Research on global path planning of unmanned vehicles based on improved ant colony algorithm in the complex road environment

Xiaowei Li1,2,3, Qing Li1 and Junhui Zhang1

Abstract
When planning the path of a non-urbanized road, the default ant colony optimization (ACO) algorithm does not consider complex road state function such as uneven surface, road attachment coefficient, and vehicle turning angle limit. Based on the actual situation of roads and vehicles, a pavement state function that considers uneven areas such as road bumps and pavement attachment is proposed to improve the description of path length. Then, a heuristic function based on the $A^*$ algorithm and an improved mechanism for the initialization of pheromone distribution is proposed, which changes the blindness of ant colony search, accelerates the convergence of the ACO, and improves the search efficiency. The global search capability of the algorithm is enhanced by improving the path selection strategy and path transition probability function. The pheromone updating method is improved by using the MAX-MIN Ant System, which increases the algorithm diversity and avoids local optima. Further, using the pruning algorithm to reduce the number of paths significantly increases the convergence speed. Simulation results show that the improved ACO algorithm has better convergence speed and global search ability. Combining road state processing with vehicle corner processing can effectively improve the safety, adaptability, and reliability of autonomous vehicles. And the global optimal path planning of unmanned vehicles on complex roads can also be realized.

Keywords
Global path planning, unmanned vehicle, improved ACO algorithm, road surface state

Introduction
Path planning is an indispensable part of autonomous navigation and obstacle avoidance for unmanned vehicles, and has become a hotspot. Path planning refers to the devising of an optimal or suboptimal path without collision from start to destination in an environment with obstacles while satisfying all constraints. It is one of the key technologies for realizing vehicle intelligence.1 Path planning algorithms are grouped into global and local path planning.2 Global path planning is generally based on the known road environment to plan a path. The accuracy of path planning depends on the degree of environmental information acquisition, which can generate the best path and avoid local minima. Local path planning is a dynamic planning algorithm, which is usually used in overtaking, obstacle avoidance, and other scenarios. Path planning algorithms, according to the techniques used, can be also divided into the following four groups:

Search-based algorithms
These algorithms mainly discretize the state space into a graph in a deterministic way. Feasible paths are then searched using heuristic search to obtain optimal solutions. Maurovic et al.3 used the shortest path graph search algorithm based on $D^*$. However, it results in too many traversed nodes and complex calculation problems. Based on this algorithm, a heuristic function was added to find the shortest distance between a search node and a target node. The results indicated

1Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China
2University of Chinese Academy of Sciences, Beijing, China
3School of Rail Transportation, Shandong Jiaotong University, Jinan, China

Corresponding author:
Xiaowei Li, Institute of Microelectronics of the Chinese Academy of Sciences, 3 Beitucheng West Road, Chaoyang District, Beijing 100029, China.
Email: lixiaowei@sdjtu.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
that the A* algorithm has the advantage of searching for the optimal trajectory quickly and effectively. It is predominantly used for global path planning problems in low-dimensional space as it tends to be more complete in such problems.

**Sampling-based algorithms**

In contrast to search-based methods, sampling-based path planning does not need to model the environment completely. In addition, these algorithms have the characteristics of random sampling, fast search speed, and high planning efficiency. Typical algorithms include RRT and RRT*, which are probability complete algorithms. They have natural support for solving high-dimensional complex problems and are often used in global path planning. However, they have the problem of optimizing and searching many nodes in the space. Thus, they improve the path quality at the expense of computational efficiency. When dealing with high dimensions, high degrees of freedom and non-economic constraints, this performance tradeoff will lead to more difficulties.

**Trajectory planning method**

The trajectory planning method is necessary if we want to consider the motion equation during the planning process. It is often used in local path planning. The typical method is based on the optimal control method, which can obtain the time information corresponding to the path points. Its most important advantages are strict satisfaction of boundary conditions and uniform treatment of various kinds of constraints. Wang et al. proposed a trajectory planning method with synchronous planning and control, which considers various constraints and deals with external disturbances, improves the online computational efficiency, and realizes fast path planning. However, when there are many obstacles in the vehicle driving environment, such methods may be very time-consuming due to the need to solve complex optimization problems, and the inherent non-convexity make the problem hard to converge without proper initial guesses.

**Intelligent optimization algorithms**

With the development of intelligent control technology, various intelligent optimization algorithms have been used in path planning. These algorithms have the characteristics of self-learning, self-determination, and self-adaptive search, obtained by simulating the biological behavior of groups. They include genetic algorithms, the particle swarm optimization algorithm, firefly algorithm, artificial fish swarm algorithm, artificial bee colony algorithm, pigeon-inspired optimization algorithm, and ACO algorithm, and their combinations. Among them, the ACO algorithm has strong robustness, good global optimization ability, and inherent parallelism. Moreover, it can be easily combined with various heuristic algorithms to improve the performance of the algorithm. Therefore, it is widely used in global path planning. However, in some application scenarios with practical computational time constraints, it cannot avoid the possibility of falling into local optima.

