PATH PLANNING STUDY OF INTELLIGENT GRAIN TRANSPORTER BASED ON GRAIN DEPOT SCENARIO
/
基于粮库场景下智能粮食转运车的路径规划研究

Boqiang ZHANG[1)], Genliang YANG[1)], Liang LI[*2)], Chenglong ZHANG[1)], Xun ZHANG[1)], Xuemeng XU[1)], Xiaoling WU[2)]
[1)] School of Mechanical and Electrical Engineering, Henan University of Technology, Zhengzhou 450001 / China;
[2)] YUTONG BUS CO.,LTD. Zhengzhou 450061 / China;
Tel: 15981863085; E-mail: pretty510@163.com
Corresponding author: Liang LI
DOI: https://doi.org/10.35633/inmateh-75-99

Keywords: *Grain depot scenario，Hybrid A* algorithm, Dynamic node extension, KD-Tree, Path planning*

ABSTRACT
To address the problems of low search efficiency, long time-consuming planning and poor adaptability to narrow passages of the original Hybrid A algorithm in path planning for intelligent grain transfer vehicles in grain depot scenarios, a Hybrid A* algorithm with variable resolution and variable step size is proposed. First, the dual-heuristic function and the cost function with steering and reversing penalties are reasonably designed to ensure that the algorithm can search for a derivable path. Second, the distance cost based on the KD-Tree algorithm is combined into the node* extension*, and the node* extension *is performed using variable resolution and variable step size according to the change of the distance cost to improve the node search efficiency. Then, Reeds-Shepp curve search fitting is used to pinpoint the target point pose. The simulation validation results show that the improved Hybrid A* algorithm reduces the number of search nodes by 83% and the planning time by 37% in simple maps, while the number of search nodes and the planning time are reduced by 80.6% and 56.6%, respectively, in complex maps, which improves the path search efficiency. After the real-vehicle test, the improved Hybrid A* algorithm reduces by 52.9% and 61.4% in terms of the number of search nodes and planning time consumed, respectively, and there is a significant improvement in the planning efficiency, which enhances the operational efficiency of the grain transfer vehicle.*

摘要
*针对粮库场景中智能粮食转运车辆在路径规划时原始 Hybrid A*算法存在搜索效率低、规划耗时长及狭窄通道适应性差等问题，提出一种变分辨率和变步长的 Hybrid A*算法。首先，合理设计双启发函数与含有转向和倒车惩罚的代价函数，保证算法能够搜索到一条可行驶的路径。其次，将基于 KD-Tree 算法的距离代价结合到节点扩展中，根据距离代价的变化使用变分辨率和变步长进行节点扩展，提高节点搜索效率。然后，采用 Reeds-Shepp 曲线搜索拟合精确定位目标点位姿。仿真验证结果表明，在简单地图中，改进 Hybrid A*算法的搜索节点数量缩短了 83%，规划耗时缩短了 37%；在复杂地图中，搜索节点数量和规划耗时分别缩短了 80.6%，56.6%，，提高了路径搜索效率。经过实车试验，改进 Hybrid A*算法在搜索节点数量和规划耗时方面分别缩短了 52.9%、61.4%，规划效率有明显提升，从而提升了粮食转运车辆的运行效率。*

INTRODUCTION
Granaries are a critical component of food storage (*Nayak et al., 2020*). However, traditional grain depots face several challenges in the processing and transfer of residual grain. These include narrow passages, complex environments with numerous grain piles, and reliance on manually operated vehicles for transportation, which results in low efficiency and high operational costs. In order to overcome this problem, intelligent grain transfer vehicle has gradually become a research hotspot (*Teng et al., 2023*). By integrating advanced sensors, high-precision positioning system and intelligent decision-making module *(Peng et al., 2021*), the intelligent grain transfer vehicle can quickly and accurately sense the obstacles in the known surrounding environment and make obstacle avoidance decisions.

Boqiang ZHANG, Senior engineer Ph.D. Eng.; Genliang YANG, M.S. Stud. Eng; Liang LI, Engineer ;
Chenglong ZHANG, M.S. Stud. Eng.; Xun ZHANG, Lect. PhD. Eng.; Xuemeng XU, Prof. Ph.D. Eng.; Xiaoling WU, Engineer

The movement of the intelligent grain transporter is inseparable from the path planning, and a good path planning algorithm can improve the efficiency of the transporter and reduce the waste of grain. At the same time, the application of automatic driving technology to the residual grain transfer in grain depot scenarios is of great significance in guaranteeing the quality of grain and the construction of smart grain depots (*Tang et al., 2024*).

Path planning, as one of the key technologies in autonomous driving technology, is to generate optimal paths that satisfy collision avoidance constraints, kinematic constraints (e.g., minimum turning radius and speed constraints), and dynamic constraints such as acceleration constraints, deceleration constraints in structured roads (e.g., urban roads) or unstructured road scenes (e.g., harbors, grain warehouses, etc.) (*Xiong et al., 2020*), which is of great significance to improve vehicle driving safety and the efficiency of residual grain transfer in grain warehouses. At present, the commonly used path planning algorithms include three major categories: ant colony algorithm (*Wu et al., 2023*) based on machine learning method, PRM algorithm *(Huang et al., 2024*) and RRT algorithm (*Yin et al., 2023*) based on sampling method, and Dijkstra algorithm *(Dijkstra, 2022*) and A* algorithm *(He et al., 2022*) based on graph search method. However, none of these path planning algorithms consider vehicle kinematic characteristics when planning paths, resulting in limitations of the planned paths in grain depot scenarios. For this reason, Dolgov et al. proposed the Hybrid A* algorithm that considers satisfying the vehicle kinematic characteristics, which is able to plan a driving path close to the global optimum due to its novel node extension, but its planned paths will suffer from the problems of curvature not being able to maintain continuity, poor quality of the paths, and inefficient searching, which is still difficult to satisfy the practical needs (*Dolgov et al., 2010*). Therefore, some scholars are attracted to study the Hybrid A* algorithm. Aiming at the problems that the curvature of the paths planned by Hybrid A* algorithm cannot be kept continuous, the path quality is poor and the search efficiency is low, scholars carry out the improvement research from the aspects of path quality and search efficiency.

