EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Path Planning of Self-driving Vehicles Combining Ant Colony and DWA Algorithms in Complex Dense Obstacles

Jing Niu[1,*], Chuanyan Shen[1] , Jiapei Wei[1], Shifeng Liu[1] and Cheng Lin[2]

[1]Tianshui Normal University, Tianshui Gansu, China
[2]Beijing Institute of Technology, Beijing, China

Abstract

INTRODUCTION: To solve the problems of low quality and weak global optimization of the DWA algorithm, especially the problems of unreasonable path planning and the inability to give consideration to speed and driving safety in the process of vehicles passing through dense obstacles, this paper proposed an improved DWA algorithm based on ant colony algorithm.
OBJECTIVES: The traffic capacity and computing efficiency of Self-driving Vehicles in complex dense obstacles can be greatly improved.
METHODS: Through the obstacle density and distance information obtained by high-precision sensors on the vehicle, the speed objective function is updating in real time by using ant colony algorithm. And the maneuverability and safety performance of vehicles passing through are considering by the way.
RESULTS: The experimental results show that this method can obviously improve the vehicle's traveling ability and uneven path planning in the case of dense obstacles, and the number of iterations of the algorithm is reduced by more than 16%.
CONCLUSION: The improved DWA algorithm integrated with the ant colony algorithm can effectively improve the operating efficiency of the algorithm, reduce the distance the car must go around outside the obstacles, and improve Car driving safety. The effectiveness and universality of the improved DWA algorithm were verified through experiments.

Keywords: Self-driving Vehicles, Dense Obstacles, DWA, Ant Colony Algorithm, Path Planning

Received on 24 December 2023, accepted on 28 March 2024, published on 05 April 2024

doi: 10.4108/ew.5651

1. Introduction

Since the 21st century, Self-driving technology has attracted wide attention because of its obvious advantages in improving vehicle active safety, reducing energy consumption and improving traffic efficiency. Self-driving car is a complex integrated system which integrates the functions of environmental awareness, high-precision map and combined positioning, intelligent decision-making and motion planning[1]. Path planning is a key part of its intelligent decision-making and motion planning system, which determines how the car can reach the designated goal, mainly involving the perception and calculation of environmental information, in order to find a safe and efficient optimal collision-free path. Path planning is mainly divided into two stages: global path planning and local path planning. Global path planning refers to determining the path of a car from the starting point to the target point with the help of high-precision maps and other positioning information. The specific process includes path search, obstacle avoidance, dynamic path adjustment and other steps, and needs to consider traffic rules, road conditions, traffic flow and other factors. Local path planning refers to fine-tuning the driving path on the basis of global path planning according to the real-time environmental information perceived by the car. The specific process mainly considers the dynamic performance of the car and the dynamic changes of obstacles, so as to ensure that the car can drive safely and effectively in the complex traffic environment.

*Corresponding author. Email: sensily@163.com

At present, the algorithms used in path planning mainly include: intelligent algorithms represented by ant colony algorithm, genetic algorithm and particle swarm optimization; Graph-based search algorithm represented by A*, Dijkstra, Hybrid A* and D*. A* algorithm is widely used, but when searching in complex unstructured scenes, it has a large amount of calculation and serious memory consumption, and narrow channels will produce oscillation.Aiming at the problem of slow operation speed of A*, the paper [2] puts forward the jumping point search method, but this method can not guarantee the global optimization of the path in complex irregular maps.Literature [3] adopts the straight line-arc strategy to smooth the path, which greatly improves the smoothness of the path. Literature [4] uses differential method to reduce the number of inflection points, but the amount of calculation increases. Literature [5] optimizes the heuristic function of A* algorithm, improves the selection strategy of key nodes, and reduces path redundancy. Dijkstra algorithm adopts traversal search method, which has a large number of path nodes and low computational efficiency. Local path planning algorithms include artificial potential field method and DWA. DWA algorithm carries out real-time local path planning based on sensor data, which has good obstacle avoidance ability, but can not meet the global path optimization. Reference [6] proposed the Curvature Velocity Method (CVM), described the obstacle avoidance problem as an optimization problem with constraints in velocity space, and established an optimization objective function including three factors: speed, safety and path. On the basis of CVM, a more perfect DWA algorithm is proposed in reference [7]. The objective function comprehensively considers three factors: heading angle, speed and obstacle distance, and the trajectory obtained is relatively smooth, which effectively solves the problem of circling around obstacles[8]-[11].

