

Smooth Path Planning of Ackerman Chassis Robot based on Improved ant Colony Algorithm

Guannan Lei, Yili Zheng

School of Technology, Beijing Forestry University

Received: May 4, 2020. Revised: June 8, 2020. Accepted: July 3, 2020. Published: July 20, 2020.

Abstract— In the domain of robotics and autonomous driving, the automatic path planning of vehicle collision-free motion is an essential task on the navigation level. It is found that the traditional path planning algorithm and the ployline path cannot fully meet the driving requirements of Ackerman chassis robot. In order to solve the autonomous navigation problem of Ackerman chassis mobile robot in structured environment, this paper presents a new improved algorithm. The method of configuration space can introduce the robot's own structural size parameters into the algorithm. Through convex polygon detection method, the local U-shaped area in the map is transformed into a closed area. The essence of these two strategies is to preprocess the map. The initial pheromone distribution is no longer globally uniform, but is distributed according to the terrain. The volatilization factor of pheromone is changed from static constant to dynamic one, which is combined with Poisson distribution law. This strategy makes the improved pheromone distribution law not only avoid the randomness and blindness in the initial stage of the algorithm, but also ensure the ant colony's exploration behavior and guiding role in the middle stage of the algorithm. Path smoothing is also a challenging task. This algorithm optimizes the path step by step by improving the evaluation function, removing redundant nodes and 2-turning algorithm. Thus, a collision free smooth path suitable for Ackerman robot is obtained. This paper combines a variety of algorithm improvement strategies, not only improving the performance of ant colony algorithm path exploration, but also planning a smooth curve path suitable rather than polyline for Ackerman mobile robot tracking. The algorithm is coded and simulated by MATLAB, and the feasibility and effectiveness of the algorithm are verified. This will provide an important basis for the subsequent algorithm migration and lay the foundation for the path tracking control of the Ackermann chassis robot.

Keywords—Ackermann chassis mobile robot; Smooth path planning; Dynamic pheromone volatilization factor; Improved ant colony algorithm

I. INTRODUCTION

AUTONOMOUS robots and vehicles have been used to perform various missions, such as, warehouse logistics, pilotless driving, surveillance/reconnaissance, exploration of unknown area, and operation in hazardous environments. Path planning requires robots to use path planning algorithm to realize autonomous navigation of unmanned vehicles according to their own perception of the environment and their own positioning information. Autonomous navigation technology of driverless vehicle has become the focus of research institutions

because of its broad application prospects.

In the development of driverless vehicle path planning algorithms, scientists have put forward various algorithms, such as, Dijkstra algorithm [1], [2], A* algorithm [3], rapidly exploring random tree algorithm (RRT) [4], artificial potential field method, particle swarm optimization (PSO) [5], ant colony optimization(ACO) [6], [7] et al. In recent years, scientists have improved the path planning algorithms in different degrees to meet the needs of the actual situation in their respective fields.

The ant colony optimization algorithm is an effective way to solve the problem of unmanned vehicle path planning [8]–[10]. On the basis of traditional ant colony algorithm, Hui yang, and Jie Qi, Xiamen University, presented an efficient double-layer ant colony optimization algorithm, called DL-ACO, for autonomous robot navigation. This DL-ACO consisted of two ant colony algorithms that run independently and successively. A parallel elite ant colony optimization method was proposed to generate an initial collision-free path in a complex map, and then, they applied a path improvement algorithm called turning point optimization algorithm, in which the initial path was optimized in terms of length, smoothness, and safety [11], [12]. The authors think that, in the real world, as time goes by, the pheromone on the path will gradually evaporate. And its volatilization is helpful for ants to explore other areas to find better path. Therefore, they improved the pheromone volatilization strategy and simulated the change of pheromone in time. On the other hand, the weighted penalty function was introduced. They divided the path into different levels of advantages and disadvantages, and adjusted the penalty function according to different levels of paths, so as to encourage ants to explore the worst path. Thus, the exploration ability of ant colony was improved.

Juntao Cheng and Zhihuai Miao [13], State Key Laboratory of Robotics and Systems, Harbin, proposed an improved ant colony algorithm. They improved the positional distribution of the initial population and increased the adaptive evaporation factor and simulated annealing. It was found through experiments that the algorithm can effectively reduce the problem of search time [14].

Zhang Miao, National University of Defense Technology, proposed an improved ant colony algorithm in 3D environment. They rastered the map using bilinear interpolation method and translate the 3D terrain into 2D terrain according to the given slope threshold. And then they combined OpenMP parallel

programming technology to accelerate this algorithm by mining the concurrency of ant colony algorithm using the idea of parallel computing [15], [16].

Ankita Khurana [17], Amity University Uttar Pradesh, has been committed to the ant colony algorithm path planning. Because every ant drops an amount of simulated pheromone at each node in which the ant has just covered up. The pheromone distribution law in the improved algorithm was modified, so that the ant colony can explore the path line including all the specific network nodes in the map [18].

Liang Kai [19], Kunming University of science and technology, improved the updating method of pheromone of ant colony by using the wolf colony assignment principle to avoid the ant colony algorithm falling into the local optimum and accelerate the convergence speed of the algorithm. And, it proposed a smoothing method of center node replacement. The purpose is to remove the sharp corner in the path and improve the smoothness of the path. However, this algorithm is suitable for indoor small mobile robot. It can probably not be transplant directly to mobile robot with Ackerman chassis.

Marco Dorigo and Thomas Stützle [20], University Libre de Bruxelles (ULB), took advantage of ant colony optimization (ACO) algorithms in preserving high diversity, and intended to extend ACO algorithms to deal with multimodal optimization. Combined with current niching methods, an adaptive multimodal continuous ACO algorithm was introduced. An adaptive parameter adjustment was developed, which takes the difference among niches into consideration. They used the algorithm to solve problems with a large number of local optimal solutions.

Michalis Mavrovouniotis, and Yang Shengxiang [21], De Montfort University, put forward an improved ant algorithm and used this algorithm in dynamic path planning. The algorithm was to increase the diversity via transferring knowledge from previous environments to the pheromone trails using the immigrant schemes. In their research, an ACO framework for dynamic environments was proposed. In the algorithm, different immigrant schemes, including random immigrants, elitism-based immigrants, and memory-based immigrants, were integrated into ACO algorithms for solving path planning problem.

These algorithms can solve their own problems in specific fields. However, these algorithms cannot be directly transplanted to the Ackerman chassis robot navigation problem, which needs further improvement and optimization.

In the research, it found that, for the large-scale Ackerman chassis robot, the traditional path planning algorithm has certain limitations. During narrow roads or corners, the Ackerman chassis unmanned vehicle frequently suffers from scraping of the body and obstacles under the guidance of conventional algorithms. On the one hand, it is due to the large size of the Ackerman chassis robot vehicle, which cannot be ignored relative to the path and obstacles; On the other hand, during the turn, the Ackerman chassis vehicle itself has Difference of Radius Between Inner Wheels (Difference of Radius Between Inner Wheels refers to the difference between the turning radius of the front inner wheel and the turning radius

of the rear inner wheel when the vehicle turns. Due to the Difference of Radius Between Inner Wheels, when the vehicle turns, the motion trajectories of the front and rear wheels do not coincide. If only paying attention to the front wheels being able to pass and forget the Difference of Radius Between Inner Wheels while driving, it may cause the rear inner wheels to drive off the road or collide with other objects.). In order to meet the motion requirements of Ackerman chassis unmanned vehicle, this paper not only improves the search performance of the ant colony algorithm to a certain extent, but also designs the algorithm specifically for the unique structural characteristics and motion laws of Ackerman mobile robots. Through MATLAB simulation, the feasibility of the algorithm is verified. This algorithm can specifically provide a reasonable smooth path for Ackerman chassis robot path planning.