In the realm of path quality enhancement, diversified methodologies have been proposed to address trajectory optimization. Deng et al. (*Deng et al., 2022*) implemented a segmented spline curve optimization framework based on quadratic programming, while Zhang (*Zhang, 2024*) investigated the motion trajectory of warehouse robots by deriving spatial kinematic equations and introducing an enhanced Ant Colony Optimization-Dynamic Window Approach (ACO-DWA) algorithm. Ren et al. (*Ren et al., 2022*) innovatively designed a variable-radius Reeds-Shepp curve, integrating it with segmented Bessel curves and gradient descent techniques to ensure path smoothness. Concurrently, Tang et al. (*Tang et al., 2021*) synergized the artificial potential field method with the Hybrid A* algorithm, collectively mitigating discontinuities in curvature and improving navigational safety. Regarding search efficiency optimization, Tian et al. (*Tian et al., 2023*) augmented the heuristic function through distance field mapping, whereas Cao et al. (*Cao et al., 2023*) refined heuristic value acquisition by incorporating bidirectional path search strategies. Qin et al. (*Qin et al., 2024*) introduced collision risk cost into node extension mechanisms, and Li (*Li, 2024*) proposed an obstacle feature extraction method via raster map preprocessing to optimize ant colony algorithms. Furthermore, Sedighi et al. (*Sedighi et al., 2019*) enhanced path exploration efficiency by integrating Voronoi diagrams with the Hybrid A* framework. Although the Hybrid A* algorithm has made significant progress in both path quality and search efficiency, the existing methods still face challenges in grain depot scenarios. Especially in the complex environment of narrow aisles and many obstacles in the grain depot, the existing methods are still deficient in dynamically adjusting the resolution and the linkage between the step size and the distance to the obstacles.

The original Hybrid A* algorithm has the problems of low node search efficiency, long time consumption and poor adaptability to narrow channels when planning paths in the grain depot scenario. In this paper, a Hybrid A* algorithm with variable node resolution and variable step size is proposed, which mainly makes reasonable design of the heuristic function, the actual cost function and reasonable improvement of the node extension method. Among them, the dynamic node extension method is evaluated with the distance cost based on the KD-Tree algorithm, which changes the static extension into the dynamic extension with variable resolution and variable step size, and improves the path search efficiency.

MATERIALS AND METHODS

Hybrid A* algorithm

The Hybrid A* algorithm is a graph search algorithm improved from the A* algorithm, and its planned path takes into account the kinematic constraints of the vehicle, so that a path close to the global optimum can be planned.

The idea of Hybrid A* algorithm is the same as that of A* algorithm, both are heuristic search. The cost function of Hybrid A* algorithm is as follows:

$$f(n) = g(n) + h(n) \tag{1}$$

In equation (1), the defining domains of the cost function *f(n)*, the actual cost *g(n),* and the heuristic function *h(n)* are all the nodes N in the search space, and the value domain R is a non-negative real number; where *n* is the current node, *g(n)* is the actual generation value from the starting node to the current node, and *h(n)* is the heuristic generation value from the current node to the goal point, and *f(n)* integrates the actual cost from the starting point to the current point and the estimated cost to the goal point.

The Hybrid A* algorithm searches as described below:

First, the Hybrid A* algorithm (*Cao, 2023*) receives the start position, end position, and a 2D raster map containing vehicle position coordinates and heading angle information. It performs path planning within this static 2D raster map (*Elfes, 1989*), which includes obstacle data and passable areas. The algorithm initializes an empty Open list and Closed list to manage the search nodes and inserts the start node into the Open list. Next, the algorithm selects the current node for expansion based on the heuristic cost to the target point, choosing the node with the lowest cost function value *f(n)* from the Open list. It then checks whether this node can be connected to the goal using an obstacle-free Reeds-Shepp curve (*Reeds & Shepp, 1990*). If a valid connection is found, the curve is accepted as the final path, and the algorithm terminates. If no direct connection is possible, the current node is moved to the Closed list and treated as a parent node. Child nodes are generated based on vehicle kinematic constraints, and the cost function *f(n)* is calculated for each. These child nodes are then added to the Open list. The process repeats, expanding nodes, checking for connections, and updating the Open and Closed lists, until either a valid path is found or the Open list is empty. The overall workflow of the algorithm is illustrated in Fig. 1.

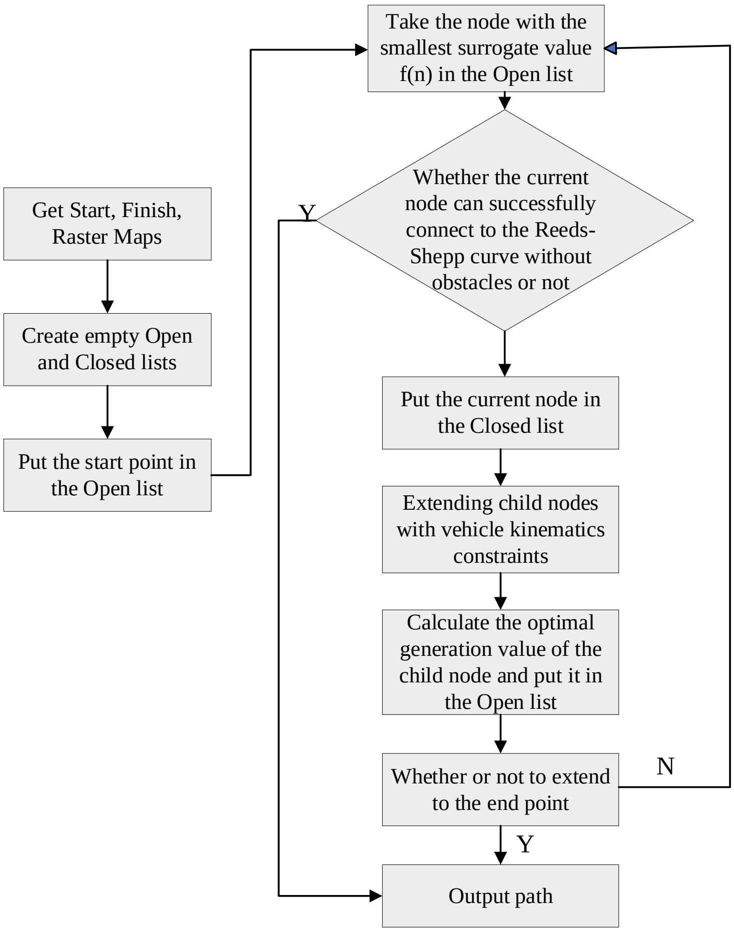


Fig. 1 - Flowchart of Hybrid A* algorithm

Improvements to the Hybrid A* algorithm

In order to solve the problems of low search efficiency and high time consumption in the original Hybrid A* algorithm when applied to complex grain depot scenarios with narrow aisles and numerous obstacles, an improved Hybrid A* algorithm is proposed.