Considering the high real-time, global, robust and adaptive nature of path generation, the existing DWA algorithm still has the following problems: (1) There are a large number of redundant nodes related to the state in the path search process, which leads to an increase in computation and a longer operation time; (2) In dense obstacle areas, cars tend to choose detours, resulting in longer paths; (3) In complex and irregular scenes, the vehicle's running track is not smooth, and it is easy to oscillate.

Based on the above analysis, this paper puts forward an improved global path planning method of DWA road, which combines ant colony algorithm to solve the problems of increasing path nodes, blind search, unsmooth vehicle trajectory and long trajectory in complex and dense obstacle scenes, obtains the map information of surrounding environment by vehicle sensors, and applies the idea of parameter adaptation to construct an adaptive DWA algorithm to obtain a more reasonable and safe global optimal path.

2. Ant colony algorithm model

The basic principle of ant colony algorithm has been explained in detail in DORIGO M[7][8]and other works. The following only briefly explains the state transition rate and pheromone increment model in the core of the algorithm.

2.1 State transition rate

At time t, the state transition rate of ants moving from state node I to adjacent state node J can be defined as:

$$P_{ij}^{m}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\cdot\eta_{ij}^{\beta}(t)}{\sum_{s\in U^m}\tau_{is}^{\alpha}(t)\cdot\eta_{is}^{\beta}(t)}, & \text{if } j \in U^m \\ 0, & \text{else} \end{cases} \qquad (1)$$

In the formula:$P_{ij}^{m}(t)$is the state transition probability of the m-th ant moving from state node I to state node J at time T.$\tau_{ij}^{\alpha}(t)$is the pheromone concentration on the path (i,j), where α is the information heuristic factor[8][9], reflecting the influence of pheromone on the ant's path selection;$\eta_{ij}^{\beta}(t)$ is the heuristic function of the m-th ant to select the adjacent state node J at the state node I, where β is the expected heuristic factor, reflecting the importance of heuristic information in guiding the ant colony search process;$U^m$is the next node set that the ant has not visited;S is an optional node set adjacent to the current position;$\tau_{is}^{\alpha}(t)$is the pheromone concentration of the m-th ant between the current state node I and the adjacent state nodes; $\eta_{is}^{\beta}(t)$ is the heuristic function of the m-th ant between the current node I and the adjacent state nodes.

Heuristic function $\eta_{ij}(t)$ can be expressed as:

$$\eta_{ij}(t) = 1/D_{ij} \qquad (2)$$

In the formula:D_{ij} is the distance between state node I and state node J.

2.2 Pheromone concentration update model

At present, the common pheromone concentration updating models are Ant-Density System(ADS), Ant-Quantity System(AQS) and Ant-Cycle System(ACS)[10]. ADS model and AQS model adopt local updating strategy, while ACS model adopts global updating strategy[10]. Considering the solution speed and obstacle avoidance ability of the algorithm, this paper adopts AQS model as the prototype.

Suppose that the set of state nodes of path (i,j) that the m-th ant currently circulates through is $X\{(i,j)|i = 1,2,\dots,n; j = 1,2,\dots,n\}$,so:

$$\Delta\tau_{ij}^{m}(t) = \begin{cases} \dfrac{Q}{D_{ij}}, & \text{if}(i,j) \in X \\ 0, & \text{else} \end{cases} \qquad (3)$$

In the formula: $\Delta\tau_{ij}^{m}(t)$ is the path pheromone concentration increment of the m-th ant moving from the state node I to the adjacent state node J from time t-1 to time T;Q is the pheromone intensity, which is a constant greater than 0.

3. Improved DWA algorithm design

3.1 Classical DWA algorithm

The implementation process of the classical DWA algorithm is to convert the position control of the self-driving car into the speed control, and describe the obstacle avoidance problem as an optimization problem with constraints in the speed space of the car[11], including the speed of the car, the driving direction and the position constraints of obstacles in the surrounding environment[12].