This paper is primarily focused on global path planning based on the improved ACO algorithm. The ACO algorithm was proposed by Dorigo et al. It is a distributed bionic algorithm with strong robustness. However, the default ACO has the problems of slow convergence and easy to fall into local optima. To this end, many methods are proposed to improve the performance of the default ACO algorithm. For example, Zhu et al. combined the A* and ACO algorithms to obtain an improved ACO algorithm for robot path planning. The heuristic function is adaptively adjusted, the evaluation function of the heuristic A* algorithm is used for reference in the later process of the path, and the direction information is introduced into the heuristic function of the Ant Colony System (ACS) algorithm to improve its search efficiency. Zheng et al. used pseudo-random state transition rules to select paths, calculate the state transition probability according to the current optimal solution and the number of iterations, and adjust the deterministic or random selection proportion. The optimal and worst solutions were introduced to improve the global pheromone update method and a dynamic penalty method to solve the deadlock problem.

The ACO algorithm applied to unmanned vehicle path planning is a simple shortest path planning algorithm. However, considering road conditions and the complex surrounding environment, such as the presence of pits, bulges, and other road irregularities, and the uncertainty of the road adhesion coefficient, how to be more efficient, safe, and reliable with the minimum energy consumption, obtaining the optimal global path planning from point to point is still a subject worth studying.

This paper proposes an improved ACO algorithm to achieve the above objectives. The main research contributions of this study are as follows: (1) Most previous studies only consider the shortest path, but in this paper, we propose a road state function and redefine the path length, so that we can sense complex road conditions, such as road leveling and ponding, through cameras and lidars sensors installed on unmanned vehicles, and better reflect the actual road conditions of vehicle movement. (2) A heuristic function combining the A* algorithm and the road state function and an initial pheromone allocation mechanism considering the turning angle of the vehicle is proposed, and the path pruning algorithm is used to avoid the “local detour” search path, changed the blindness of the ant colony algorithm in the complex road environment, accelerated the convergence speed of the algorithm, and improved the search efficiency; (3) Improved the path
selection strategy and the path transition probability function, and based on the MAX-MIN Ant System, proposed an optimal path pheromone incremental selection strategy improves the pheromone update method, increases the diversity of the algorithm, avoids entering the local optimal solution, and improves the global search ability of the algorithm in complex road environments.

The remainder of this paper is organized as follows. Section 2 presents the problem description and environment modeling. Section 3 describes the proposed improved ACO algorithm for global path planning. Section 4 presents the simulation. Section 5 concludes the paper.

**Problem description and model settings**

**Problem description**

The global path planned by an unmanned vehicle is not a simple shortest path. Careful consideration is given to the vehicle’s front wheel turning angle, road surface unevenness, other road smoothness conditions, and the road surface adhesion state. Cameras and other sensors are used to detect the bumpy area of the road surface in the global path and the static obstacles on the ground, such as vehicles and railings. The subsequent path is then chosen based on the road state.

**Model building**

To ensure the best driving state for unmanned vehicles in road environments, the grid method is used to divide the environment according to the road information conditions, and static obstacles on the ground are identified, and otherwise impassable areas are noted. As shown in Figure 1, it can be represented by a twodimensional grid map, and the impassable regions such as static obstacles on the ground can be represented by “0”; otherwise, they are represented by “1.” Zero and one’s matrix is used to describe the unmanned driving static obstacle grid environment information for vehicle operation. Let the grid model be M rows by M columns, in the order from left to right and top to bottom. The first grid coordinate in the upper left corner is then (1,1), START represents the starting point, GOAL represents the endpoint, the serial number is n, and the grid coordinates are (x, y). Then,

$$x = \text{int}\left(\frac{n}{M}\right) + 1$$

$$y = \text{mod}(n - 1, M) + 1$$

In this paper, the unmanned vehicle is regarded as a mass point; considering that the vehicle is a front-wheel-drive system, the front wheel is the driving wheel, and the front wheel angle \( w \) is the steering wheel angle, where \( w \in [0, w_{\text{max}}] \) (e.g. \( w_{\text{max}} = 45^\circ \)). When \( w_{\text{max}} \) is between 0° and 22.5°, the vehicle can only go straight. When \( w_{\text{max}} \) is between 22.5° and 67.5°, the vehicle can go straight, or drive to the front left or right. When \( w_{\text{max}} \) is between 67.5° and 90°, the vehicle can go straight, drive to the front left or right, or turn to the left or right. To achieve more accurate vehicle path planning, as shown in Figure 2, after the unmanned vehicle has taken the steering angle into consideration, the ants perform path finding. When starting from a node to the next node, the path direction selection factor associated with the steering angle should be considered.

**Definition of road surface condition function**

In a real environment, the motion of the vehicle depends on the ground, so the terrain greatly affects the motion of the car. In this study, we map the road state according to the sensor perception into the grid map, and describe each grid according to the road leveling and attachment conditions. As shown in Figure 1, each number in the grid represents the road state.