Firstly, a more effective design of the heuristic function and the actual cost function is implemented to enhance the accuracy and relevance of the path planning. Secondly, the KD-Tree algorithm (*Ram & Sinha, 2019*) is used for nearest neighbor and distance range search to compute the distance to the nearest obstacle. When multiple points have the same calculated distance and surrogate value, to ensure the smoothness and safety of the driving path, priority is given, based on the calculation results, to the point with a smaller angle relative to the current driving direction, or to the point that is farther away from the obstacle. At the same time, the calculated obstacle distance cost is fused into the node expansion process. Based on this distance, dynamic expansion with variable resolution and variable step size is applied, improving the efficiency of path searching. Finally, the Reeds-Shepp curve (*Reeds & Shepp, 1990*) is used to generate the final path, ensuring it meets vehicle motion constraints and is optimal for driving. Combined with the vehicle schematic shown in Fig. 2, the Hybrid A* algorithm is specifically improved in the following ways:

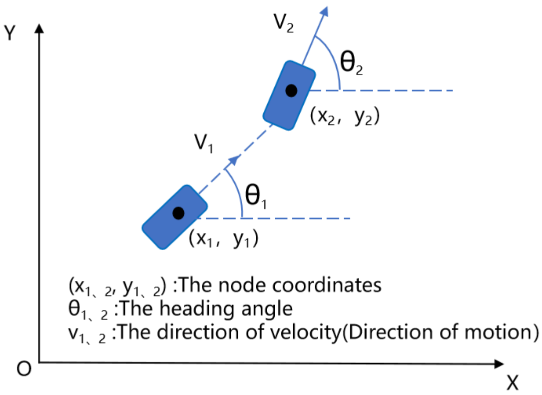


Fig. 2 - Schematic diagram of the vehicle

Heuristic function design

In Hybrid A* algorithm, the heuristic function plays a key role to estimate the heuristic generation value from the current node to the goal point to help the path search. To satisfy the vehicle kinematics and environmental constraints, two heuristic functions are used in parallel and the larger value of the two is selected as the heuristic search value as shown in equation (2).

$$h(n) = \max\{h_1(n), h_2(n)\} \tag{2}$$

In equation (2), the domain of definition of $h_1(n)$ and $h_2(n)$ is the set N of all nodes in the search space, where each node $n \in$ N contains the position coordinates *(x, y)* and the heading angle θ, and the value domain R is a nonnegative real number; where $h_1(n)$ is the first heuristic function that considers the kinematic constraints of the vehicle while ignoring the environmental obstacle constraints. When considering the kinematic constraints of the vehicle, it is necessary to incorporate factors such as minimum turning radius and speed into the heuristic function. This ensures that the planned path aligns with the vehicle's actual motion capabilities and avoids generating infeasible trajectories. Typically, this is achieved by using Reeds-Shepp curves to compute the shortest feasible path from the current node (x_n, y_n, θ_n) to the goal point (x_g, y_g, θ_g). The length of this path serves as the first heuristic function, denoted as $h_1(n)$. The second heuristic function, $h_2(n)$, takes into account environmental obstacle constraints while ignoring the vehicle's kinematic limitations. To enable obstacle avoidance, an obstacle distance field is introduced into the heuristic. The A* algorithm is used to evaluate the distance between the path and surrounding obstacles, where greater distances yield higher heuristic values. This steers the path away from obstacles, generating a safer route from the current node (x_n, y_n, θ_n) to the goal point (x_g, y_g, θ_g). The length of this path defines the value of the second heuristic function, $h_2(n)$. Here, *n* represents the current node, g is the goal point, *(x, y)* are the node coordinates, and θ is the heading angle.

$h_1(n)$ incorporates the feasible steering range of the Reeds-Shepp curve, helping to avoid dead ends where the vehicle cannot turn around. In contrast, $h_2(n)$ performs a search on the raster map using the A* algorithm, enforcing obstacle avoidance by utilizing obstacle grid markers. The combination of these two heuristics ensures that the planned path remains clear of hazardous areas. As a result, the final driving path not only complies with the vehicle's kinematic constraints but also effectively navigates around environmental obstacles.

Cost function design

In order to avoid unnecessary steering and reversing of the vehicle, it is necessary to ensure the continuity of the search path. In this paper, a penalty term is added for steering and reversing in the cost function to make the planned path more continuous.

The actual cost of each child node $n_i(x_i, y_i, \theta_i)$ in the search path of the Hybrid A* algorithm is shown in equation 3; where the child node n_i is a new node generated by expanding from the current node through vehicle kinematics constraints.

$$\begin{cases} g_i(n) = g_p(n) + g_{turn}(n) \times \omega_1 + g_{reverse}(n) \times \omega_2 \\ g_{turn}(n) = \left|\theta_i - \theta_p\right| \\ g_{reverse}(n) = L \end{cases} \tag{3}$$

In equation (3), $g_i(n)$ is the generation value of child node n_i, $g_p(n)$ is the generation value of parent node n_p, $g_{turn(n)}$ is the vehicle steering penalty coefficient, $g_{reverse(n)}$ is the vehicle reversing penalty coefficient, L is the length of the variable step length, θ is the heading angle, and ω_1, ω_2 are the weighting coefficients. Through several experiments, the parameters are adjusted according to the needs of vehicle driving in different scenarios, usually $\omega_1 > 1$ and $\omega_2 > 1$ (*Wilt & Ruml, 2012*).

In the designed cost function, $g_{turn(n)}$ represents the penalty associated with changes in the heading angle at a path point - that is, the angle between the vehicle's direction of movement and the X-axis of the global coordinate system. This change reflects a shift in the vehicle's actual trajectory. By applying a penalty for turning, the number of steering actions is reduced, thereby improving the smoothness of the planned path. The corresponding weight ω_1 can be flexibly adjusted based on the specific application scenario; for instance, if frequent steering is undesirable, a higher value of ω_1 can be used. Similarly, $g_{reverse(n)}$ represents the penalty for reverse driving. This is combined with the variable step length L to reduce the number of backward steps in the path, thereby encouraging forward motion. The weight ω_2 can also be flexibly adjusted; for example, if reversing is permitted but should be minimized, a higher value of ω_2 can be applied. By designing the cost function in this way, the planned path can be adapted to meet different driving requirements, balancing between path smoothness, forward motion preference, and scenario-specific constraints.

Node Extension Method Design

Although the original Hybrid A* algorithm satisfies vehicle kinematics and can generate extended trajectories based on these constraints, it expands nodes using a fixed resolution and fixed step size. This approach limits the diversity of path options, making it difficult to approximate optimal paths, reduces the flexibility of the search process, and results in high computational cost. Consequently, the node expansion process - where the current node generates a series of potential new nodes based on vehicle kinematics and the algorithm's search strategy - becomes time-consuming and inefficient. To address these issues, the node expansion method is improved by incorporating the distance $d_{nearest}(n)$ from the current node to the nearest obstacle, calculated using the KD-Tree algorithm. A dynamic node expansion strategy is proposed, which adjusts both the resolution and step size according to $d_{nearest}(n)$. This enhancement increases the flexibility of node expansion in the Hybrid A* algorithm and reduces extension time.