As determined by the physical constraints of obstacles around the vehicle trajectory node, the speed set $U_s(u_a, \omega_r)$ composed of the longitudinal speed of the autonomous vehicle and the angular velocity limit of the yaw which determines the direction of the vehicle must be met:

$$U_s = \{(u_a, \omega_r)|0 \le u_a \le u_{amax}, -\omega_{rmax} \le \omega_r \le \omega_{rmax}\} \quad (4)$$

The vehicle trajectory can be considered to be composed of n broken line segments in n time periods, and the connection point of the broken line segment is considered to be close to the position of the obstacle on the premise of meeting the expansion size limit of the obstacle[13]. In order to ensure that obstacles encountered in the process of automobile movement do not collide, after the time dt can be obtained by kinematic conditions,The speed set U_a must satisfy:

$$U_a = \left\{(u_a, \omega_r)\middle|u_a \le \sqrt{2\cdot dist(u_a,\omega_r)\cdot \dot{u_a}}, \omega_r \le \sqrt{2\cdot dist(u_a,\omega_r)\cdot \dot{\omega_r}}\right\} \quad (5)$$

In the formula: $dist(u_a, \omega_r)$ indicates the linear distance between the car and the obstacle at the next moment.

Assume that the speed set of the self-driving car at the current moment is $(u_{acurr}, \omega_{rcurr})$, then the speed set U_d at the next moment must satisfy:

$$U_d=(u_{ad}, \omega_{rd})= \begin{cases} u_{acurr} - \dot{u_{amax}} d_t \le u_a \le u_{acurr} + \dot{u_{amax}} dt \\ \omega_{rcurr} - \dot{\omega_{rmax}} d_t \le \omega_r \le \omega_{rcurr} + \dot{\omega_{rmax}} dt \end{cases} \quad (6)$$

The final speed set u can be expressed as:

$$U = U_s \cap U_a \cap U_d \quad (7)$$

The car predicts the speed set at the next moment through the objective function. The objective function defined in this paper comprehensively considers the moving speed, orientation and collision safety, as shown below:

$$G(u_{ad}, \omega_{rd}) = l\cdot\theta + m\cdot dist(u_a, \omega_r) + n\cdot u_{amax} \quad (8)$$

Where: θ indicates the included angle between the driving direction of the vehicle and the target line; $dist(u_a, \omega_r)$ indicates the shortest distance between the car position and the obstacle; L, M and N are three weight coefficients respectively, which are usually normalized into constant coefficients between [0,1].

The objective function $G(u_{ad}, \omega_{rd})$ corresponding to each possible trajectory of the car's movement is calculated through formula (8),The speed set that maximizes the function value is the optimal set. Each weight coefficient in the common DWA algorithm is fixed[14]. Using the grid method to model and run the environment map of the self-driving car, it can be seen that when all the weight coefficients are large, the number of running steps for the car to move from the starting point to the target point is relatively small, and the calculation time is short (as shown in Figure 1). In the case of a smaller angle weight coefficient, the number of running steps is too many and the calculation time is long (as shown in Figure 2). In the case of a small distance and speed weight coefficient, the number of running steps increases significantly and the car detours. The calculation time increases significantly (as shown in Figure 3).