**Figure 1.** Grid Map Model: 1, free area; 0, obstacle area.

**Figure 2.** Vehicle steering angle range.
The road surface state (φ, θ) of each node is numerically described in the grid graph, and the ants judge the optimal road surface state node through the numerical description of the road surface state of the current node and surrounding nodes, as shown in Figure 3.

**Definition of path length**

When the unmanned vehicle performs path planning from the starting point to the ending point, it needs to select a path according to the road smoothness and road attachment state of the adjacent nodes around each node, to ensure that a globally optimal path can be obtained and vehicle driving safety, stability, and reliability can be realized. To map the road surface state to the grid map, it is necessary to convert the flat state such as unevenness and road attachment into the length of the path in the grid map. Consequently, for any node \( j \), the converted path length \( L_{ij} \) from node \( i \) to \( j \) is as shown in equation (3):

\[
L_{ij} = \begin{cases} 
L_{ij} & \text{if } e_i = e_j \\
\frac{L_{ij}}{c_1} & \text{if } e_i < e_j \\
\frac{L_{ij}}{c_2} & \text{if } e_i > e_j
\end{cases}
\]

where \( c_1 \) and \( c_2 \) are constants greater than one and \( c_1 < c_2 \). \( e_i \) and \( e_j \) are the road surface states of node \( i \) and node \( j \), respectively.

**Improved ACO algorithm**

**ACO algorithm description**

The ACO algorithm, which was proposed by Dorigo et al. in the early 1990s, is a new type of evolutionary algorithm. With the introduction of a positive feedback parallel mechanism, the algorithm has the advantages of strong robustness, excellent distributed computing mechanism, and easy combination with other methods.

The path transfer probability is as follows:

\[
p_{ij}^k(t) = \begin{cases} 
\frac{[\tau_k(t)]^\alpha [\eta_k(t)]^\beta}{\sum_{k' \neq j} [\tau_k(t)]^\alpha [\eta_k(t)]^\beta}, & \text{if } j \in J_k(i) \\
0, & \text{else}
\end{cases}
\]

where \( \tau_k(t) \) is the transition probability of ant \( k \) from node \( i \) to the next node \( j \); \( \alpha \) is the relative importance of the pheromone, \( \beta \) is the relative importance of the heuristic factor; \( J_k(i) \) is the set of nodes that the ants are allowed to visit in the next step. \( \eta_k(t) \) is the pheromone concentration left by the ants from node \( i \) to node \( j \) at time \( t \); \( s \) is a node in \( J_k(i) \); \( \eta_k(t) \) is the heuristic function of the ants from the current node to the next node at time \( t \), and the mathematical expression is shown in equation (5):

\[
\eta_{ij} = \frac{1}{d_{ij}}
\]

Throughout the cycle, ants will continue to release pheromones, while the previous ants will continue to disappear. We can set a parameter \( \rho \) (0 < \( \rho \) < 1) to indicate the degree of volatility of pheromones. After \( n \) moments, the ant needs to update the pheromones of each path in real time, as shown in equation (6):

\[
\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t)
\]

\[
\Delta \tau_{ij}(t) = \sum_{k=1}^{m} \Delta \tau_{ij}^k(t)
\]

In the equation, \( \Delta \tau_{ij}^k(t+1) \) is the concentration of pheromones released by ant \( k \) on \((i, j)\) at time \((t, t + 1)\). The \( \Delta \tau_{ij}(t+1) \) presented in equation (7) shows the
Algorithm improvement

The default ACO algorithm basically considers only the ideal conditions of the road, which results in poor practical application effect. This paper considers the actual road environment of unmanned vehicles, constructs a new mathematical model, and improves the ACO algorithm to achieve a rapid optimization effect.

Improvement of initial pheromone distribution. When the ACO algorithm searches for the first time, the pheromone uniform distribution method is used. The pheromone on each path is initialized to a constant C, which causes the algorithm to converge slowly in the early search stage and increases the search time of the algorithm. To optimize the initial pheromone, this paper proposes a method to optimize the initial pheromone according to the angle between the Euclidean distance between nodes i and j and the Euclidean distance between the starting point s and the ending point g, as shown in equation (8). The method speeds up the initial convergence speed of the algorithm, reduces the number of iterations, and improves the real-time performance of the algorithm.

\[
\tau_{ij}(0) = \omega_{ij}(0) \tag{8}
\]

\[
\omega_{ij}(0) = \frac{w_{\text{max}}}{\omega} \tag{9}
\]

\[
\omega = \tan^{-1} \left( \frac{k_{sg} - k_{ij}}{1 + k_{ij}k_{sg}} \right) \tag{10}
\]

In equation (9), \(w_{\text{max}}\) is the maximum steering angle of the unmanned vehicle, \(k_{sg}\) is the slope between the starting point \(s\) and the ending point \(g\), and \(k_{ij}\) is the slope between nodes \(i\) and \(j\). According to the ratio of \(w_{\text{max}}\) and the included angle \(\omega\), the distance between the next node and the Euclidean distance between the starting and ending points is judged. The larger the ratio is, the trend of the ants selecting the node is the shortest straight-line distance from the target endpoint.

g. The higher the initial pheromone concentration, the smaller the ratio \(k_{ij}\). This means that the path node selected by the ant deviates from the direction of the target endpoint. The farther the distance from the shortest straight line, the lower the initial pheromone concentration, as shown in Figure 4.