II. ALGORITHM IMPROVEMENT STRATEGY

A. Characteristics Analysis of Ackerman Chassis Vehicle Model

In order to solve the autonomous navigation and path planning problems of Ackerman chassis robots, its structure need to be analyzed firstly. When the absolute size of the mobile robot is small enough, or when the size of mobile robot is much smaller than that of obstacles and road space, the mobile robot can be abstracted as a particle. Neglecting its dimensions can simplify the problem greatly. And the algorithm can use memory resources as much as possible to find the most efficient path. However, when the overall dimension of the mobile robot cannot be ignored, in actual operation, the planned road under the traditional algorithm may cause the car body and obstacles to collide. This could scratch or damage the mobile robot. In this case, the external dimension of the mobile robot needs to be introduced into the algorithm as an important parameter.

In the Fig. 1, it shows the parameters of the Ackermann chassis and the parameters of the relationship between the vehicle and the road during the turn.

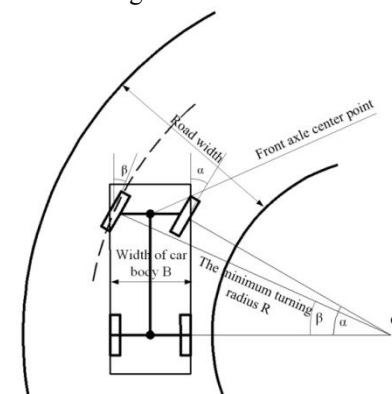


Fig. 1 Modeling of Ackerman chassis vehicle and extraction of key parameters in turning process

According to the vehicle construction modeling, R is the minimum turning radius of the vehicle. α and β is the yaw angle of the two front wheels of the vehicle. B is the distance of wheels. These parameters play an important role when the

Ackerman chassis vehicle moves or turns.

B. The Principle of Obstacle Expansion Strategy

Grid map is a common method to build map environment in robot path planning [22], [23]. The map of ant colony algorithm is constructed by grid map. In this method, the map is segmented by a square grid of equal size and the matrix model is established [24]–[26]. In order to consider the dimensions of the Ackerman chassis vehicle, based on grid map, this paper adopts the method of configuration space. The essence of configuration space is the reconstruction of the map. However, the method used in this paper is not to construct a reconstruction obstacle through the collision detection method of a disk robot, but to directly expand the obstacle with specific size. Specially, for Ackerman chassis robot, it is necessary to expand 1/2 of the vehicle body width on the basis of obstacles. Because if the grid cell size is relatively large and the area of the obstacle expanding outward is less than one grid cell, then the grid cell can only be determined as the expansion area. This may result in areas that could otherwise pass through not being able to pass through. If the grid cell is too small, the computation will increase geometrically. This requires that the map construction and grid division must be reasonable. Considering both of the map construction and configuration space, the map is divided with 1/2 car body as the grid cell.

Then, this makes it possible to consider the impact of body width on path planning by configuration space. As shown in the Fig. 2 below is the new map generated by the above strategy.

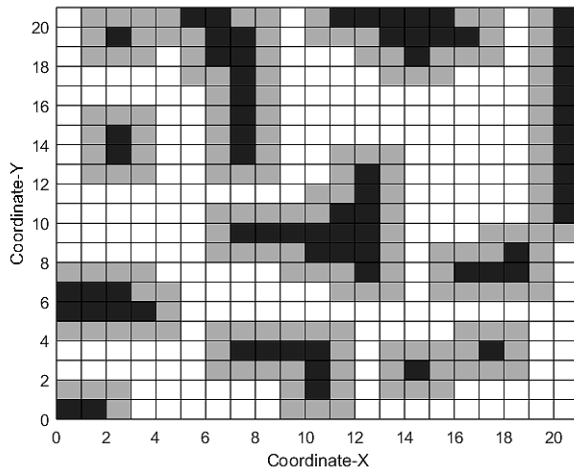


Fig. 2 Schematic diagram of configuration space

This strategy needs to detect the grid cells one by one. If the detected grid cell color is black, it means that it is an obstacle area, then, the accessible area of the eight grid areas adjacent to the grid cell will be set as the obstacle extended area. The corresponding position of the obstacle extended area in the map matrix model is expressed as 0.5. The extended obstacle area in the figure is represented by gray. In this way, the inaccessible area includes two parts, obstacle area and extended obstacle area, which are represented by black grid cells and gray grid cells respectively.

Through this strategy, although the vehicle is still regarded as

a particle in the algorithm, the vehicle width factor in path planning can be considered in the algorithm. As shown in Fig. 2, reasonably build maps for Ackerman robots through meshing and construction space.

C. Local Map Reconstruction Based on the Convex Polygon Detection

Compared with the configuration space, similarly, the significance of the convex polygon detection algorithm lies in the map reconstruction in the early stage of the algorithm [27], [28]. As shown in the Fig. 3, the obstacles connected by the red polyline form different shapes and sizes of U-shaped semi closed areas. The experimental results show that there is a certain degree of blindness in ant's exploration of path in the early stage during the algorithm processing. A large number of ants would fall into these U-shaped areas, which may lead to the failure of the ants' search. This will greatly reduce the efficiency of the algorithm.

Therefore, a convex polygon detection algorithm is proposed. When the algorithm detects that obstacles form a U-shaped area, it determines whether the target is in the U-shaped area. As long as the end point is not in the U-shaped area, it will connect the head and the tail of the red polyline with green lines to form a closed area. When the grid has not less than half of the area in the enclosed area, it is marked purple. The corresponding position in matrix model is changed to 1. This algorithm makes the U-shaped region have a rejection effect on ants. To some extent, it solves the problem that ants find their way into the dead end because of the U-shaped area.

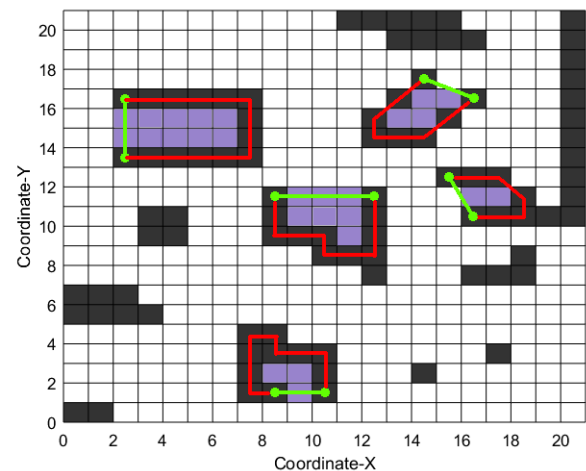


Fig. 3 Concave obstacle detection and local map reconstruction

The above two strategies can be regarded as the map preprocessing before the algorithm starts. In essence, it realizes the purposeful modification and reconstruction of the existing map.