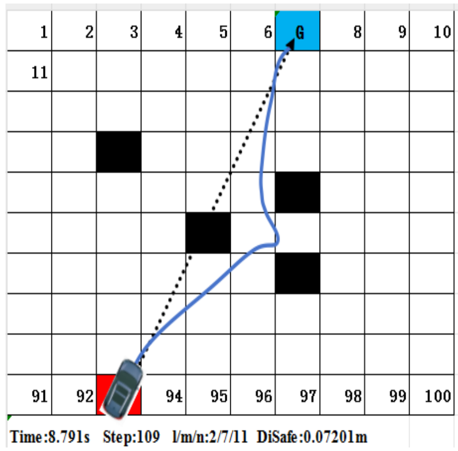

Figure 1. High weight coefficients trajectory

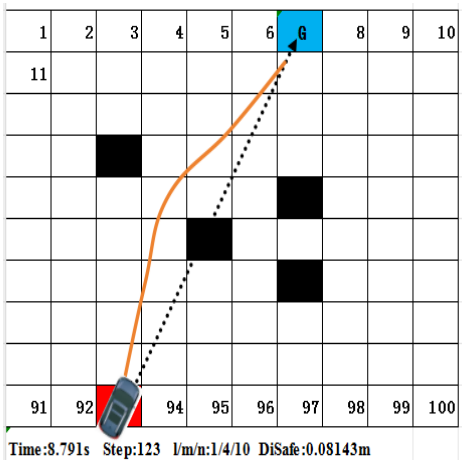

Figure 2. Low angle weight coefficient trajectory

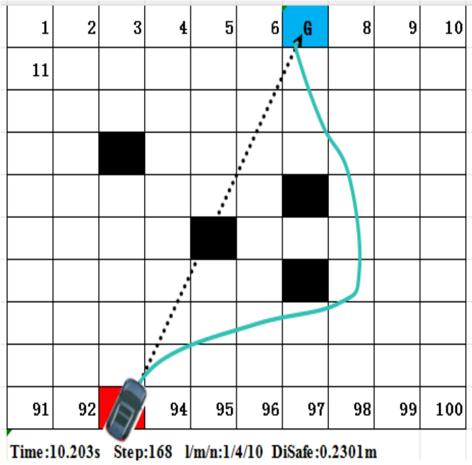

Figure3. low distance and low speed weight coefficients trajectory

Therefore, when a self-driving car is moving at high speed, if the density of surrounding obstacles is too high, the classic DWA algorithm will cause the car's passability to significantly deteriorate, and the planned path will make the car too close to the obstacles, significantly reducing driving safety. Especially when the density of obstacles further increases, the above algorithm will cause the car to go around multiple obstacles, resulting in a path that is too long. In some cases, the path will not be smooth, causing the car to oscillate between multiple obstacles. The fundamental reason for the above problems is that once the weight coefficient in the vehicle speed planning objective function is determined, it cannot adapt to various complex and changeable environmental factors.

This paper combines the advantages of the ant colony intelligence algorithm in global optimization and continuous convergence of the search process, and considers the three weights of the above objective function from the aspects of reducing the car's detour distance between dense obstacles and improving driving maneuverability and safety[15-20]. The coefficients are all iterated and updated globally according to the ant colony algorithm. The specific process of dynamically updating the above weight coefficients based on the ant colony algorithm model will be elaborated below.

3.2 Improvement of DWA algorithm integrating ant colony algorithm

The high-precision radar installed in self-driving cars can quickly detect the size and distance information of obstacles in the surrounding environment. Assume that at the current time t, there are obstacles with a certain density in the area in the direction of the car, as shown in Figure 1.

If the number of obstacles in the area is M,The shortest distance between the car and the i-th obstacle is D_i, The azimuth angle is θ_i. Define that when M is greater than the threshold, the area is a dense area of obstacles.Define the shortest distance between the i-th obstacle and the j-th obstacle as D_{ij}:

$$D_{ij} = \sqrt{D_i^2 + D_j^2 - D_i D_j \cos(\theta_i - \theta_j)}, \quad \theta_i \gg \theta_j \tag{9}$$

Taking into account the safety and maneuverability of a car passing between obstacles, in order to measure the passability of a car between two obstacles, the car's passing function number D_s is defined as:

$$D_s = a \cdot \frac{\theta_{max}}{\omega_{ramx}} + b \cdot \frac{u_{amx}}{\dot{u}_a} \tag{10}$$

In the formula: ω_{rmax} is the maximum value in ω_r, θ_{max} is the maximum value in θ_i, a and b are constants. The larger the azimuth angle, the easier it is for the car to pass through, and this number is directly proportional to it. The greater the yaw angular velocity of the car, the worse the handling stability and the reduced driving safety. This number is inversely proportional to it. The greater the maximum speed of the car, the easier it is for the car to pass through, and this number is directly proportional to it. The greater the vehicle's longitudinal acceleration, the weaker its braking ability and reduced driving safety, and this number is inversely proportional to it.