**Improved heuristic function.** The transition probability of the default ACO algorithm depends on pheromone and local heuristic functions, both of which are determined by the path length. However, on the actual road, the selection of the optimal road by the unmanned vehicle also needs to consider factors such as the road surface state on the path. This paper proposes an adaptive heuristic function that integrates the \(A^*\) algorithm to improve the convergence speed of the ACO algorithm.

The \(A^*\) algorithm is a typical heuristic search algorithm that was proposed by Hart in 1968. The algorithm combines the advantages of Dijkstra’s algorithm and the best-first search algorithm. Adding a heuristic function to the Dijkstra algorithm to determine the search direction solves the problems of large path search range and long response time, and has high search efficiency. The cost function expression of the \(A^*\) algorithm is as follows

\[
f(n) = g(n) + h(n) \tag{11}
\]

In equation (11), \(f(n)\) is the evaluation function of the node \(n\), \(g(n)\) is the actual cost from the next node to the starting point, which is the distance between the next node and the starting point; \(h(n)\) is the estimated cost between the next node and the target node, and the value is the distance between the next node and the target node. In the ACO algorithm, the estimation function of the \(A^*\) algorithm is used as the heuristic information to search for the global optimal path. In this paper, it is assumed that the state value of the road surface for unmanned vehicles is known, which is the inspiration for using the ACO algorithm. The road state function \(\epsilon_{\text{r}}\) is added to the equation value to improve the convergence speed and select a smooth optimal path for the road. The improved heuristic information equation is as follows:

\[
\eta_{ij} = \frac{\epsilon_i}{h(j) + g(j)} \tag{12}
\]

where \(\epsilon_i\) is the road leveling state value of node \(j\).

**Improvement of path transition probability \(p_{ij}^t(t)\).** Considering that in the actual environment, the turning angle of an unmanned vehicle is \(w < w_{\text{max}}\), the road surface condition also affects the safety and reliability of the vehicle. Turning points should be reduced in path planning, and areas with poor road surface conditions should be avoided. This paper introduces the steering heuristic and the road state function when calculating the path.
transition probability. The node transition probability formula is as follows:

$$P_{ij}(t) = \begin{cases} \frac{[\tau(t)]^\alpha[\eta(t)]^\beta[\omega(t)]^\delta}{\sum_{j \neq i} [\tau(t)]^\alpha[\eta(t)]^\beta[\omega(t)]^\delta}, & j \in S_i(t) \\ 0, & \text{otherwise} \end{cases}$$ (13)

where $\alpha$ and $\beta$ represent the influence factors of pheromone and heuristic, respectively, $\omega(t)$ is the steering heuristic function of node $i$ to node $j$, and $\delta$ is the steering heuristic factor:

$$\delta = \begin{cases} 1, & \omega \leq 45 \\ 0, & \omega > 45 \end{cases}$$ (14)

In equation (14), $\omega$ is the steering angle of the unmanned vehicle, and the smaller the angle, the larger the $\delta$.

In equation (13), $\epsilon(t)$ is the road surface state heuristic function. This paper assumes that this value is obtained by the vehicle perception system, the function is positive, and the road surface state heuristic factor $\theta = ab$, which can be adapted according to different road conditions. Further, different road surface state heuristic factors $\theta$ can be chosen.

**Improvements in path selection strategy.** In the initial stage of the ACO algorithm, the pheromone concentration is set by differentiation. At the beginning of the path search, the ants will choose a path with a high probability of passing. However, the algorithm will fall into a local optimum state when the pheromone on the path gradually increases. Therefore, the path state selection strategy needs to be adjusted as follows:

$$j = \begin{cases} \arg \max \{ [\tau(t)]^\alpha[\eta(t)]^\beta[\omega(t)]^\delta[\epsilon(t)]^\theta \}, & q \leq q_0 (\text{mode 1}) \\ \text{roulette}(P_{ij}^t), & q > q_0 (\text{mode 2}) \end{cases}$$ (15)

In equation (15), $q$ is a random variable uniformly distributed in the range $[0,1]$: $q_0$ is the threshold uniformly selected by pseudo-random probability, in the range of $(0,1)$; when $q \leq q_0$, the next node $j$ is determined by mode 1, which combines the maximum value of the pheromone concentration and each heuristic function. When $q > q_0$, node $j$ is determined by mode 2, the roulette mode. In conclusion, if the value of $q_0$ is larger, the path is more likely to adopt mode 1, which speeds up the convergence but reduces the global search ability. Conversely, if the value of $q_0$ is small, the path transition will select mode 2 with high probability, which increases the randomness of path selection and improves the global search ability. Therefore, the setting of the $q_0$ value has an important influence on the convergence speed and global search ability. This paper proposes a method to determine the value of $q_0$ according to the number of iterations $N$ and the adjustment coefficient.