D. Improvement of Initial Pheromone Distribution

Initial pheromone distribution is the distribution rule of pheromones in the map at the beginning of map construction. In traditional ant colony algorithm, the distribution of initial pheromone is fixed and equal everywhere in the whole map [29]–[31]. This leads to the blindness and randomness of the

traditional ant colony algorithm in the early stage of path search, and it leads to the low efficiency of ant search in the early stage of the algorithm. Since pheromones play a guiding role in ant path search, this paper improves the distribution of initial pheromones.

As shown in the Fig. 4, firstly, it should connect the starting point and the target point of the ant colony.

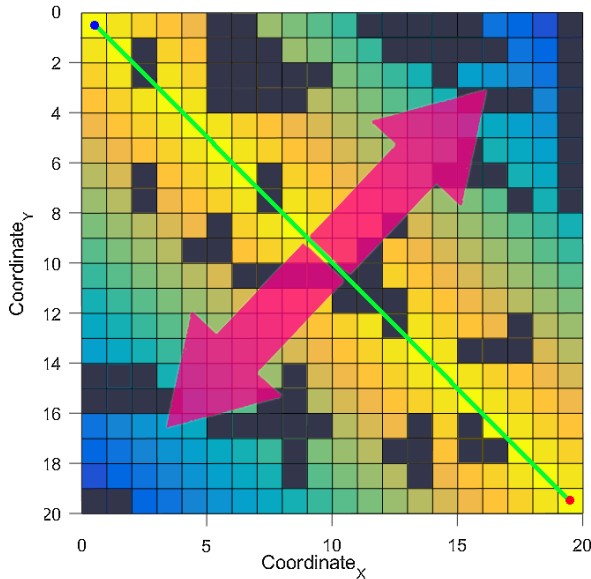


Fig. 4 Initial pheromone distribution gradient map

The straight line L is the shortest path that can be achieved theoretically when there is no obstacle in the map. Then, based on the line, the direction perpendicular to the line is the direction of pheromone gradient. So, the closer the grid cell on the map is to the line, the higher the initial pheromone concentration is, and the farther away from the line, the lower the initial pheromone concentration is. In this way, it can make the ants search near the line L as much as possible in the early stage of path search, instead of searching the far path. The simulation result shows that, the rational distribution element of the initial pheromone can play a better role in guiding ants. Of course, the reason for this design is that the function of large-scale Ackerman chassis unmanned vehicle is generally used for cargo transportation in large warehouse, and its map structure is relatively simple, rather than complex maze structure map. Thus, to some extent, it can avoid the blindness and randomness of the traditional ant colony algorithm in the early stage of path search.

E. Design of Volatilization Factor of Dynamic Pheromone

Pheromones left in the path of ants will volatilize with time. Volatilization factor is used to measure the volatilization rate of residual pheromones on the pathway. The pheromone content on the path consists of two parts [32], [33]: one part includes the rest of the pheromone left by the ants before, the other part is the pheromone secreted by the current ant in the path. Therefore, for the volatilization of pheromone, an index, pheromone volatilization factor, is introduced to measure the volatilization rate and residue of pheromone.

In traditional ant colony algorithm, the pheromone volatility factor is usually a fixed constant [34], [35]. In this paper, the dynamic pheromone volatilization factor (D-Rho) is used in the algorithm. In order to find out the change rule of dynamic pheromone volatilization factor, by exploration, we decided to make the dynamic pheromone volatilization factor satisfy Poisson distribution law in the whole process. D-Rho is improved according to the following equation:

$$\rho(x = k) = A \frac{\lambda^k}{k!} e^{-\lambda} + B, \quad k = 0, 1, 2 \dots \quad (1)$$

The core of the (1) is the Poisson distribution. On this basis, (1) is reconstructed according to the role of the pheromone in the algorithm, so that it can meet the needs of the algorithm. In the equation (1), ρ is the value of pheromone volatilization factor. x obeys the Poisson distribution with parameter $\lambda (\lambda > 0)$. And, according to the experiment, the value of λ is obtained. In this paper, λ takes the value 10. k equals the number of ant colony iterations N . A is the proportional coefficient. B ($0 < B < 1$) is the offset.

Experiments show that when the volatilization factor is large, it is conducive to the convergence of the algorithm, and when the volatilization factor is small, it is conducive to the behavior of ant colony exploration. The Poisson distribution variant constructed by the above (1) can reasonably utilize this characteristic of the algorithm. As shown in the Fig. 5 below, the horizontal axis is the number of the iteration, and the vertical axis is the pheromone volatilization factor. The yellow curve is the curve of Poisson distribution after upward translation. Red curve is (1) corresponding curve. The dynamic distribution of pheromone volatilization factors is shown in the Fig. 5 ($A = -5$). In the early and late stages of the algorithm, the pheromone volatilization factor remained at a relatively high level, while in the middle of the algorithm progressing, the curve exhibited a trough state. The pheromone volatilization factor has a significant tendency to decrease first and then rise.

The reason for this design is that the pheromone volatilization factor of traditional ant colony algorithm is constant. If the pheromone volatility factor is set too large, the ants may lose their ability to explore paths and the algorithm will be premature. However, if the value of pheromone volatilization factor is set too small, it may increase the blindness of the algorithm and make the algorithm not convergent. Pheromone is the substance left by ants in the path. It is of great significance to guide ant path exploration behavior. However, the pheromone remaining on the path will evaporate over time. Pheromone volatilization factor is an index used to measure the speed of pheromone volatilization. When the pheromone volatilization factor is too large, it means that the pheromone on the path volatilizes faster. After a period of time, there are fewer pheromones remaining on the path. The guidance effect of pheromone on ants will be weakened, and the ability of ants to randomly explore paths will be improved. When the pheromone volatilization factor is too small, it means that the pheromone volatilization on the path is relatively slow. After a period of time, there are more pheromones remaining on the path, and the pheromone's guiding effect on the ant will

become stronger. The ant is more likely to first explore the path of the previous ant under the pheromone's guidance, and the algorithm converges quickly.

In the algorithm proposed in this paper, the dynamic pheromone volatilization factor will automatically adjust the value of the pheromone volatility factor during the algorithm. In the algorithm proposed in this paper, the D-Rho will automatically adjust the value of the pheromone volatility factor during the algorithm. The value of D-Rho in the early stage of the algorithm is relatively high. At the same time, under the influence of the initial pheromone, the blindness of the algorithm is reduced to a certain extent, and the guidance ability of the algorithm to ants is enhanced. In the middle of the algorithm, reducing the value of D-Rho appropriately can improve the ability of ants to explore new paths to some extent. The value of D-Rho of the algorithm rises up in the later period, which is beneficial to the convergence of the algorithm. This strategy can also be reflected to a certain extent in subsequent simulations.

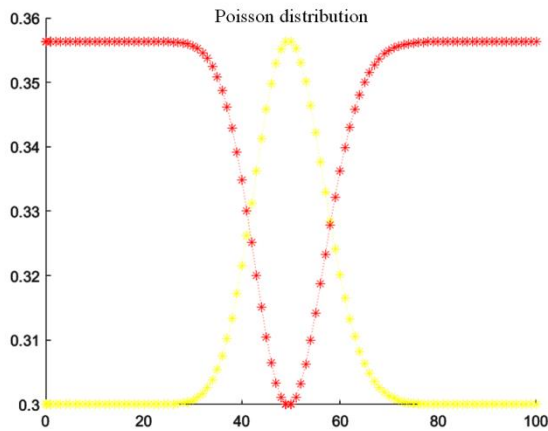


Fig. 5 Variation of dynamic pheromone volatilization factors in based on Poisson distribution

F. The Next Node Selection Scheme Improvement

In traditional ant colony algorithm, roulette is the method that each ant colony chooses the next node [36], [37]. The probability of selecting each node is determined by the pheromone content of each node. The equation is as follows (2):

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in \{N - tabu_k\}} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}, & j \in \{N - tabu_k\} \\ 0, & Others \end{cases} \quad (2)$$

In the equation (2), p_{ij} is the probability of selecting the node. $\tau_{ij}(t)$ is the pheromone concentration of the node. η_{ij} is a heuristic information factor. α and β is the weight coefficient.