KD-Tree is a tree-based data structure designed for storing and efficiently retrieving points in k-dimensional space, enabling fast nearest neighbor searches (*Bi et al., 2022*). When using conventional methods to calculate the nearest distance (i.e., the minimum distance from the current node to an obstacle), it is necessary to compute the distance to each obstacle and compare the results, which is inefficient and yields poor performance. To address this, obstacles in the map are structured into a 2D KD-Tree, allowing both nearest neighbor and range searches between the current node and surrounding obstacles. By applying the KD-Tree algorithm within the 2D raster map, the nearest distance from an expanding node to nearby obstacles can be quickly determined, significantly improving the efficiency and safety of the path search.

In this context, obstacles refer to static objects that obstruct vehicle movement, and their positions in the 2D raster map are represented as coordinate points (i.e., obstacle coordinates).

The nearest distance between a node and an obstacle is calculated using the Euclidean distance formula: $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$ which represents the straight-line distance between two points. The implementation process is illustrated in the flowchart shown in Fig. 3.

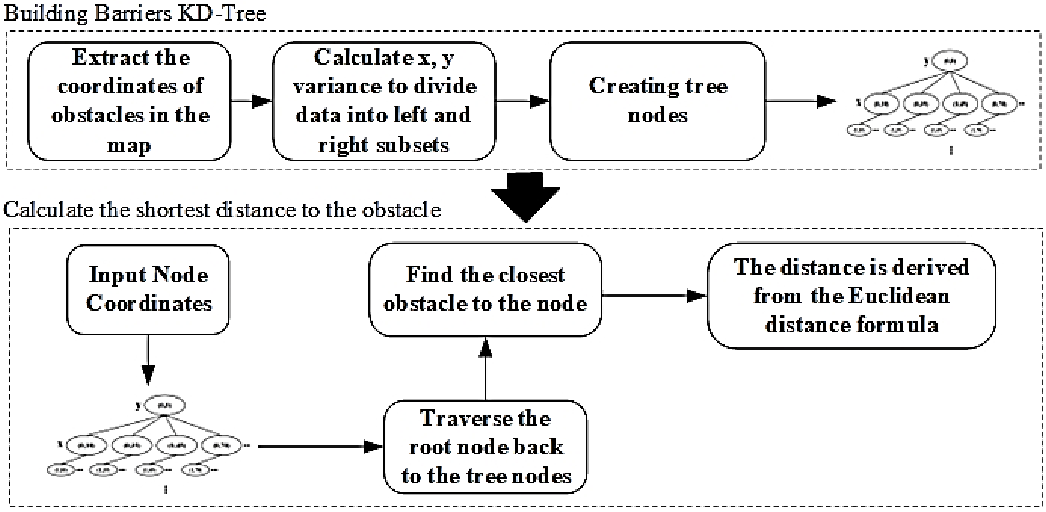


Fig. 3 - Flowchart of KD-Tree algorithm implementation

Based on the nearest distance $d_{nearest}(n)$ between the current vehicle node and the nearest obstacle, calculated using the previously constructed 2D KD-Tree of the obstacle data, the raster map can be globally partitioned into different regions according to d_{max} and d_{min}. Specifically, when $0 < d_{nearest}(n) < d_{min}$, the node is considered to be within a collision region, as illustrated in Fig. 4.

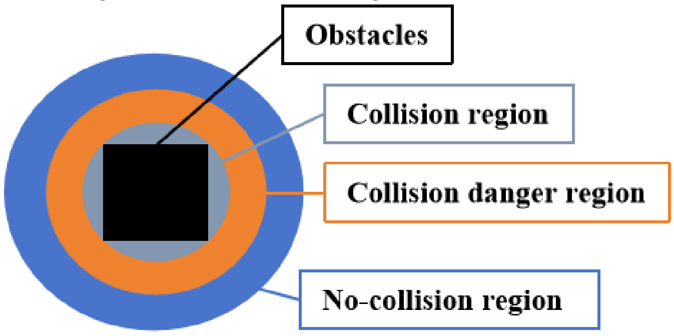


Fig. 4 – Schematic diagram of the region

The value of $d_{nearest}(n)$ is used to determine which region the vehicle occupies along the planned path. Simultaneously, the variable resolution K and variable step length L are linked to the nearest obstacle distance $d_{nearest}(n)$, enabling a dynamic adjustment of the node expansion strategy. As $d_{nearest}(n)$ changes, the search mode for node expansion is dynamically modified. Upper and lower bounds are defined for both the resolution and step length, and their values are adjusted in real time using Equations (4), (5), and (6). This adaptive approach allows the algorithm to select the most suitable node expansion mode based on the surrounding environment, significantly enhancing the efficiency of the path search.

$$K=\begin{cases} K_{\min} , & d_{max} \le d_{nearest}(n) \\ K_{\min}+a_1 \times b_1, & d_{\min} \le d_{nearest}(n) \le d_{\max} \\ K_{max} , & 0 \le d_{nearest}(n) \le d_{\min} \end{cases} \tag{4}$$

$$L=\begin{cases} L_{\max} , & d_{max} \le d_{nearest}(n) \\ L_{\min}+a_2 \times b_2, & d_{\min} \le d_{nearest}(n) \le d_{\max} \\ L_{\min} , & 0 \le d_{nearest}(n) \le d_{\min} \end{cases} \tag{5}$$

where the values of a_1, a_2, b_1, b_2 are shown in equation (6):

$$\begin{cases} a_1 = K_{max} - K_{\min} \\ a_2 = L_{max} - L_{\min} \\ b_1 = \dfrac{d_{max} - d_{nearest}(n)}{d_{max} - d_{\min}} \\ b_2 = \dfrac{d_{nearest}(n) - d_{\min}}{d_{max} - d_{\min}} \end{cases} \tag{6}$$

In the above, d_{min} is the minimum safe distance from the obstacle and d_{max} is the maximum safe distance from the obstacle. In equation (5), K_{max} is the maximum discretization level of the variable resolution during node expansion and K_{min} is the minimum discretization level. Similarly, in Equation (6), L_{max} and L_{min} represent the maximum and minimum values, respectively, of the variable step length used during node expansion.