$$q_0 = \frac{N_{\text{max}} - N}{N_{\text{max}}} \times \mu$$ (16)

In equation (16), $N_{\text{max}}$ is the set maximum number of iterations, $N$ is the current number of iterations, and $\mu$ determines the adaptive adjustment coefficient according to the current path selection situation and the current number of iterations.

$$\mu = \begin{cases} 2 \left( N < N_{\text{max}} \right) \& L_{\text{best}} \leq L_{\text{st}} \\ 1 \left( N < N_{\text{max}} \right) \& L_{\text{best}} > L_{\text{st}} \end{cases}$$ (17)

In equation (17), $L_{\text{st}}$ is the Euclidean distance between the starting point and the path’s endpoint, which is the lower limit of the ACO. Once the shortest path is less than $L_{st}$ and the current iteration number is less than the maximum iteration number through the comparison of paths, it means the local optimum is being entered. However, redetermining $q_0$ through the $\mu$ value selection facilitates escape from the local optimum solution.

**Improved pheromone trajectory update rules.** The selection of path nodes determines the convergence speed of ant colony search, and pheromones affect the possibility of path selection. All ants complete one cycle. At this time, the pheromone on each path the ants pass needs to be adjusted, which is an important factor affecting whether each ant can find the optimal path. Based on the possible changes in vehicle information in the driving process of unmanned vehicles, not only the global path pheromone update is needed, but also the local path pheromone needs to be updated. In this paper, we propose an improved pheromone update rule for the MAX-MIN Ant System (MMAS) based on the changes in vehicle driving information. The improved pheromone trajectory update rule is as follows.

1. Setting of upper and lower pheromone limits

Set the upper and lower limits of the pheromone; the upper limit is $\tau_{\text{max}}$, and the lower limit is $\tau_{\text{min}}$. The upper and lower limits of the pheromone are as follows:

$$\tau = \begin{cases} \tau_{\text{max}}, & \text{if } \tau \geq \tau_{\text{max}} \\ \tau_{\text{min}}, & \text{if } \tau < \tau_{\text{min}} \end{cases}$$ (18)

In equation (18), $\tau_{\text{max}} = Q/(1 - \rho) \times L_{gb}$ ($\rho$ is the pheromone volatilization rate, $L_{gb}$ is the global optimal path length, $Q$ is a constant greater than one), $\tau_{\text{min}} = \tau_{\text{max}} \sqrt{P_{\text{best}}}$, where $P_{\text{best}}$ is the probability of constructing the optimal solution. This setting can ensure that the algorithm has good diversity and avoids various local optima. However, the difference between the path pheromone is too large to ensure the diversity of the algorithm.
Global pheromone update

The unmanned vehicle path-planning process based on the ACO algorithm is mainly divided into algorithm optimality seeking and algorithm stagnation phases. In the algorithm optimality seeking phase, it is desired to achieve fast convergence and maintain stochasticity; in the algorithm stagnation phase, it is expected to jump out of the local optimal solution, enhance the diversity of the algorithm, and strengthen the global search capability.

\[ \tau_{ij}(t + 1) = \frac{1}{C_0 r(t)} + \Delta \tau_{ij}^{best} \]

In equation (19), \( \Delta \tau_{ij}^{best} \) is the pheromone increment on the optimal path, which is specifically defined as follows:

\[ \Delta \tau_{ij}^{best} = \begin{cases} \frac{1}{L_{gb}}, & N > N_0 \\ \frac{1}{L_{ib}}, & N < N_0 \end{cases} \]

where \( N \) is the current number of iterations and \( N_0 \) is the selection threshold for the number of iterations. \( L_{gb} \) and \( L_{ib} \) represent the global optimal path length and the contemporary optimal path, respectively. \( L_{gb} \) will change with the number of iterations. When only \( L_{gb} \) is selected, the algorithm will enter the local optimal solution too quickly, limiting the search for a better solution. Using a dynamic mixing strategy. In this paper, an optimal path pheromone incremental selection strategy is proposed. In the proposed strategy, the selection threshold of the number of iterations is set to \( N_0 \), and the contemporary optimal solution is used for updating when \( N < N_0 \) to ensure the diversity of the algorithm in the early stage. When \( N > N_0 \), the global optimal solution is used. At this time, the convergence speed can also be considered.