In this way, the algorithm will directly select a node with a high pheromone concentration as the next node with greater probability. So, it brings a problem. Assume that the node with the highest pheromone concentration is away from the end point of the target point among all the nodes that the ant may select next. However, according to the roulette algorithm, the probability of selecting this point is the largest. It increases the

possibility of ants "going back". A number of redundant nodes may appear in the planned path, thereby increasing the length of the path.

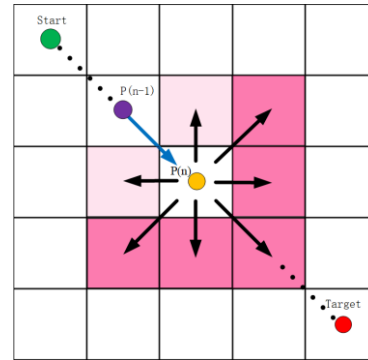


Fig. 6 Classification method of child nodes to be selected

In response to this problem, this paper makes improvements to the choice of the next node. As shown in the Fig. 6, P_n is the current node, $P_{(n-1)}$ is the parent node of P_n . The red grid cells area indicated by the black arrow is the area where the next node $P_{(n+1)}$ may be selected. Here, all possible next nodes (the seven red grid cells in the Fig. 6) are divided into two categories: The first type (Make a vector $\vec{\alpha}$ from P_n to the target, the angle between the vector indicated by the black arrow and $\vec{\alpha}$ is not more than 90 degree. That is, the dark red area in the Fig. 6 is the node pointing to the target point. The second type (the angle between the vectors indicated by the black arrow and $\vec{\alpha}$ is more than 90 degree. That is, the Light red area in the Fig. 6 is the node that faces away from the target point. The first type of nodes has a higher priority than the second type of nodes. The next node will be selected firstly in the nodes of the first type, the method is still roulette method. The second type of nodes will be selected only when the nodes in the direction of the target point in the taboo table have been selected. The improved algorithm can effectively prevent ants from taking unnecessary paths and reduce redundant nodes.

G. Path Evaluation Function Reconstruction

The traditional ant colony algorithm evaluation function is directly measured by the length of the path [38]–[40]. However, after a lot of experiments, it finds that the path smoothness of the ant colony algorithm is poor, and there are many inflection points and redundant nodes [41], [42]. Because of the existence of these redundant nodes, when the mobile robot tracks the established trajectory, it will constantly adjust its direction and posture as the trajectory changes. Even in some nodes the robot need to do the action of in-situ rotation. But the Ackerman chassis robot cannot rotate in place due to the particularity of its chassis structure. It has a minimum turning radius limit. In addition, the frequent occurrence of local zigzag paths in the route not only increases the path length, but more importantly brings great challenges to the vehicle's motion control. Such a path is not realistic for it. And this greatly increases the energy consumption and time cost of the mobile robot during actual motion. So, the cost function (3) is reconstructed in this article.

$$F(n) = A \times Pkm(n) + B * T(n) \quad (3)$$

In the equation: $F(n)$ is the evaluation of the entire path cost, which is used to comprehensively measure the path cost. Path cost is measured by two metrics. The first indicator is P_{km} , which represents the cost of travel for each ant that successfully reaches the target point. For each step of ants in horizontal or vertical direction, P_{km} increases by one unit length, and for each step in oblique direction, P_{km} increases by 1.4 unit length. The first measure is established by the proportionality between path length and cost. This is also the only indicator of traditional ant colony algorithm to measure the path cost. The second indicator is $T(n)$, which is the cost of turning in the path. On the one hand, considering the particularity of the structure of the Ackerman chassis vehicle, it has a minimum turning radius and cannot make in-situ steering. A path with sharp angles does not fully satisfy the actual situation of this type of vehicle. On the other hand, consider the further smoothing of the paths involved in the chapters III of this paper. This increases the path cost, control cost and time cost to a certain extent. Therefore, it is necessary to take this index into the entire path planning cost. For each inflection point in the path, $T(n)$ is increased by one. A and B are the weights of the two indexes respectively. This paper uses these two indicators to comprehensively measure the cost of the path, so as to achieve a reasonable and comprehensive evaluation of the path. Taking the inflection point of the path into the evaluation function can reduce unnecessary redundant points and corner points in the path, and to a certain extent obtain the effect of preliminary smooth path.

III. PATH SMOOTHING METHOD

A. Deletion of Redundant Nodes on the Path

The first step of the path Smoothing Algorithms is redundant node removal. The strategy first judges each node of the initially determined path. The node P_n is selected as the current node, $P_{(n-1)}$ is the parent node of P_n and $P_{(n+1)}$ is the child node of P_n . Then, it will determine whether the parent node of the current node is in a straight line with the child node. If the three points are not in a straight line, it means that the node P_n is an inflection point. Then further determine whether there is an obstacle between the parent node and the child node of P_n . The specific operating principle is shown in the following Fig. 7. The black polyline is the current path, and P_n is the path node. Since the coordinates of each point are known, connect the parent node of P_n to the child node and make a straight line L . Therefore, the analytical equation of the straight line L can be obtained from their coordinates:

$$k = \frac{(y_{n+1} - y_{n-1})}{(x_{n+1} - x_{n-1})} \quad (4)$$

$$y = k(x - x_{n-1}) + y_{n-1} \quad (5)$$

If line P_1P_3 does not intersect the obstacle, P_2 is considered as redundant node and can be removed from the path list. It can get the second figure. Then, calculate each node in the path in turn to get the new path. The purpose of this method is to remove redundant nodes from the path list, and at the same

time, it can achieve preliminary smoothing of the path.

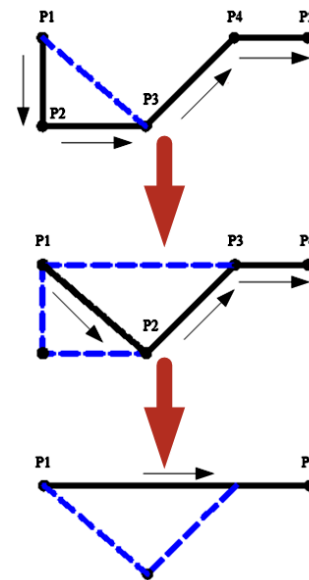


Fig. 7 Redundant node removal and local new path generation

B. Turning Algorithm for Ackerman Vehicles

Based on the redundant node removal strategy to obtain a smoother path, this part will use the 2-turning algorithm to completely turn the polyline path into a smooth curve, which can be applied to the track route of Ackerman robot tracking.

At present, the path smoothing methods mainly include: polynomial least squares fitting, curve fitting (such as Bezier curve, B-spline curve) and the like. When smoothing the collision avoidance path of the mobile robot in the Ackerman chassis, the following factors must be considered: the minimum turning diameter of the vehicle (as shown in the Fig. 1), the deviation between the smoothed path and the initial path (the deviation should be as small as possible), and the smoothed path cannot intersect the obstacle areas.