Taking into account the expansion radius of the obstacle and introducing the expansion radius influence coefficient σ, the conditions for a car to pass safely between two obstacles is:

$$D_s > \frac{D_{ij}}{\sigma} \tag{11}$$

Then the update model of dynamic pheromone is as follows:

$$\Delta\tau_{ij}^m = \begin{cases} \frac{\sigma \cdot D_{max} - D_{ij}}{D_{ij} - \sigma \cdot D_{min}}, \delta > \varepsilon \\ \frac{D_{ij} - \sigma \cdot D_{max}}{D_{ij} - \sigma \cdot D_{min}}, \delta \le \varepsilon \end{cases} \tag{12}$$

In the formula: $\delta = D_{max} - D_{min}$, ε is the acceptable error for the nth iteration, which is a constant, D_{max} refers to the maximum value of the traversing function number obtained by moving between any two obstacles after the car moves to the local obstacle avoidance area, D_{min} refers to the maximum value of the traversing function number obtained by moving between any two obstacles after the car moves to the local obstacle avoidance area. The minimum value of the number of travel functions.

3.3 Algorithm flow

Step 1: Obtain environmental map information from vehicle-mounted sensors, locate the starting point and target point of vehicle movement; obtain information on all state nodes in the space, and calculate the adjacent matrix and heuristic information matrix.

Step 2: Parameter initialization. The number of initialization iterations is N, the ant colony size M, the information heuristic factor α, the expectation heuristic factor β, the pheromone volatilization coefficient ρ, and the pheromone concentration τ.

Step 3: Calculate the density of obstacles, the actual distance and orientation between the car and each obstacle in real time.

Step 4: Path selection update. Query the adjacent matrix to obtain the set of nodes where it is feasible for the current node i to move to the next node, and calculate the probability of the m-th ant selecting the adjacent node according to formulas (1) to (3). During the node update process, based on the map information captured by the vehicle-mounted sensor, the obstacle density, actual distance and orientation between the car and each obstacle and other information are used to determine whether it has entered the obstacle-dense area: if it has entered, go to steps five to seven; If you do not enter, all weight coefficients in step five will be set to fixed values.

Step 5: Eliminate the node set with $\mathrm{dist}(u_a, \omega_r)$ greater than the threshold from the global path planning nodes, and update the ant sequence number.The on-board processor calculates the distance D_{ij} between the obstacles and the car's passing function number D_s, obtains the D_{max} and D_{min} values, calculates and updates the pheromone, and obtains the dynamically updated weight coefficients l, m and n.

Step 6: Obtain the alternative speed space $U(u_a, \omega_r)$ according to formula (8), normalize the three weight coefficients respectively, and obtain the optimal speed set of the car at time t+1.

Step 7: Execute the speed to determine whether the target point is reached: if so, end the iteration process; otherwise, return to step 1.

4. Simulation test and analysis

4.1 Simulation parameter initialization

In order to verify the global performance of the improved DWA algorithm, this paper uses the raster method to construct an environment map of 10×10 (as shown in Figure 1), and conducts autonomous vehicle path planning experiments on this basis. In Figure 1, the red square represents the starting point and the target point, the black square represents the obstacle, and the white square represents the open space[24]. The area of the square reflects the expanded size of the car and obstacles, and the dotted line with an arrow represents the initial start of the global planned path. Value, the colored solid line represents the actual running path. The parameter selection during simulation operation is shown in Table 1 and Table 2. The location information of the starting point and target point is shown in Table 3.

Table 1. DWA algorithm parameters

Puffing radius m	Time resolution s	Resolution of linear speed m/s	Resolution of angular velocity rad/s	Prediction time s	Distance threshold m
0.5	0.01	0.02	1	30	0.5

Table 2. Ant colony algorithm parameters

K	M	α	β	ρ	Q
100	80	2	7	0.2	100

Table 3. Position information of starting point and target point

parameter	x/m	y/m	θ/rad	u_a/(km/h)	ω_r/(rad/s)
starting point	0.25	0.05	0.68	10	0.20
Target point	0.65	0.95	0.21	0	0