Redistribution of path pheromones

When the algorithm enters the stagnation phase, the pheromone is maximum on the optimal path. At this time, to avoid entering the local optimal solution, the amount of pheromone needs to be adjusted to increase the algorithm diversity selection. This paper combines the concept of pheromone proportional update and proposes the allocation of pheromone increments for non-optimal paths based on the ratio of the current number of iterations to the maximum number of iterations to be determined adaptively:

\[ \tau_{ij} = \tau_{ij} + \frac{N_{max} - N_{st}}{N_{max}} (\tau_{max} - \tau_{ij}) \]

In equation (21), \( N_{st} \) is the number of iterations in the algorithm stagnation stage, and \( N_{max} \) is the maximum number of iterations. When the algorithm stagnation stage is detected, the diversity of the algorithm is improved by increasing the pheromone increment on the non-optimal path, and the earlier the entry is in the stagnation stage of the algorithm, the larger the increment of pheromone redistribution by equation (21) is; it is beneficial to jump out of the local optimal solution faster. When the number of iterations tends to the end of the algorithm, the optimal path pheromone amount tends to remain unchanged.

Path pruning. After each ant completes a path search, the output path will have a phenomenon of “local detour.” Using the pruning algorithm to prune the path can greatly increase the convergence speed. As shown in Figure 5, the algorithm steps are as follows:

1. Initialize the new path as empty; let \( i \) be the starting node of the ant’s current path and add \( i \) to the new path.
2. Let \( j \) be the target node of the current path and backtrack forward, if \( j \) is the neighbor node of \( i \), then let \( i = j \), and add \( i \) to the new path.
3. Determine whether \( i \) is the target node, if not, jump to Step 2; if it is, the algorithm ends and the new path is output as the clipped path.

The proposed algorithm. In summary, the specific path-planning steps for unmanned vehicles based on the improved ACO algorithm are as follows:
Step 1. Use the grid map to model the work environment and determine the start, end point, and static obstacles.

Step 2. Initialize the ant colony system. Set the number of ants $m$, number of iterations, and road state function. Then, determine the relatively important parameter $\alpha$ of the pheromone, parameter $\beta$, the global pheromone volatility coefficient $\rho$, and other related parameters.

Step 3. Update the tabu table. Put ant $k (k = 1, 2, \ldots, m)$ on the current node and add the current node to the corresponding tabu table.

Step 4. Select the next mesh. Then, calculate the heuristic function according to equation (12), determine the value of $q_0$ according to the number of iterations $N$ and the adjustment coefficient, and select the next feasible grid according to the set equation (15) with a deterministic or random probability function. If the ant reaches the target grid, it will go to Step 5, otherwise it will go to Step 3.

Step 5. If the ant reaches the target node, perform path clipping, replace the current path with the pruned path, and go to Step 6.

Step 6. Continue to repeat Step 3 until each ant finishes searching for the target during its iteration, then go to Step 7.

Step 7. Update the pheromone. After each iteration, if the number of iterations satisfies the inequality $N \leq N_{\text{max}}$, update the path pheromone and determine if it satisfies the convergence condition. The algorithm will exit if the convergence conditions are met. If not, go to Step 3. If the number of iterations satisfies the inequality $N > N_{\text{max}}$, no further counts will be made. As long as the end condition is met, the final result is output.

Simulation experiments

To verify the effectiveness of the improved algorithm and compare it with the default ACO algorithm, we conducted a simulation comparison on the MATLAB 2020a simulation platform. First, a grid map was set up. The map was divided into $20 \times 20$ grids, and each unit was one. The starting point was $(2,2)$, and the ending point was $(19,19)$. According to the known road surface state, the grid map was obtained as follows:

![Figure 6. Description of experimental road surface state.](image)

In the Figure 6, “1” represents the drivable area (road smoothness and adhesion coefficient are both “1”); “0” represents the obstacle area; between “0” and “1” represents the road surface condition coefficient (first row: smoothness, second row: adhesion coefficient).
The parameter settings of the algorithm are shown in Table 1.

| Parameter | Value |
|-----------|-------|
| Number of ants | 20 |
| Number of iterations | 30 |
| $\alpha$ | 1 |
| $\beta$ | 5 |
| $w_{\text{max}}$ | $22.5^\circ$–$67.5^\circ$ |
| $a$ | 0.5 |
| $b$ | 0.5 |
| $\rho$ | 0.1 |
| $Q$ | 100 |
| $N_0$ | 5 |

The results, convergence curve, and time consumption of each algorithm are shown in Figure 8. Compared with the algorithm proposed in this paper, the default algorithms cannot bypass regions with poor surface state. The path length of the original ACO algorithm is much larger than that of the improved algorithm, and the difference between the converged path and the optimal path is large. After adding kinematic constraints, the search path will not make large-angle turns, so the path length will be greatly reduced. After adding the path clipping mechanism, the algorithm easily converges to a near-optimal solution. Adding all the above mechanisms to the algorithm in this paper, it could converge quickly and the path length was the shortest.

In terms of time consumption, the improved algorithm greatly shortens the calculation time. After adding kinematic constraints, the algorithm will not search the rear area, which improves the search efficiency. After path prune, the algorithm will ignore the detour path and disperse pheromones on the critical path, which makes the convergence speed faster. After adding the above mechanisms, the search time is shortened to about 23% of the default algorithm.