In theory, the minimum turning radius of a vehicle in an Ackermann chassis is only related to its own mechanical structure. In this paper, the turning radius of the two headings is selected as the minimum turning radius R of the robot. On the one hand, the reason is to ensure the turning efficiency. The larger the turning radius, the more time and distance it takes. Moreover, the larger the turning radius, the larger the maximum turning offset. On the other hand, when the radius of the turn is too large, it is possible to cause the robot to collide with the obstacle.

The main idea of the 2-turning method is to use two circumscribed circles to smooth two adjacent ployline paths. As shown in the Fig. 8 below, the path, L_1 and L_2 , form a corner at the junction. The direction of the blue arrow is the direction of the vehicle driving. O_1 and O_2 are the centers of two circumscribed circles of line L_1 and line L_2 respectively. S_1 and S_3 are the tangent points of two circumscribed circles and the original path (L_1 and L_2) respectively. S_2 is the tangent point of two circumscribed circles. The two red arcs in the Fig. 8 are the turning path.

The turning process is divided into two steps, the first

heading change is called collision avoidance turning path, and the second heading change is called resuming path. α and β are yaw angle and return angle respectively. The 2-turning method uses the turning law of the Ackerman chassis of the mobile robot for smoothing. In this way, the polyline path is completely changed into a smooth path.

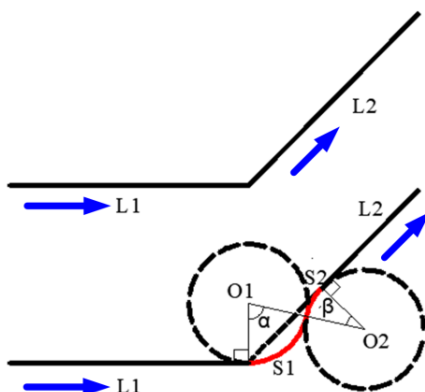


Fig. 8 The mathematical principle of the 2-turning path smoothing algorithm

IV. DISCUSSION

A. Algorithm Performance Analysis and Path Smoothness Comparison

Based on the above strategy of remove redundant nodes, Fig. 9 shows the processing effect of this strategy. In the Fig. 9, the purple point is the starting point, the red point is the target point, and the green point is the node in the path list. The ployline is the path planned by the two algorithms. Comparing the two figures, Fig. 9(a) contains a large number of intermediate nodes. Fig. 9(b) shows the new path after removing redundant nodes.

Three different typical redundant nodes are represented in the three red circles. The first type of redundant node is P_2 in the Fig. 9(a) It is generated by the inherent attributes of the map used by the ant colony algorithm. Because the map is rasterized, and each grid cell represents a node. It means that, in traditional ant colony algorithm, the path planned must move from node P_1 to node P_3 through intermediate node P_2 . However, it is clear that there is no barrier between P_1 and P_3 , and P_2 is a redundant node that can be removed. Therefore, as shown in Fig. 9(b), during the path post processing, P_2 is deleted from the path node list.

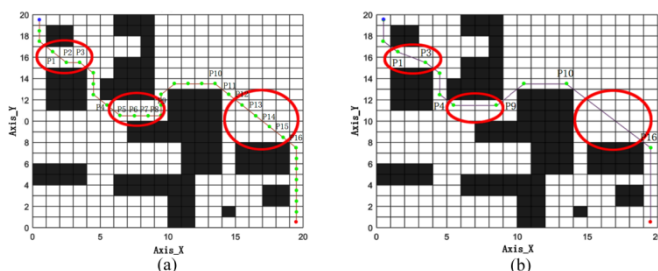


Fig. 9 Schematic diagram of three types of redundant nodes and their processing methods.

As shown in the second red circle, it is the second type of redundant node. The reason for the second type of redundant node is caused by the traditional algorithm itself. Especially, as the map grows, the grid cells in the map increase too. It will inevitably produce such redundant nodes. Obviously, there are no obstacles from P_4 to P_9 . Therefore, P_5 - P_8 is deleted by the redundant node strategy, and a passable path is directly formed between P_4 and P_9 . In fact, it can be seen that a small (5 nodes) partial path re-planning is done here.

The last red circle represents the third type of redundant node. The reason for this kind of node is also based on the inherent properties of the raster map. The difference between the first two redundant nodes is that the five nodes of P_{11} - P_{15} are located on the shortest path between P_{10} and P_{16} . However, in order to simplify the nodes in the path list as much as possible, P_{11} - P_{15} is still considered as redundant nodes removed from the path list. The aim is to simplify the output and feedback of unnecessary commands between the unmanned vehicle control system and the executive components.

As shown in the Fig. 9 above, the implementation of the redundant node removal strategy reduces the number of nodes in the path node list from 33 to 12. Not only the smoothness of the path is improved, but also the redundant commands of the control system and the unnecessary actions of the chassis are reduced. In addition, it can simplify the system's control process for unmanned vehicles and reduces unnecessary vehicle attitude adjustments. For example, P_2 between P_1 and P_3 is a redundant node. After removing P_2 , the vehicle can maintain a fixed course and directly drive from P_1 to P_3 . Otherwise, the vehicle needs to drive from P_1 to P_2 first, and then drive from P_2 to P_3 by readjusting the heading angle. This not only increases the control cost, but also increases the path length and time cost. As the size of the map increases, redundant nodes inevitably appear in the path. When there are many redundant nodes in the path, this strategy will further demonstrate its advantages.

B. Comprehensive Analysis of Algorithm Performance

In order to further verify whether the proposed algorithm can meet Ackerman chassis vehicle trajectory tracking requirements. This paper adds a vehicle kinematics model based on the vehicle model established in Chapter II. According to the relationship between the grid and the vehicle size, the wheelbase of the vehicle is set to 2m, and the width of the vehicle is set to 1m. In the simulation process, the vehicle travels at a constant speed of 0.3 m/s at a low speed. Because there is a linear relationship between the steering wheel angle and the front wheel angle of the vehicle, in the experiment we simplified the model and directly controlled the front wheel angle through the algorithm. In addition, the maximum angle of the front wheels of general small Ackerman chassis vehicles is 30-40°, so we sets the maximum turning angle to 40°. To simplify the model, in the simulation process, we only consider the longitudinal control of the vehicle and ignore the lateral motion control and lateral disturbance.

In this section, the performance of the traditional ant colony algorithm and the algorithm provided in the paper was

compared. The language of the MATLAB is used for simulation. The simulations are implemented on PC that has a Intel Duo Core i5-4210M CPU @ 2.60GHz and 4GB of RAM under Windows 7. The control parameters of the algorithm in this paper are set the following as: The number of the generation are $N = 150$, the number of the ants are $M = 500$ in each generation, $\alpha = 1$, $\beta = 0.3$. The reasonable values of N and M have an important influence on the running result of the algorithm. In order to ensure the optimal running result of the algorithm, the values of N and M must not be too small. If the values of N and M are too small, the algorithm will be premature. If the values of N and M are too large, it will increase the amount of calculation and greatly reduce the efficiency of the algorithm, so it is necessary to make a balance between the two. In addition, if M is too small, it may cause the current generation of ants to fail in pathfinding, and the resulting path is not optimal. Therefore, it is necessary to ensure that there are a sufficient number of ants in each generation. After a lot of experiments and attempts, it is found that $N=(150-160)$ and $M=(500-600)$ can meet the algorithm requirements. α is the pheromone importance factor, and β is the heuristic function importance factor. The two factors both affect the ant's path exploration behavior. As the value of α increases, the result converges faster, but if α is set too large, it is easy to weaken the random searchability and easily fall into the local optimum. As the value of β increases, the ants have more exploration ability, but when the β value is too large, the results may not converge. Synthesizing the influence of various factors on the results, set $\alpha=1$, $\beta=0.3$.