4.2 Algorithm optimization results

The DWA algorithm integrated with the ant colony algorithm can significantly speed up the local path dynamic update capability when the car approaches the obstacle area. Through the obtained optimal objective function, the car's movement speed, driving direction and collision safety can be guaranteed. As shown in Figure 4, when the raster map enters the obstacle-dense area, at time 10.450s, the predicted trajectory line of the classic DWA algorithm is green, and the trajectory line of the improved DWA algorithm fused with the ant colony algorithm is red. It can be seen that the car driving path represented by the green line has increased significantly. At this time, the distance between the car and the nearest obstacle, vehicle speed, yaw angular velocity and objective function in the two situations are shown in Table 4 below. At this moment, the objective function obtained by the classic DWA algorithm is significantly greater than that of the improved DWA algorithm. According to the algorithm logic of maximizing the objective function, the classic DWA algorithm will cause the car to continue to increase its speed and deflection angle, further bypassing the obstacle-dense area, and finally causing path growth.

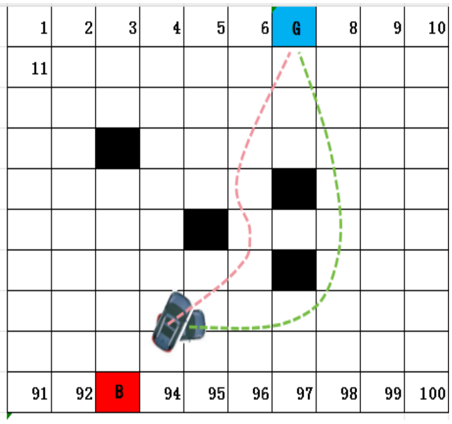

Figure4. Predicted trajectory at 10.450s

Table 4. Car driving status information

parameter	dist/m	u_a/(km/h)	ω_r/(rad/s)	G
Classic DWA	0.309	36	0.20	0.91
Improve DWA	0.167	27	0.78	0.67

Improved DWA test under changing vehicle speed

As shown in Figure 4: the car's starting speed is 10km/h, the speed is relatively slow, and some oscillations occur between the three obstacles. In order to verify the impact of vehicle speed on the improved DWA algorithm, the initial vehicle speed of the vehicle was increased to 50km/h, and the predicted trajectory as shown in Figure 5 was obtained. It can be seen that the trajectory oscillation is significantly eliminated as the vehicle speed increases.

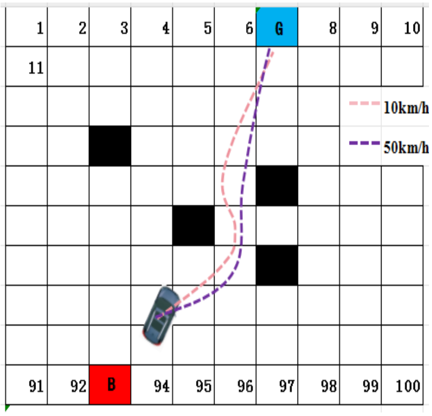

Figure 5. Predicted trajectory of changing starting speed

Figure 6 shows the simulation results of the objective function in this case. It can be seen that when the three weight coefficients of the objective function are iterated by the ant colony algorithm, if the car is far away from the obstacle, the optimized alternative speed set is higher; if the car is closer to the obstacle, the optimized speed set The alternative speed set is lower. The above algorithm logic is fully consistent with the guiding ideology of dynamically adjusting the speed and trajectory of autonomous vehicles, ensuring the mobility and safety of the vehicle in dense obstacle areas.