Therefore, after obtaining the vehicle position in the path, the algorithm determines whether the vehicle collides with the obstacle through $d_{nearest}(n)$. If in the collision-free region, the algorithm uses the resolution of the minimum discretization and the step size of the maximum length for node extension, and the resolution and step size of the node extension process if the obstacle is encountered in the search path will be changed with the change of the nearest distance $d_{nearest}(n)$. Fig. 5 shows the schematic diagram of dynamic extension, in which black squares are obstacles, green circles are collision-free regions, and red circles are dangerous collision regions. Fig. 6 shows the comparison diagram before and after the improved node extension, in which the blue box is the starting point of the vehicle, the red box is the end point of the path, the black square is the obstacle, the white grid part is the passable area, and the red track is the node trajectory. This improved node extension method can reduce the collision probability and improve the path accuracy by increasing the resolution discretization of the search path and decreasing the step length in the narrow channel and complex environment of the granary; in the normal environment of the granary, it can reduce the number of nodes to be searched and accelerate the search speed by decreasing the resolution discretization of the search path and increasing the step length. This not only improves the search efficiency of the Hybrid A* algorithm, but also ensures that the vehicles search over long distances and narrow aisles, further guaranteeing the safety of the path.

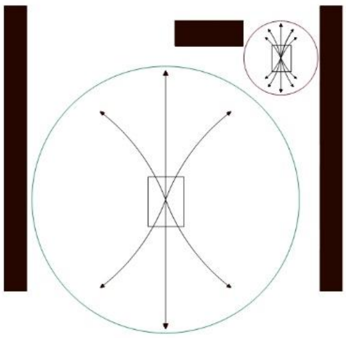
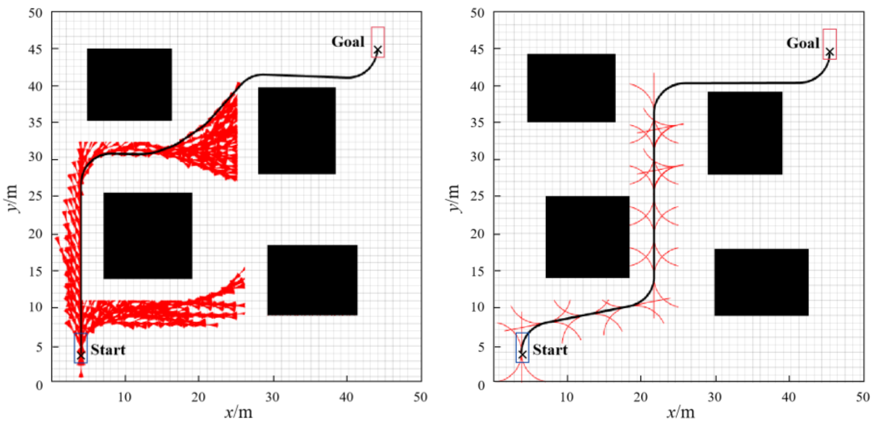


Fig. 5 - Schematic diagram of dynamic node extension



(a) Dynamic node extensions not used (b) Using dynamic node extensions

Fig. 6 - Comparison of planning before and after improving the way of extension nodes

Table 1

Comparison of dynamic node extension data

Title	Dynamic node extensions not used	Using dynamic node extensions
Number of nodes/piece	843	49
Search time/s	1.95	0.62

From the comparison data in Table 1, it can be concluded that the use of dynamic extension nodes reduces the number of search nodes by 94% and the search time by 68% compared to not using dynamic extension. The path search efficiency of the improved Hybrid A* algorithm can be improved by using dynamic extension nodes.

Search Path Fitting

It may not be possible to search the exact target point pose by the improved node extension approach, and in this paper, the Reeds-Shepp curve, which takes into account the vehicle kinematics constraints, is used for the final path search fitting. As shown in Fig.7, the blue box is the starting point, the red box is the end point, the black square is the obstacle, and the red dashed line is the Reeds-Shepp search curve. When the Hybrid A* algorithm is extended to the current node and the target point, it will use the Reeds-Shepp curve for path search; if there is no collision with obstacles during the path fitting process, the fitted path is a valid path to stop searching; otherwise, it continues to use the Hybrid A* algorithm to conduct path searching until a drivable path is obtained.

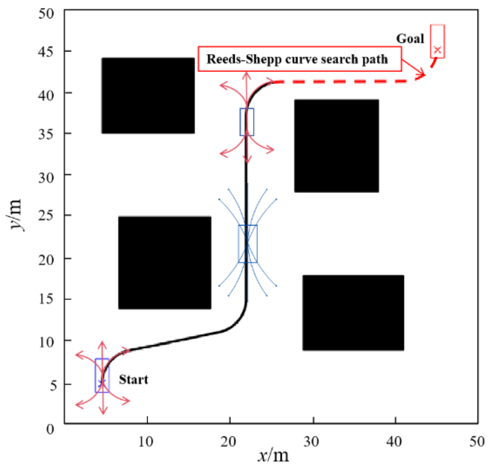
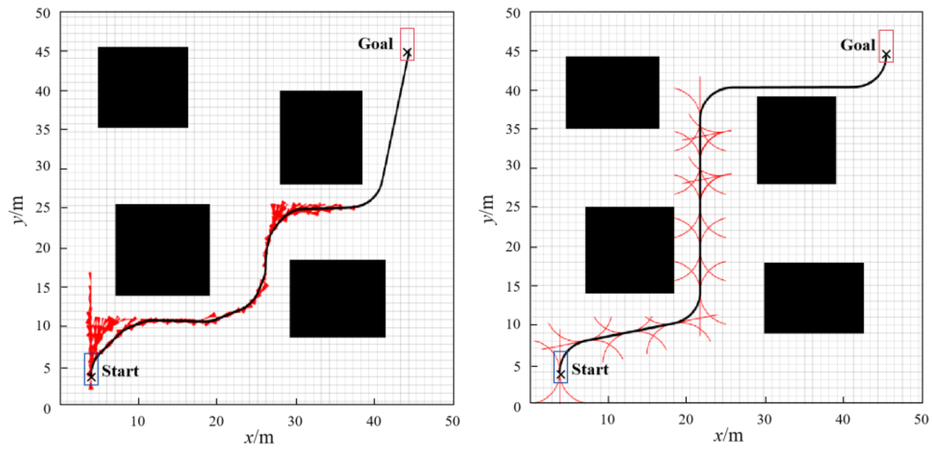