As shown in the Fig. 10 below, the map of the grid-based environment is 21×21 . Smooth red curve is the result of path planning based on the algorithm of this paper. In order to make the running result clearer, the red circle in the Fig. 10 shows a partial enlarged view of the turning path indicated by the arrow. Each curve is a smooth curve.

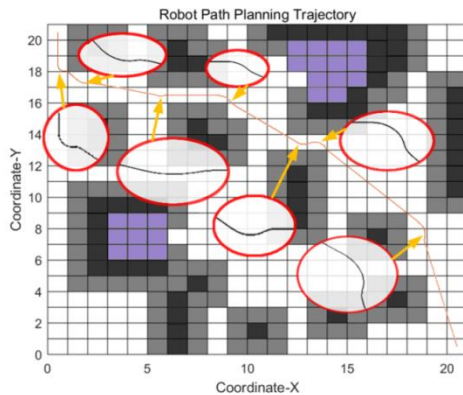


Fig. 10 Simulation results based on MATLAB

Under the map environment of Fig. 10, as shown in the figure below, it is the algorithm convergence curve based on the traditional ant colony algorithm and the algorithm of this paper.

The abscissa in the Fig. 11 is the number of ant colony iterations $N = 150$. The left ordinate indicates the shortest path length of each generation of ants. The right ordinate is the value of the pheromone volatilization factor during the operation of

the algorithm. The purple ployline in the Fig. 11(a), is the convergence curve by the traditional ant colony algorithm. The blue ployline is the convergence curve based on the improved ant colony algorithm. The yellow curve and the blue curve correspond to the least squares fit curves of the two polylines, respectively. The red scatter indicates the value of the pheromone volatilization factor for each generation in the algorithm.

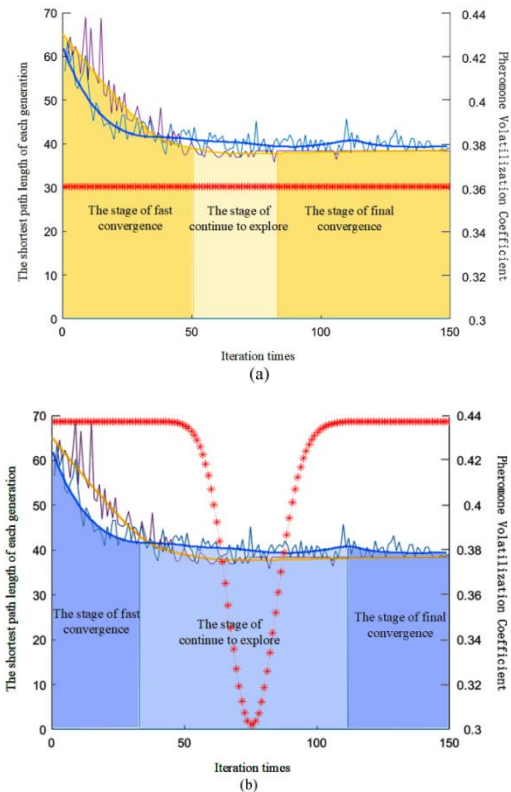


Fig. 11 Convergence curve comparison between traditional ant colony algorithm and improved ant colony algorithm.

In the Fig. 11, we divide the algorithm running process into three stages: the stage of fast convergence (the first stage), the stage of continue to explore (the middle stage), and the stage of final convergence (the final stage). It can be found that by comparing the Fig. 11(a) and Fig. 11(b): In the first stage, the value of dynamic pheromone volatilization factor is higher than the value of static pheromone volatilization factor. This is conducive to the rapid convergence of the algorithm. And the blue curve drops faster than the yellow curve due to the effect of improving the initial pheromone distribution and the dynamic pheromone volatilization factor strategy.

Then, the performance of the curve in the second and third stage can illustrate that, obviously, the purple ployline fluctuates less in the second phase than the blue ployline. Even in the third stage, the purple ployline almost becomes a horizontal straight line. This means that in the middle and late stages of the traditional ant colony algorithm, ants are always on a repetitive path and lose the ability to explore new roads. This is because that the pheromone content on the path plays an important role in guiding the ants to find the way, and the pheromone volatilization factor in the traditional ant colony

algorithm is a global constant. The accumulation of residual pheromones on the same path will always cause this phenomenon in traditional ant colony algorithms. By contraries, the blue ployline not only has a good convergence effect. But also, in the second and the final stages, the blue ployline fluctuation display that the ants still have good exploration ability. On the other hand, the improved algorithm has a wider second stage and the ants explore time of the path is longer in the second stage. Therefore, to a great extent, the application of improved initial pheromone distribution strategy and dynamic pheromone volatilization factor strategy improves global search ability and global search efficiency. By comparing the two algorithms, the improved ant colony algorithm has better ability of path exploration, and its benefits are more obvious than the traditional ant colony algorithm.

In order to further analyze the performance of the algorithm proposed in this paper, it is compared with the conventional ACO and the elite ant colony algorithm. Three maps are used here. The Map 1 is the Fig. 10. The Map 2 and Map3 is the Fig. 12 and Fig. 13 which are 40×40. In the Map 2 and Map3, the number of the generations are $N = 200$, the number of the ants are $M = 1000$ in each generation.

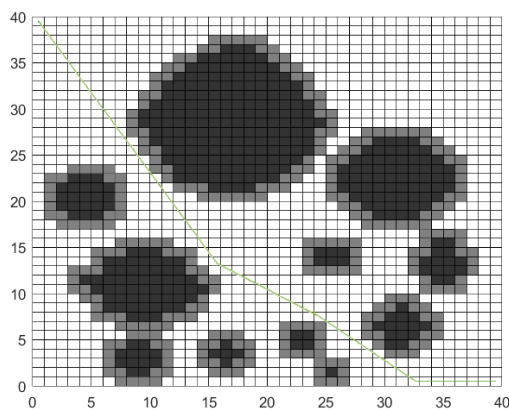


Fig. 12 The simulation of Map 2.

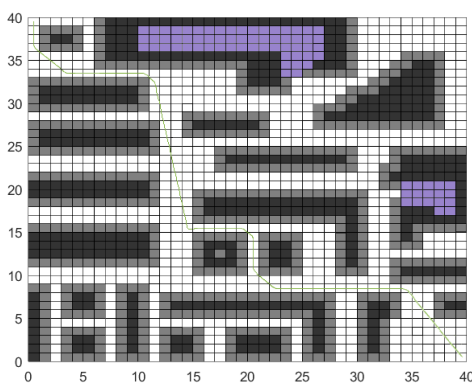


Fig. 13 The simulation of Map 3.