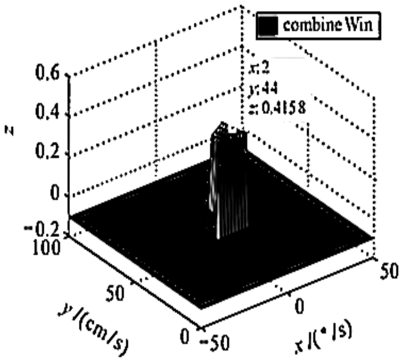

Figure6. 3D diagram of the objective function at 50km/h

Improved DWA test when the number of obstacles changes

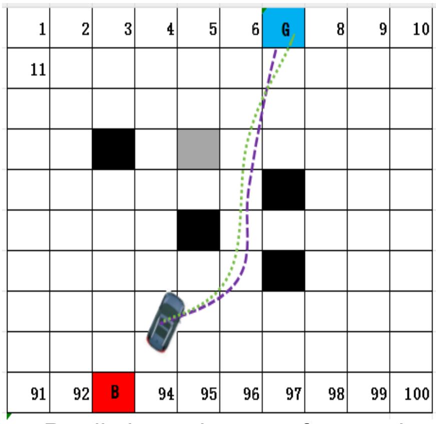

Figure7. Predictive trajectory of more obstacles

As shown in Figure 7: after adding a gray obstacle to the grid map, the initial vehicle speed is set to 50km/h. Run the improved DWA algorithm and find that: the new predicted trajectory represented by the green line is slightly weaker than the previous oscillation. Under the premise of ensuring safety, the overall efficiency of the car's movement from the starting point to the target point has been significantly improved. Calculating the number of iterations it was reduced by 20.69%, and the total running time dropped by 16.78%. Figure 8 shows the simulation results of the objective function in this case.

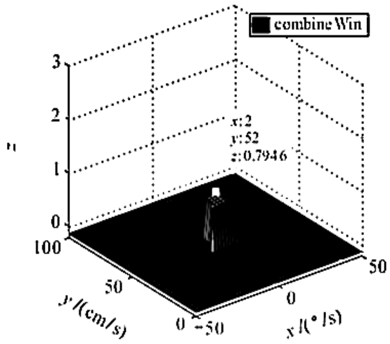

Figure8. 3D diagram of the objective function after increased obstacles

In order to quantitatively compare the comprehensive performance before and after the improved algorithm in a variety of different scenarios and verify the universality of the algorithm, due to space limitations, Table 5 below only lists the comparison results of several different obstacle numbers. It can be seen that according to the improved DWA algorithm for car path planning, increasing the number of obstacles will have little impact on the car's motion performance.

Table 5. Comparison results of different number of obstacles

Scenes number	Safety dist /m	Number of iteration steps /step	Total time /s	total distance /m
3	0.246	339	17.801	15.320
4	0.213	350	18.091	15.231
5	0.177	407	17.443	14.998

5. Conclusion

This article proposes a path planning method for self-driving cars to drive in areas with high density of obstacles. The improved DWA algorithm integrated with the ant colony algorithm can effectively improve the operating efficiency of the algorithm, reduce the distance the car must go around outside the obstacles, and improve Car driving safety. The effectiveness and universality of the improved DWA algorithm were verified through experiments. This method significantly improved the convergence speed of the optimization of the global path of autonomous vehicles. In the future, we will further study the applicability of the DWA algorithm in dynamic obstacle scenarios on this basis.

References

[1] Harikirshnan Vijayakumar, Dezong Zhao, Jianglin Lan, Wenjing Zhao, Daxin Tian, Dachuan Li, Quan Zhou, Kang Song, "A Holistic Safe Planner for Automated Driving Considering Interaction With Human Drivers," in proceedings of the IEEE Transactions on Intelligent Vehicles Conference, 2023, pp. 1-6.
[2] Zhao Xiao, Wang Zheng, Huang Chengkan, etc, "Mobile robot path planning based on improved A* algorithm," Robot, 2018, pp. 903-910.
[3] Zhang Wen, Liu Yong, Zhang Chaofan, etc, "Real-time path planning of greenhouse robot based on directional A* algorithm," Journal of Agricultural Machinery, 2017, vol. 48, no. 7, pp. 22– 28.
[4] TROCATO K， DORST L. Differential A*, "IEEE Transactions on Knowledge and Data Engineering," 2002, vol.14, no. 6, pp. 1218-1229.
[5] Cheng Chuanqi, Hao Xiangyang, Li Jiansheng, etc, "Global dynamic path planning integrating improved A* algorithm and dynamic window method," Journal of Xi'an Jiaotong University, 2017, vol. 51, no. 11, pp.137-143.
[6] Jian hua Zhang, Qi Feng, Ai di Zhao, Wei He, Xun Hao, "Local path planning of mobile robot based on self-adaptive dynamic window approach," in proceedings of ACAE2020, 2021, pp.1472-1490.
[7] Fox D, Burgard W, Thrun S, et al, "The dynamic window approach to collision avoidance," Robotic &Automation Magazine, 1997, vpl. 4, no.1, pp. 23-33.
[8] Bao Min, Yang Shipin, Xu Yunchen, Li Lijuan,"An Improved Ant Colony Algorithm for Traveling Salesman Problem," in proceedings of IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2019, pp. 796- 799.
[9] Seder M,Petrovic I, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles,"in proceedings of IEEE Int Conf on Robotics and Automation, 2007, pp. 1986-1991.
[10] Songzhu Xia, Su Wu, "Ant Colony-Based Energy Aware Multipath Routing Algorithm for Wireless Sensor