Fig. 7 - Schematic diagram of Reeds-Shepp curve path search fitting

RESULTS

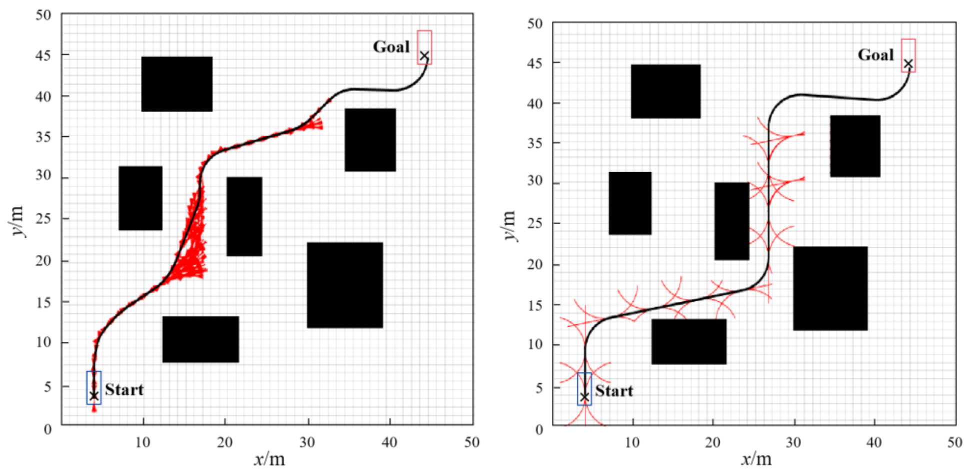
Algorithm Testing

To verify the effectiveness and feasibility of the proposed algorithmic improvements, two types of obstacle distribution maps, simple and complex, were carefully designed within environment maps featuring diverse topologies. These maps included narrow passages and multi-obstacle layouts typical of grain depot scenarios. Based on these two simulated grain depot environments, the performance of the original Hybrid A* algorithm and the improved version proposed in this study was evaluated. The comparison focused on two key indicators: search time and number of search nodes, allowing for a comprehensive analysis of algorithmic performance. The experiment was conducted using MATLAB R2020a. The simulated vehicle parameters are as follows: a length of 4.0 m, a width of 1.6 m, a wheelbase of 2.75 m, and front and rear overhangs of 1.0 m each. The maximum turning angle of the front wheels is 0.6 rad. The simulation environment is a 50 m × 50 m raster map. The vehicle's starting position and orientation, as well as the distribution of obstacles, are illustrated in Fig. 8 and Fig. 9. In the figures, the blue box indicates the vehicle's starting position, the red box marks the target position, black squares represent obstacles, white grids denote traversable areas, and the red line shows the trajectory through the path nodes. According to the obstacle layout, Fig. 8 represents a simple map with a sparse and regular distribution of obstacles, while Fig. 9 depicts a complex map characterized by a high density and irregular distribution of obstacles. The simulations were performed on a Windows 10 platform with an Intel i5-1135G7 processor and 16 GB of RAM. MATLAB R2020a was used to simulate the proposed algorithm in the context of two grain depot environments. In both scenarios, the starting and target positions remained fixed, and the obstacle configuration was unchanged. Under these conditions, the generated path trajectories were largely consistent across multiple runs. Performance metrics such as the number of nodes, search time, and path length were evaluated based on the average results from more than ten simulation trials.



(a) Original Hybrid A* algorithm (b) Improved Hybrid A* algorithm

Fig. 8 - Comparison of simple map path planning



(a) Original Hybrid A* algorithm (b) Improved Hybrid A* algorithm

Fig. 9 - Comparison of complex map path planning

Table 2

Simple map simulation test data comparison

Algorithm	Number of nodes/piece	Search time/s	Path length/m
Original Hybrid A*	301	0.99	66.2
Improved Hybrid A*	49	0.62	71.4

Table 3

Complex map simulation test data comparison

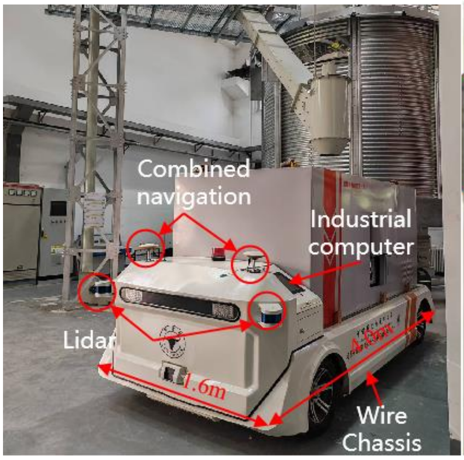
Algorithm	Number of nodes/piece	Search time/s	Path length/m
Original Hybrid A*	191	1.66	64.1
Improved Hybrid A*	37	0.72	71.5

As shown in Table 2 and Fig. 8, in the setup of simple grain depot map, the improved Hybrid A* algorithm reduces by 83% and 37% in the number of search nodes and path search time, respectively, compared to the original Hybrid A* algorithm. As shown in Table 3 and Fig. 9, the improved Hybrid A* algorithm reduces the number of search nodes and path search time by 80.6% and 56.6%, respectively, compared with the original Hybrid A* algorithm in the setup of complex grain depot maps. The simulation results show that the improved Hybrid A* algorithm has a significant improvement in the number of search nodes and the search path time compared to the original Hybrid A* algorithm in the setup of two kinds of maps of grain depot environments, but in terms of path length, the improved algorithm takes vehicle kinematics constraints and collision avoidance into consideration when planning paths and chooses paths more in line with the driving requirements, so there is no substantial enhancement. The efficiency order of the original and improved algorithms may be different under different grain depot obstacle topologies.

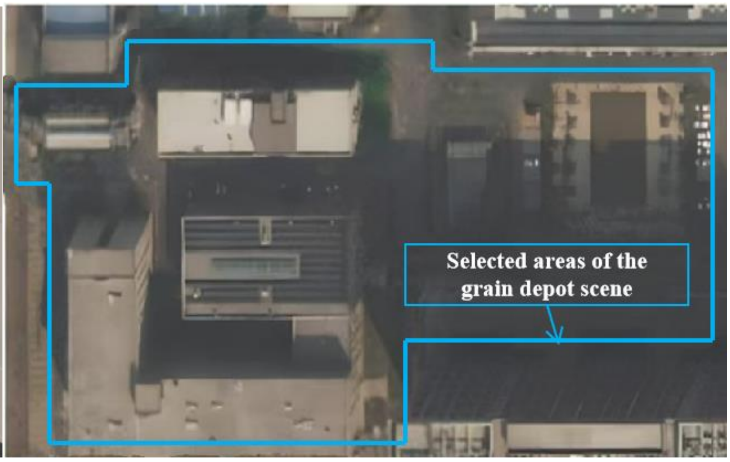
When the distribution of obstacles is sparse and the shape is regular, the performance of the two algorithms may be similar; however, when the distribution of obstacles is dense and the shape is complex, the improved algorithm may show higher efficiency.