Table 1 Comparison results of three algorithms

	Conventional ant colony algorithm		Elite ant colony algorithm		The algorithm improved in the paper	
	Path length	Search time	Path length	Search time	Path length	Search time
Map 1	78.2	302	50.5	269	37.9	103
Map 2	157.3	2429	87.4	1581	58.9	964
Map 3	216.9	2653	133.7	1835	98.2	1207

In the Table 1, compared with the traditional ant colony algorithm, the improved two algorithms are performs better. Compared with the elite ant colony algorithm, the algorithm proposed in this paper has some improvement in path length and search time. More importantly, the path of the algorithm proposed in this paper is smooth and more suitable for the safe navigation of the Ackerman chassis unmanned vehicle.

Intuitively, the algorithm combines the obstacle expansion strategy, convex polygon detection method and 2-turning smoothing algorithm described above. Compared with the polyline path of the traditional ant colony algorithm, the algorithm proposed in this paper can more reasonably plan a smooth path that satisfies the characteristics of the Ackermann chassis. Under the combination of the guidance of the path and the positioning of its own coordinates, the algorithm can guide the robot to correctly track the path and reach the destination without collision. This algorithm will provide important basis and support for the autonomous navigation of the Ackerman chassis robot.

V. CONCLUSION

In order to plan the path for the Ackerman chassis robot, this paper combines the structural characteristics and motion law of the Ackerman mobile robot to improve the defects of the traditional ant colony algorithm. This paper considers the structural size of the mobile robot into the algorithm through the obstacle expansion strategy. It avoids unnecessary collision of mobile robot due to its own width size during actual operation. Then, in order to improve the efficiency of the ant colony algorithm, the algorithm is improved in many aspects. The application of the convex polygon detection is essentially a purposeful reconstruction of the initial map. The use of this strategy to a certain extent prevents a large number of ants from falling into a U-shaped dead end, thereby improving the efficiency of the algorithm. The reasonable distribution of initial pheromones has a good guiding effect on ants in the early stage of the algorithm, thus to a certain extent avoiding the blindness and randomness of the traditional ant colony algorithm in the early stage of path search. The improvement of dynamic pheromone volatilization factor avoids the premature phenomenon of traditional ant colony algorithm, so that the algorithm can maintain good path exploration ability in the middle and late stages. In order to make the generated path as smooth as possible, we first redesigned the evaluation function, taking the turning of the path into the evaluation function at a certain cost. Then the redundant nodes are used to remove

further smooth paths, and finally the 2-turning algorithm is used to obtain a smooth path suitable for Ackerman chassis vehicles. A smooth path is available for the Ackermann mobile robot. The feasibility and effectiveness of the algorithm are verified by MATLAB platform simulation, which will provide an important basis for the subsequent algorithm migration, and lay a solid foundation for the path tracking control of Ackerman chassis robot.

In addition, the advantage of the algorithm proposed in this paper for path tracking is that it improves the tracking efficiency of Ackerman chassis vehicles and reduces the uncertainty of control during vehicle driving. Generally, the turning and obstacle avoidance of Ackerman vehicles require sensors to monitor the distance between the vehicle and the obstacles in real time, and the control system frequently issues commands to adjust the vehicle attitude based on the detection results of the sensors. The 2-turning algorithm proposed in this paper has planned a path suitable for Ackerman chassis vehicles in advance where the vehicle needs to turn or avoid obstacles. Obstacle avoidance and return movements can be achieved by adjusting the front wheel heading angle at a specific angle twice, which avoids redundant operation of the control system. So it can improve the efficiency of vehicle trajectory tracking. However, there are some defects. Facing a bigger map, it will take a long time to search the optimal path. In the future, we need to reduce search time.

ACKNOWLEDGMENT

This research received the financial support from the National Natural Science Foundation of China (grant no. 31670719).

REFERENCES

- [1] S. Khalid, S. Ullah, N. Ali, A. Alam, I. Rabbi, and I. Ur, "Navigation aids in collaborative virtual environments: comparison of 3DML, audio, textual, arrows-casting," *IEEE Access*, vol. 7, no. 1, pp. 979-989, Oct. 2019.
- [2] W. Yue, Y. Xi, and X. Guan, "A new searching approach using improved multi-ant colony scheme for multi-UAVs in unknown environments," *IEEE Access*, vol. 7, no. 1, pp. 161094-161102, Oct. 2019.
- [3] T. Ren, R. Zhou, J. Xia, and Z. Dong, "Three-dimensional path planning of UAV based on an improved A* algorithm," *IEEE Navigation and Control Conference*, vol. 1, no. 1, pp. 140-145, Jan. 2016.
- [4] L. Chen, Y. Shan, W. Tian, B. Li, and D. Cao, "A fast and efficient double-tree RRT*-Like sampling-based planner applying on mobile robotic system" *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2568-2578, Dec. 2018.
- [5] S. K. Gharghan, R. Nordin, M. Ismail, J. A. Ali, "Accurate wireless sensor localization technique based on hybrid PSO-ANN algorithm for indoor and outdoor track cycling," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 529-541, Sep. 2015.
- [6] H. Yang, J. Qi, Y. Miao, H. Sun, and J. Li, "A new robot navigation algorithm based on a Double-Layer ant algorithm and trajectory optimization," *IEEE Transactions on Industrial Electronic*, vol. 66, no. 11, pp.8557-8566, Dec. 2018.
- [7] X. Wang, T. M. Choi, H. Liu, and X. Yu, "Novel ant colony optimization methods for simplifying solution construction in vehicle routing problems," *IEEE Transactions on Intelligent Transportation System*, vol. 17, no. 11, pp. 3132-3141, Apr. 2016.
- [8] H. Wang, F. Guo, H. Yao, S. He, and X. Xu, "Collision avoidance planning method of USV based on improved ant colony optimization algorithm," *IEEE Access*, vol. 7, no.1, pp.52964-52975, Mar. 2019.
- [9] W. Deng, J. Xu, H. Zhao, "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," *IEEE Access*, vol. 7, no. 1, pp. 20281-20292, Feb. 2019.
- [10] C. Solnon, "Combining two pheromone structures for solving the car sequencing problem with ant colony optimization," *IEEE European Journal of Operational Research*. Vol. 191, no. 3, pp. 1043-1055, Dec. 2008.
- [11] D. Zhang, X. You, S. Liu, and K. Yang, "Multi-Colony ant colony optimization based on generalized jaccard similarity recommendation strategy," *IEEE Access*, vol.7, no. 1, pp. 157303-157317, Oct. 2019.
- [12] C. F. Juang, C. W. Hung, and C. H. Hsu, "Rule-based cooperative continuous ant colony optimization to improve the accuracy of fuzzy system design," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 4, pp. 723-735, Aug. 2014.
- [13] J. Cheng, Z. Miao, and B. Li, "An improved ACO algorithm for mobile robot path planning," in *IEEE International Conference on Information and Automation*, Ningbo, China, 2017, pp. 1-6.
- [14] D. Sudholt, and C. Thyssen, "Running time analysis of ant colony optimization for shortest path problems," *Elsevier Journal of Discrete Algorithms*, vol. 10, no. 2, pp. 165-180, Jan. 2012.
- [15] X. Li, and D. Yu, "Study on an optimal path planning for a robot based on an improved ant colony algorithm," *Springer Automatic Control and Computer Sciences*, Vol. 53, no. 3, pp. 236-243, May. 2019.
- [16] M. Zhang, Z. Jiang, and L. Wang, "Research on parallel ant colony algorithm for 3D terrain path planning," *Springer Modeling, Design and Simulation of Systems*, vol. 10, no. 2, pp. 74-82, Aug. 2017.
- [17] A. Khurana, S. K. Khatri, and A. V. Singh, "Performance evaluation of robot path planning using hybrid TSP based ACO," *Springer International Conference on Advanced Informatics for Computing Research*, vol. 955, no. 2, pp. 448-457, Dec. 2018.
- [18] S. D. Bopardikar, S. L. Smith, F. Bullo, and J. P. Hespanha, "Dynamic vehicle routing for translating demands: stability analysis and receding-horizon policies," *IEEE Trans. Autom. Control*, vol. 55, no. 11, pp. 2554-2569, Apr. 2010.
- [19] L. Kai, and J. Mao, "Path planning of indoor mobile robot based on improved ant colony algorithm," *IEEE Electronic Measurement technology*, vol. 42, no. 11, pp. 65-69, Jan. 2019.
- [20] Q. Yang, W. Chen, Z. Yu, T. Gu, Y. Li, H. Zhang, and J. Zhang, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.* Vol. 21, no. 2, pp. 191-205, Jul. 2017.
- [21] M. Mavrouniotis, and S. Yang, "Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors," *Elsevier Appl. Soft Comp.*, vol. 13, no. 10, pp. 4023-4037, Oct. 2013.
- [22] A. Acan, "An external partial permutations memory for ant colony optimization," in *Springer Evolutionary Computation in Combinatorial Optimization*, vol. 3448, Heidelberg, Germany, 2005, pp. 1-11.
- [23] I. Alaya, C. Solnon, and K. Ghédira, "Ant colony optimization for multi-objective optimization problems," in *IEEE 19th International Conference on Tools with Artificial Intelligence*, vol. 1, Patras, Greece, 2007, pp. 450-457.
- [24] S. Benedettini, A. Roli, and L. D. Gaspero, "Two-level ACO for haplotype inference under pure parsimony," in *Springer 6th International Workshop International Conference on Ant Colony Optimization and Swarm Intelligence*, vol. 5217, Heidelberg, Germany, 2008, pp. 179-190.
- [25] C. Blum, and M. Dorigo, "Search bias in ant colony optimization: on the role of competition balanced systems," *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, pp. 159-174, Apr. 2005.
- [26] M. Liu, X. You, X. Yu, and S. Liu, "KL divergence-based pheromone fusion for heterogeneous multi-colony ant optimization," *IEEE Access*, vol. 7, no. 1, pp. 2646-2657, Oct. 2019.
- [27] X. Sun, K. Zhang, M. Ma, and H. Su, "Multi-population ant colony algorithm for virtual machine deployment," *IEEE Access*, vol. 5, no. 1, pp. 27014-27022, Nov. 2017.
- [28] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels, "The self-organizing exploratory pattern of the argentine ant," *Springer J. Insect Behav.* Vol. 3, no. 2, pp. 159-168, Mar. 1990.
- [29] A. M. Abdelbar, and K. M. Salama, "Parameter self-adaptation in an ant colony algorithm for continuous optimization," *IEEE Access*, vol. 7, no. 1, pp. 18464-18479, Jan. 2019.
- [30] J. Eaton, S. Yang, and M. Gongora, "Special issue on ant algorithms and swarm intelligence," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 317-365, Nov. 2017.

