorni, 'Ant System: An Autocatalytic Optimizing Process', p. 21, 1991.

[13] M. Dorigo, G. D. Caro, L. M. Gambardella, 'Ant Algorithms for Discrete Optimization', Artificial Life, pp. 137–172, Apr. 1999.

[14] M. Yousefikhoshbakht, F. Didehvar, F. Rahmati, 'An Efficient Solution for the VRP by Using a Hybrid Elite Ant System', INT J COMPUT COMMUN, p. 340, Apr., 2014.

[15] Z. Jiao, K. Ma, Y. Rong, P. Wang, H. Zhang, S. Wang, 'A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs', Journal of Computational Science, pp. 50–57, Mar., 2018.

[16] M. Dorigo, L. M. Gambardella, 'Ant colony system: a cooperative learning approach to the traveling salesman problem', IEEE Trans. Evol. Computat., pp. 53–66, Apr., 1997.

[17] A. C. Zecchin, H. R. Maier, A. R. Simpson, M. Leonard, J. B. Nixon, 'Ant Colony Optimization Applied to Water Distribution System Design: Comparative Study of Five Algorithms', J. Water Resour. Plann. Manage., vol. 133, pp. 87–92, Jan., 2007.

[18] Q. Luo, H. Wang, Y. Zheng, J. He, 'Research on path planning of mobile robot based on improved ant colony algorithm', Neural Comput & Applic, pp. 1555–1566, Mar., 2020.

[19] X. Wu, G. Wei, Y. Song, X. Huang, 'Improved ACO-based path planning with rollback and death strategies', Systems Science & Control Engineering, pp. 102–107, Jan., 2018.

[20] Y. Yao, Q. Ni, Q. Lv, K. Huang, 'A novel heterogeneous feature ant colony optimization and its application on robot path planning', in 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, 522–528, May, 2015.

[21] J. Liu, J. Yang, H. Liu, X. Tian, M. Gao, 'An improved ant colony algorithm for robot path planning', Soft Comput, pp. 5829–5839, Oct., 2017.

[22] J. Zhao, D. Cheng, C. Hao, 'An Improved Ant Colony Algorithm for Solving the Path Planning Problem of the Omnidirectional Mobile Vehicle', Mathematical Problems in Engineering, vol. pp. 1–10, 2016.

[23] G. Xia, Z. Han, B. Zhao, C. Liu, X. Wang, 'Global Path Planning for Unmanned Surface Vehicle Based on Improved Quantum Ant Colony

Algorithm', Mathematical Problems in Engineering, pp. 1–10, Apr., 2019.

[24] K. Akka, F. Khaber, 'Mobile robot path planning using an improved ant colony optimization', International Journal of Advanced Robotic Systems, p. 172988141877467, May, 2018.

[25] W. Zhaowei, H. Weihua, Y. Wenkai, Z. Zheng, 'Path Planning Based on Ant Colony Algorithm with Bayesian Decision-making', Modular Machine Tool & Automatic Manufacturing Technique, pp. 41–46, 2020.

[26] X. Dai, S. Long, Z. Zhang, D. Gong, 'Mobile Robot Path Planning Based on Ant Colony Algorithm With A* Heuristic Method', Front. Neurorobot., Apr., 2019.

[27] L. Hai, Y. Xiaoliu, X. Linghua, 'Robot Navigation Path Planning Under Honeycomb Grid Based on Dynamic Grouping Ant Colony Algorithm', Machinery Design & Manufacture, pp. 279–283,287, 2020.

[28] Y. Noguchi, T. Maki, 'Path Planning Method Based on Artificial Potential Field and Reinforcement Learning for Intervention AUVs', 2019 IEEE Underwater Technology (UT), pp. 1–6, 2019.

[29] F. Liu, C. Chen, Z. Li, Z.-H. Guan, and H. O. Wang, 'Research on path planning of robot based on deep reinforcement learning', 2020 39th Chinese Control Conference (CCC), 2020, pp. 3730–3734.

[30] C. Qu, W. Gai, M. Zhong, J. Zhang, 'A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning', Applied Soft Computing, vol. 89, p. 106099, Apr., 2020.

[31] F. Leiva, J. Ruiz-del-Solar, 'Robust RL-Based Map-Less Local Planning: Using 2D Point Clouds as Observations', IEEE Robot. Autom. Lett., pp. 5787–5794, Oct., 2020.

[32] F. Jian, F. MinRui, M. ShiWei, 'RL-ART2 Neural Network Based Mobile Robot Path Planning', Sixth International Conference on Intelligent Systems Design and Applications, pp. 581–585, 2006.

[33] J. Ming, W. Fei, G. Yuan, S. Longlong, 'Research on path planning of mobile robot based on improved ant colony algorithm,' Chinese Journal of Scientific Instrument, pp. 113–121, 2019.

Biographies

Wangwang Yu is a Master's Degree candidate at the School of Electrical Engineering, Shanghai Dianji University. He received his BS in Anhui Agricultural University, Hefei, in 2019. His research interest is robot control system.

Jun Liu is presently a Professor at the School of Electrical Engineering, Shanghai Dianji University, China. He received his Ph.D. in East China University of Science and Technology, Shanghai, in 2010. His current projects are in the areas of intelligent control and microcomputer control technology.

Jie Zhou He is currently pursuing the master degree in electrical engineering with China Shanghai Dian Ji University. His research interests include PMSM sensorless control and power electronics.