Networks," in proceedings of 2009 Second International Symposium on Knowledge Acquisition and Modeling, 2009, pp. 1098-1099.

[11] Xiaoyu Yuan, Chengchang Tong, Guoxiang He, Hongbo Wang, "Unmanned Vessel Collision Avoidance Algorithm by Dynamic Window Approach Based on COLREGs Considering the Effects of the Wind and Wave," in proceedings of Journal of Marine Science and Engineering, 2023, pp. 897-900.

[12] Doopalam Tuvshinjargal, "Hybrid motion planning method for autonomous robots using kinect based sensor fusion and virtual plane approach in dynamic environments," Sensors, 2015, vol.5, pp.1-13.

[13] AnaLopes A, Rodrigues J, PerdigaoJ, et al, "A new hybrid motion planner: Applied in a brain-actuated robotic wheelchair," in proceedings of IEEE Robotics & Automation Magazine, 2016, vol. 23, no.4, pp. 82-93.

[14] Beomjoon Kim, "Socially adapti ve path planning in human environments using inverse reinforcement learning," Social Robotics, 2016, vol.8, no.1, pp. 51-66.

[15] Ren Jianhua, Li Wenchao, Zhao Kailong, etc. Research on mobile robot path planning method [J]. Electromechanical Technology, 2019, 42 (4): 26 29.

[16] He Zhiming, Zheng Li, Liang Wen, "Vehicle path planning based on adaptive dynamic search ant colony algorithm," Computer Engineering and Design, 2021, vol.42, no.2, pp.543-551.

[17] Yang Xu, Shen Peng, Feng Liu, et al, "Potential field ant colony algorithm for autonomous vehicle path planning under reachable set constraints," Science and Technology Bulletin, 2020, vol.36, no.4, pp.113-119.

[18] Zhang Jian, Fan Xiaowu, "Highway collaborative rescue path planning based on improved ant colony algorithm," Computer Age, 2021, vol.3, no.3, pp. 1-16.

[19] Zhao Mentong, Li Yinghong, Fan Jingjing, "Research on time-optimal path planning based on improved ant colony algorithm," Vehicle and Power Technology, 2020, vol.3, pp. 7-10.

[20] Li Yi, Tang Qian, Liu Lianchao, et al, "Solution to automobile mixed-flow assembly scheduling model based on improved ant colony algorithm," China Mechanical Engineering, 2021, vol.32, no.9, pp. 1126-1133.

[21] Qiang Ning, Gao Jie, Kang Fengju, "Multi-robot global path planning based on PSO and cubic spline interpolation," Journal of System Simulation, 2017, vol.29, no.7, pp.1397-1404.

[22] Zhao Di, Yu Liping, Hu Mengya, et al, "Research on lunar detector path planning optimization algorithm," Mechanical Science and Technology, 2021, vol.40, no.3, pp. 364-370.

[23] Hu Lihua, Ma Rui, Zhang Mingshi, etc, "Intelligent car path planning method based on improved ant colony algorithm," Journal of Taiyuan University of Science and Technology, 2020, vol.41, no.6, pp. 463-469.

[24] Shi Xiangnan, Yang Yanbo, Xu Qiwei, Li Teng, Zhang Jiawei, "Research on grassland grazing environment planning based on GA-ACO hybrid algorithm," in proceedings of 2022 International Conference on Networking and Network Applications, 2022, pp. 324-328.