- [31] D. Lin, L. He, X. Feng, and W. Luo, "Niching pareto ant colony optimization algorithm for Bi-Objective pathfinding problem," *IEEE Access*, vol. 6, no. 1, pp. 21184-21194, Apr. 2019.
- [32] M. L. Ibanez, and T. Stutzle, "The automatic design of multi-objective ant colony optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 16, no. 6, pp. 861-875, Feb. 2012.
- [33] F. Massen, M. L. Ibanez, T. Stutzle, and Y. Deville, "Experimental analysis of pheromonebased heuristic column generation using irace," *Springer International Workshop on Hybrid Metaheuristic*, vol. 7919, no. 1, pp. 92-106, Sep. 2013.
- [34] H. Peng, C. Ying, S. Tan, B. Hu, and Z. Sun, "An improved feature selection algorithm based on ant colony optimization," *IEEE Access*, vol. 6, no. 1, pp. 69203-69209, Nov. 2018.
- [35] R. Gan, Q. Guo, H. Chang, and Y. Yang, "Improved ant colony optimization algorithm for the traveling salesman problems," *IEEE Journal of Systems Engineering and Electronics*, vol. 21, no. 2, pp. 329-333, Apr. 2010.
- [36] X. Chen, P. Zhang, G. Du, and L. Fang, "Multiple traveling salesmen problem for multi-robot systems," *IEEE Access*, vol. 6, no. 1, pp. 2169-3536, Apr. 2018.
- [37] M. R. Jabbarpour, H. Zarrabi, J. J. Jung, and Pankoo Kim, "A green ant-based method for path planning of unmanned ground vehicles," *IEEE Access*, vol. 5, no. 1, pp. 1820-1832, Jan. 2017.
- [38] R. Rashid, N. Perumal, I. Elamvazuthi, M. K. Tageldeen, A. Khan, and S. Parasuraman, "Mobile robot path planning using ant colony optimization," in *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, vol. 7753, Ipoh, Malaysia, 2017, pp. 365-372.
- [39] E. Liao, C. Li, "A hierarchical algorithm based on density peaks clustering and ant colony optimization for traveling salesman problem," *IEEE Access*, vol. 6, no. 1, pp. 38921-38933, Jul. 2018.
- [40] A. Rehman, M. M. Rathore, A. Paul, and F. Saeed, "Vehicular traffic optimisation and even distribution using ant colony in smart city environment," *IEEE IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 594-601, Sep. 2018.
- [41] W. Zhang, X. Gong, G. Han, and Y. Zha, "An improved ant colony algorithm for path planning in one scenic area with many spots," *IEEE Access*, vol. 5, no.1, pp.13260-13269, Jul. 2017.
- [42] D. Cai, Y. Yu, and J. Wei, "A modified artificial bee colony algorithm for parameter estimation of fractional-order nonlinear systems," *IEEE Access*, vol. 6, no.1, pp. 48600-48610, Jul. 2018.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US



Guannan Lei was born in Changzhi, City, Shanxi Province, China in 1993. He received the B.S degree in Mechanical engineering and automation from Nanjing Agricultural University, China, in 2017. He is currently studying for a doctorate in control theory and control engineering at Beijing Forestry University, Beijing, China.

From 2017 to 2019, His research interests include driverless and path planning, driverless vehicle navigation, trajectory tracking and control, and mechanical structure design.

Dr. Lei's awards and honors include the First class Postgraduate Scholarship (Beijing Forestry University).



Yili Zheng was born in Nanyang City, Henan Province, China in 1981. He received the B.S degree in automation from National University of Defense Technology Changsha, China, in 2004. In 2009, he received the Ph.D. degree in mechatronics engineering from Beijing University of Posts and Telecommunications, Beijing, China.

From 2009 to 2013, he was a lecturer of Beijing Forestry University. Since 2014, he has been an associate professor at the school of technology, Beijing Forestry University. He is currently an

associate professor of forestry electrification and automation at Beijing Forestry University. He is the author of one book, more than 26 articles, and 8 inventions. His research interests include virtual reality, intelligent forestry equipment, soil moisture sensor and forestry informatization.

Prof. Yili Zheng was a recipient of Ministry of science and technology Liangxi Forestry Science and Technology Award in 2018.