![Figure 7. Results of path pruning algorithm ($\omega_{\text{max}} = 90^\circ$) (Green track: unpruned; blue path: pruned).](image-url)
Comparison with default A* algorithm and ACO

As shown in Figure 9, when the pavement state is set at (11, 12) and (12, 12), the default A* algorithm and ACO do not consider the pavement state, and the planned path passes through the area with poor pavement state and cannot be bypassed. Even if the A* algorithm adds a pavement state factor, a path with a turning angle that is too large will be planned, such as a turning angle of 90° at (12, 11) and (10, 12).

Compared with the improved ACO algorithm proposed in this paper, after the default A* algorithm considers the road surface state and adds kinematic constraints, although the planned path turn angle is less than 90°, its path length is greater than the path obtained by the proposed algorithm, and it is not the optimal path. It can be seen that the algorithm proposed in this paper can plan a more reasonable path under complex road conditions.
Analysis of influence of pavement state parameters on results

In this paper, a pavement state description method is proposed and added to the ACO algorithm. When the vehicle is running on the actual road, the final planned path can be controlled by adjusting the road surface leveling state factor $a$ and the attachment state factor $b$. As shown in Figure 10, when the values of $a$ and $b$ are large, the ACO algorithm will ignore the road surface state and identify it as a drivable area. When the values of $a$ and $b$ are small, the ACO algorithm will try to avoid passing the road areas with poor surface condition. Therefore, by adjusting the values of the $a$ and $b$ factors, the passing of the vehicle on the road surface with different levels of smoothness and different surface adhesion states can be controlled.

Path search diversity

By improving the global pheromone update mechanism, this study ensures that the algorithm maintains diversity during path search. As shown in Figure 11, when other parameters are fixed, only by setting different random seeds, the output paths with large differences are finally obtained.

To assess diversity, define path diversity, “diversity” as follows:

$$
\text{diversity} = \frac{\sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{n=1}^{L} \text{dist} \left( \text{path}_i^n, \text{path}_j^n \right)}{N^2}
$$

where $N$ is the number of paths, $\text{path}_i^n$ is the $n$th point of path $i$, $\text{path}_j^n$ is the $n$th point of path $j$, and $L$ is the path with the shorter length when considering the lengths of paths $i$ and $j$. Through experimental observation, we determined that when the value of path diversity is greater than five, the diversity of the algorithm is good. Six experiments were performed with different maps; the results are shown in Table 2.

Pheromone changes during algorithm iteration

To observe the pheromone changes in the iteration process of the ACO algorithm, we plotted the results of the 1st, 5th, 10th, and 30th iterations, as shown in Figures 12 and 13. Figure 13 shows the results of the no-path pheromone redistribution mechanism. The number in each square represents the pheromone content, and it is displayed as “0” when the pheromone content is less than 0.01. It can be seen that as the number of
Figure 10. Influence of different values of $a$ and $b$ on the path.

Figure 11. Path diversity (Using different random seeds, the final output path of the ACO algorithm is obtained.).
Table 2. Statistics of the experimental results.

| Experiment number | Convergence time (s) | Optimal value   | Diversity |
|-------------------|----------------------|-----------------|-----------|
| 1                 | 12                   | 29.7279         | 7.54      |
| 2                 | 10                   | 30.3137         | 6.08      |
| 3                 | 17                   | 30.8995         | 6.57      |
| 4                 | 9                    | 29.1421         | 5.89      |
| 5                 | 14                   | 29.7279         | 6.42      |
| 6                 | 11                   | 30.3848         | 5.44      |

Figure 12. Results without pheromone redistribution mechanism. (The numbers indicate the pheromone content, and it is displayed as “0” when the pheromone content is less than 0.01.): (a) 1st iteration result, (b) 5th iteration result, (c) 10th iteration result, and (d) 30th iteration result.
iterations increases, the pheromone finally converges to a path. Figure 13 shows the simulation results obtained when the path pheromone redistribution mechanism is added. It can be seen that the initial path and the final converged path show great differences, and the algorithm has the ability to jump out of the local optimal solution.

**Conclusion**

An improved ant colony algorithm is proposed to aim at the road conditions of unmanned vehicles in a non-urban road environment. Under complex road conditions, a new type of road state function is defined, and a variety of ACO improvement strategies are proposed,
which not only accelerates the convergence speed but also avoids the algorithm from entering local optimum. Simulation results show that the improved ant colony algorithm can effectively solve unmanned vehicles’ global path optimization problem under complex road conditions. This paper only considers the influence of general road conditions on global path planning. However, when the number of obstacles increases or the terrain obstacles become more complex, the complexity of the path planning algorithm increases significantly. Therefore, the next step will be to reduce the algorithm’s complexity further and speed up the search.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Science and Technology Department of Jiangsu Province (2020Z411).