Real-vehicle Verification

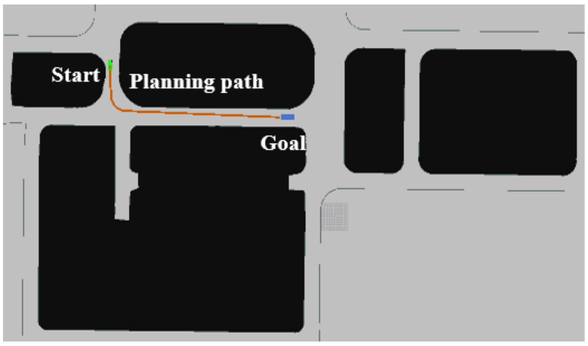
In order to further verify the effectiveness and feasibility of the improved Hybrid A* algorithm, the test vehicle used in this paper is an intelligent grain transfer vehicle conforming to the Ackermann steering model (*Ren et al., 2009*), with a vehicle length of 4.0 m, width of 1.6 m, height of 2.2 m, wheelbase of 2.75 m, a minimum turning radius of 5.5 m, and a maximum turning angle of 0.54 rad at the front wheels, and is equipped with 16-wire LIDAR. RTK inertial guidance, millimeter wave radar and other sensors are shown in Fig.10 (a). Tested using the ROS operating system under Ubuntu 18.04, the original planning algorithm in the ROS system was replaced with the improved Hybrid A* algorithm and the original Hybrid A* in the form of the ROS plugin, and then experimental testing was carried out. The map used for navigation is constructed by RTK inertial guidance acquisition of road boundary points, and is processed into the raster map needed in the ROS system through image processing. The experimental results are shown in Fig. 10, the blue line area in (b) is a selected part of the grain depot scene, the black part in (c) and (d) is the obstacle, the green square is the starting point, the blue square is the end point, and the yellow line is the planned trajectory. As can be seen in Fig. 9, both Hybrid A* algorithms can plan a drivable path. In terms of search time, the original Hybrid A* algorithm takes 25.34s and the improved Hybrid A* algorithm takes 9.76s, and the improved algorithm improves 61.4% compared to the original one. In terms of the number of search nodes, the original algorithm searches for 34 nodes and the improved algorithm searches for 16 nodes, which is 52.9% higher compared to the original algorithm. In terms of the path length, the original algorithm and the improved algorithm plan a length of 80.85m and 80.73m, respectively, with no significant improvement. The results show that the improved Hybrid A* algorithm proposed in this paper is effective in improving the path search efficiency and safety in the presence of narrow aisles and the complex environment of the grain depot scenario.



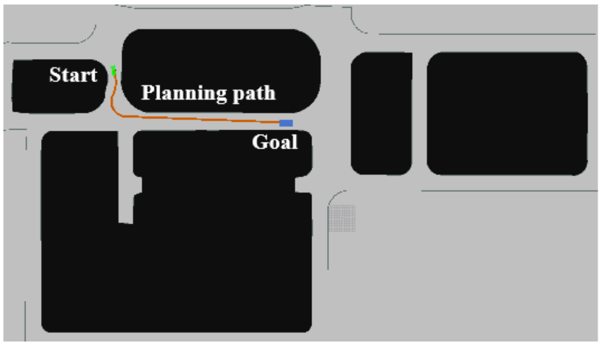
(a) Test vehicle



(b) Test scenarios



(c) Original Hybrid A* algorithm planning results



(d) Improved Hybrid A* algorithm planning results

Fig. 10 - Real vehicle validation results

CONCLUSIONS

In this paper, a Hybrid A* algorithm based on variable resolution and variable step size is proposed for the deficiencies of the original Hybrid A* algorithm in the grain depot scenario. Firstly, by introducing weight coefficients and penalty terms, the heuristic function and cost function are reasonably designed to reduce the number of times that the vehicle turns and reverses in path planning. Secondly, combining KD-Tree to calculate the distance cost between nodes and obstacles, the resolution and step size are dynamically adjusted to realize the flexibility of node extension. Finally, in order to accurately locate to the target position, Reeds-Shepp curves are used for path search fitting to ensure the accuracy of the planned path. Simulation results show that the improved Hybrid A* algorithm reduces 83% and 37% in simple maps compared to the original Hybrid A* algorithm in terms of the number of search nodes and planning elapsed time, respectively. In complex maps, the reduction is 80.6% and 56.6%, respectively. The results of the real-vehicle experiments show that the improved Hybrid A* algorithm reduces the number of search nodes and planning time by 52.9% and 61.4%, respectively, compared with the original Hybrid A* algorithm. In summary, the improved Hybrid A* algorithm proposed in this paper significantly improves the path search efficiency of the intelligent grain transporter in the grain depot scenario by dynamically adjusting the resolution and step size, combining the KD-Tree distance cost with the Reeds-Shepp curve search fitting, which further validates the feasibility of the improved algorithm. Future research topics may include improving the adaptability of the algorithm for real-time planning in dynamic obstacle environments as well as conducting experiments in obstacles with different topologies to expand the scale and diversity of the experiments, while considering vehicle travel time and energy consumption to enhance the effectiveness of the improved algorithm. At present, although the improved algorithm has been experimentally validated, its in-vehicle validation needs further improvement. At this stage, the research focuses on being able to improve the search efficiency when planning paths in grain depot scenarios, and subsequent research will be further expanded on this basis.

ACKNOWLEDGEMENT

The authors thank the anonymous reviewers for their constructive comments. This study was partially supported by the National Key Research and Development Program of China [Grant No. 2022YFD2100201], the Key Research, Development and Promotion Project of Henan Province [Grant No. 231111241100], Supported by Cultivation Project of Tuoxin Team in Henan University of Technology [Grant No. 2024TXTD16], and the Science and Technology Tackling Project of Henan Province [Grant No. 252102241052].

REFERENCES

[1] Bi, W. H., Ma, J. W., Zhu, X. D., Wang, W. X., & Zhang, A. (2022). Cloud service selection based on weighted KD tree nearest neighbor search. *Applied Soft Computing,* Netherlands, 131, 109780.

[2] Cao, Y., Yan, J., Li, X., & Cao, L. (2023). A Method of Path Search for Automatic Parking System Based on Improved Hybrid A*Algorithm (基于改进混合 A*算法的自动泊车系统路径搜索方法), *Automobile Technology*, Changchun/China, 6(06), 37-41. https://doi.org/10.19620/j.cnki.1000-3703.20230268

[3] Cao, L., (2023). *Research on Automatic Parking Path Planning Algorithm Based on Hybrid A** (基于Hybrid A*的自动泊车路径规划算法研究), MSc dissertation, Dalian University of Technology, Dalian/China.

[4] Dijkstra, E. W. (2022). A note on two problems in connexion with graphs. *Numerische Mathematik,* Germany, 1(1), 287-290. https://doi.org/10.1007/BF01386390

[5] Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2010). Path Planning for Autonomous Vehicles inUnknown Semi-structured Environments. *International Journal of Robotics Research,* England, *29*(5), 485-501. https://doi.org/10.1177/0278364909359210

[6] Deng, M., Lin, Y., Huang, J., & Luo, Y. (2022). Research on Hybrid A*Based Path Optimization of Unmanned Mine Truck (基于混合 A*算法的无人驾驶矿用卡车路径优化研究), *Control and Information Technology*, Zhuzhou/China, 5(05), 60-67. https://doi.org/10.13889/j.issn.2096

[7] Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, USA, 22(6), 46-57. https://doi.org/10.1109/2.30720

[8] He, Z. B., Liu, C. G., Chu, X. M., Negenborn, R. R., & Wu, Q. (2022). Dynamic anti-collision A-star algorithm for multi-ship encounter situations. *Applied Ocean Research*, Netherlands, 118, 102995. https://doi.org/10.1016/j.apor.2021.102995

[9] Huang, Y. Z., Wang, H., Han, L., & Xu, Y. Q. (2024). Robot path planning in narrow passages based on improved PRM method. *Intelligent Service Robotics*, Germany 17(3), 609-620.

[10] LI, S. (2024). Multi-UAV Task Allocation and Path Planning Method for Agricultural Patrol Scenes. *INMATEH-Agricultural Engineering*, Romania, 74, 582-591. https://doi.org/10.35633/inmateh-74-52
[11] Nayak, M. K., Daglish, G. J., Phillips, T. W., & Ebert, P. R. (2020). Resistance to the Fumigant Phosphine and Its Management in Insect Pests of Stored Products: A Global Perspective. In A. E. Douglas (Ed.), *Intelligent Service Robotics*, Germany, 65, 333-350. https://doi.org/10.1146/annurev-ento-011019-025047
[12] Peng, Y., Jiang, M., Ma, Z., & Zhong C. (2021). Review on development of key technology for autonomous vehicle (汽车自动驾驶关键技术研究进展), *Journal of Fuzhou University(Natural Science Edition)*, Fuzhou/China, 49(05), 691-703. https://doi.org/10.7631/issn.1000-2243.21272
[13] Qin, D., Zhang, W., Wang, T., & Chen, J. (2024). A Path Planning Method for Parallel Parking in Constrained Parking Channels (受限泊车通道下自动驾驶平行泊车的路径规划方法), *Journal of Zhengzhou University(Engineering Science),* Zhengzhou/Chian*, 45*(05), 1-7.
[14] Ren, X., & Cai, Z. (2009). Using the Ackerman principle for kinematic modeling of wheeled mobile robots (基于阿克曼原理的车式移动机器人运动学建模), *CAAI Transactions on Intelligent Systems*, Haerbin/China, 4(06), 534-537. https://doi.org/10.3969/j.issn.1673-4785.2009.06.011
[15] Reeds, J. A., & Shepp, L. A. (1990). Optimal Paths for a Car That Goes Both Forwards and Backwards. *Pacific Journal of Mathematics*, USA, 145, 367-393. https://doi.org/10.2140/PJM.1990.145.367
[16] Ren, B., Wang, X., Deng, W., & Nan, J. (2022). Path Optimization Algorithm for Automatic Parking Based on Hybrid A* and Reeds-Shepp Curve with Variable Radius (基于混合 A*和可变半径 RS 曲线的自动泊车路径优化方法), *China Journal of Highway and Transport*, Xian/China, 35(07), 317-327.
[17] Ram, P., & Sinha, K. (2019). Revisiting kd-tree for nearest neighbor search. *Paper presented at the Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, USA, 1378-1388. https://doi.org/10.1145/3292500.3330875
[18] Sedighi, S., Nguyen, D.-V., Kapsalas, P., & Kuhnert, K.-D. (2019). Implementing Voronoi-based Guided Hybrid A* in Global Path Planning for Autonomous Vehicles. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland/New Zealand, 3845-3851. https://doi.org/10.1109/ITSC.2019.8917427
[19] Tang, F., & Li, Y. (2024). Current Status and Recommendations for the Development of Smart Grain Depot Construction in China (中国智慧粮库建设发展现状及建议), *Science and Technology of Cereals, Oils and Foods,* Beijing/China*, 32*(05), 35-41+257. https://doi.org/10.16210/j.cnki.1007-7561.2024.05.005
[20] Teng, S. Y., Hu, X. M., Deng, P., Li, B., Li, Y. C., Ai, Y. F., . . . Chen, L. (2023). Motion Planning for Autonomous Driving: The State of the Art and Future Perspectives. *IEEE Transactions on Intelligent Vehicles*, USA, 8(6), 3692-3711. https://doi.org/10.1109/TIV.2023.3274536
[21] Tian, Y., Lin, S., Fang, D., & Jiang, J. (2023). Path Planning of Robot with Nonholonomic Constraint based on Improved Hybrid A* algorithm (基于改进 HybridA*的非完整约束机器人路径规划算法研究), *Machinery*, Chengdu/China, 50(03), 63-71. https://doi.org/10.3969/j.issn.1006-0316.2023.03.010
[22] Tang, B., Hirota, K., Wu, X., Dai, Y., & Jia, Z. (2021). Path Planning Based on Improved Hybrid A* Algorithm. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Japan, 25(1), 64-72. https://doi.org/10.20965/jaciii.2021.p0064
[23] Wu, L., Huang, X. D., Cui, J. G., Liu, C., & Xiao, W. S. (2023). Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot. *Expert Systems with Applications*, England, 215, 119410. https://doi.org/10.1016/j.eswa.2022.119410
[24] Wilt, C., & Ruml, W. (2012). When does weighted A* fail. *Paper presented at the Proceedings of the International Symposium on Combinatorial Search,* USA, 3(1),137-144. https://doi.org/10.1609/socs.v3i1.18250
[25] Xiong, L., Yang, X., Zhuo, G., Leng, B., & Zhang, R. (2020). Review on Motion Control of Autonomous Vehicles (无人驾驶车辆的运动控制发展现状综述), *Journal of Mechanical Engineering,* Beijing/China, 56(10), 127-143.https://doi.org/10.3901/JME.2020.10.127
[26] Yin, J. H., Hu, Z., Mourelatos, Z. P., Gorsich, D., Singh, A., & Tau, S. (2023). Efficient Reliability-Based Path Planning of Off-Road Autonomous Ground Vehicles Through the Coupling of Surrogate Modeling and RRT. *IEEE Transactions on Intelligent Transportation Systems,* USA, 24(12), 15035-15050.
[27] Zhang, K. (2024). Obstacle Avoidance Path of Wheeled Agricultural Handling Robots in Warehouse Based on Improved ACO-DWA Algorithm. *INMATEH-Agricultural Engineering,* Romania, 74, 571-581. https://doi.org/10.35633/inmateh-74-51