ORCID iD
Xiaowei Li https://orcid.org/0000-0002-6548-9666

References
1. Gonzalez D, Perez J, Milanes V, et al. A review of motion planning techniques for automated vehicles. *IEEE T Intell Transp* 2016; 17: 1135–1145.
2. Kala R and Warwick K. Motion planning of autonomous vehicles in a non-autonomous vehicle environment without speed lanes. *Eng Appl Artif Intell* 2013; 26: 1588–1601.
3. Maurovic I, Seder M, Lenac K, et al. Path planning for active SLAM based on the D* algorithm with negative edge weights. *IEEE Trans Syst Man Cybern* 2018; 48: 1321–1331.
4. Song R, Liu Y and Bucknall R. Smoothed A* algorithm for practical unmanned surface vehicle path planning. *Appl Ocean Res* 2019; 83: 9–20.
5. Noreen I, Khan A and Habib Z. A comparison of RRT, rrt* and RRT*-smart path planning algorithms. *Int Comput Sci Net* 2016; 16: 20–27.
6. Wang XW, Peng HJ, Liu J, et al. Optimal control based coordinated taxiing path planning and tracking for multiple carrier aircraft on flight deck. *Def Technol* 2022; 18: 238–248.
7. Wang X, Liu J, Su X, et al. A review on carrier aircraft dispatch path planning and control on deck. *Chin J Aeronaut* 2020; 33: 3039–3057.
8. Wang X, Liu J, Peng H, et al. A simultaneous planning and control method integrating APF and MPC to solve autonomous navigation for USVs in unknown environments. *J Intell Robot Syst* 2022; 105: 1–16.
9. Chen X and Gao P. Path planning and control of soccer robot based on genetic algorithm. *J Ambient Intell Humaniz Comput* 2020; 11: 6177–6186.
10. Wu Y, Li Z, Sun C, et al. Measurement and control of system resilience recovery by path planning based on improved genetic algorithm. *Meas Control* 2021; 54: 1157–1173.
11. Das PK and Jena PK. Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Appl Soft Comput* 2020; 92: 106312.
12. Patle BK, Pandey A, Jagadeesh A, et al. Path planning in uncertain environment by using firefly algorithm. *Def Technol* 2018; 14: 691–701.
13. Zhang Y, Guan G and Pu X. The robot path planning based on improved artificial fish swarm algorithm. *Math Probl Eng* 2016; 2016: 3297585.
14. Goel P and Singh D. An improved ABC algorithm for optimal path planning. *Int J Sci Res* 2013; 2: 261–264.
15. Wang BH, Wang DB and Ali ZA. A Cauchy mutant pigeon-inspired optimization-based multi-unnanned aerial vehicle path planning method. *Meas Control* 2020; 53: 83–92.
16. Miao C, Chen G, Yan C, et al. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. *Comput Ind Eng* 2021; 156: 107230.
17. Wang L. Path planning for unmanned wheeled robot based on improved ant colony optimization. *Meas Control* 2020; 53: 1014–1021.
18. Yu X, Gao X, Wang L, et al. Cooperative Multi-UAV task assignment in Cross-Regional Joint Operations Considering Ammunition Inventory. *Drones* 2022; 6: 77.
19. Dai X, Long S, Zhang Z, et al. Mobile robot path planning based on ant colony algorithm with A* heuristic method. *Front Neurorobot* 2019; 13: 15.
20. Dorigo M, Maniezzo V and Colomi A. Ant system: optimization by a colony of cooperating agents. *IEEE Syst Man Cy B* 1996; 26: 29–41.
21. Zhu Y, You XM, Liu S, et al. Research for robot path based on improved ant colony system (ACS) algorithm. *CEA* 2018; 54: 129–134.
22. Zheng Y, Luo Q, Wang H, et al. Path planning of mobile robot based on adaptive ant colony algorithm. *J Intell Fuzzy Syst* 2020; 39: 5329–5338.
23. Ding Y, Ma H and Li S. Path Planning of Omnidirectional mobile vehicle based on road condition. In: 2019 IEEE international conference on mechatronics and automation (ICMA). Tianjin, China, 4–7 August, 2019, pp.1425–1429. New York, NY: IEEE.
24. Wang X, Liu X, Wang Y, et al. Research on path planning of mobile robot based on improved A* Algorithm. In: *IE&EM 2019* (eds, CF Chien, E Qi and R Dou) Singapore, 25 April 2020, pp.153–161. Springer.
25. Yuan Z, Yu H and Huang M. Improved Ant Colony optimization algorithm for intelligent vehicle path planning. In: 2017 International conference on industrial informatics-computing technology, intelligent technology, industrial information integration (ICICH), Wuhan, China, 2–3 December 2017, pp.1–4. New York, NY: IEEE.
26. Stützle T and Hoos HH. MAX–MIN Ant System. *Future Gener Comput Syst* 2000; 16: 889–914.
27. Zhang Q and Ma J. Improved Ant Colony algorithm based path planning for mobile robot. *J Northeast Univ* 2013; 34: 1521.
28. Stützle T and Hoos H. Improvements on the ant-system: introducing the max-min ant system. In: *Artificial neural nets and genetic algorithms*. Vienna: Springer, 1998, pp.245–249.