Improved Guided A-Star Path Planning Method Based on APF Algorithm

Junjie Yang[1]*

[1]School of Electrical and Electronic Engineering, Shangqiu Normal University, Henan Province, 476000, China

* yangjunjie@sqnu.edu.cn

Abstract. Path planning is a core challenge in robot autonomous navigation. Traditional artificial potential field methods have problems such as local minimum traps, while the A* algorithm, although capable of generating globally highest-quality trajectorys, cannot handle dynamic environments. The separation of these methods restricts the application efficiency of robots in real-time scenarios. This research introduces an enhanced APF algorithm, guided by A* macro-level navigation planning. By using the node sequence planned by A* as the anchor points for the potential field direction and reconstructing the repulsive field boundary range, this attempt aims to address the local minima issue inherent in traditional APF algorithms and improve their dynamic obstacle avoidance capabilities. The result shows this integrated framework enables robots to inherit the reliable macro-level navigation planning capability of the A* algorithm while leveraging the real-time obstacle avoidance agility of APF. The improved A* algorithm optimizes path length by 5.4% and reduces turns by 33.3% on average. Meanwhile, the improved APF algorithm shows significant advantages in dynamic obstacle avoidance. It has faster response and larger safe distance than traditional APF. For example, in extreme speed combination (0.125:1.25), traditional APF fails while improved APF responds in 0.125 s with 2.313 m safe distance.

Keywords: Path planning, Dynamic obstacle avoidance, Artificial potential field method

1 Introduction

Path planning represents a critical issue in robotics and autonomous navigation systems, designed to guide robots from their initial position to the target location while circumventing obstacles in the path. Among the various algorithms developed for this purpose, the A* method and the artificial potential field approach are widely recognized for their unique advantages [1,2]. The A* algorithm is efficient and easy to implement in global static path planning, using a heuristic approach to efficiently explore the search space and identify the highest-quality trajectory. However, the A* algorithm lacks dynamic obstacle avoidance capabilities, which are essential in environments where obstacles may move or change. In contrast, the artificial potential field approach

J. Iqbal (ed.), *Proceedings of the 2025 2nd International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2025)*, Advances in Engineering Research 272,
https://doi.org/10.2991/978-94-6463-821-9_108

excels in dealing with fluctuating surroundings by generating forces that attract and repel the robot, enabling real-time navigation. Although the APF algorithm can quickly respond to new obstacles and adjust the path, it may encounter problems such as local minima, while effective in many scenarios, has notable limitations. Robots employing this method may become trapped or experience oscillations, preventing them from advancing toward the target. These issues can significantly impact the method's utility in intricate or complex settings.

Various algorithms have been developed to improve on the above problems. Gao et al. proposed a hybrid path planning method based on the improved A* and CSA-APF algorithms, aiming to enhance the efficiency and reliability of path planning by combining the advantages of both algorithms [3]. Sang et al. studied a hybrid trajectory planning algorithm based on the improved A* and artificial potential field for the path planning of unmanned surface vessel formations, with the goal of improving their obstacle avoidance capabilities and path optimization effects in complex environments [4]. Liang et al. achieved autonomous collision avoidance for unmanned surface vessels by combining the A* algorithm with the minimum heading change algorithm [5]. Seo et al. proposed an A* algorithm based on CRI for ship collision avoidance path planning [6]. Moradi et al. optimized maritime routes through reinforcement learning methods [7].

The paper presents an innovative approach that optimizes the global path generated by the A* algorithm through the removal of redundant nodes, while maintaining the nodes selected as target points by the APF method. This approach integrates the efficient planning of the A* algorithm with the dynamic obstacle evasion advantage of the APF algorithm. By taking the global path as a reference, the path planned by the APF method can effectively avoid static and dynamic obstacles while maintaining a consistent trajectory with the overall goal. Additionally, this paper also enhances the repulsive force function of the APF method, allowing the range and intensity of the repulsive field to adjust based on the velocity and direction of moving obstacles. This improvement boosts the hybrid algorithm's ability to avoid moving obstacles dynamically. The combination of these two methods not only strengthens the system's dynamic obstacle avoidance capability but also avoids the problems related to local minima in the APF method. This study will conduct simulations to compare and analyze the adaptability and robustness of the collaborative path planning method that merges the simplified A* algorithm and the improved APF algorithm in both static and dynamic environments. Finally, this research aims to contribute to the development of effective autonomous navigation strategies in complex and dynamic environments and provide ideas for performance improvement in real-world applications. By leveraging the advantages of these two algorithms, this paper hopes to create a more effective and robust path planning solution capable of effectively addressing the challenges posed by dynamic environments. Local path planning and dynamic obstacle avoidance based on the APF algorithm.

2 Improved APF

2.1 Traditional APF Algorithm

In the APF method, the gravitational potential field function and the attractive force are correspondingly defined as Equations (1) and (2).

$$U_{\text{att}}(q) = -\frac{1}{2}\xi\rho^2(q, q_{\text{goal}}) \qquad (1)$$

$$F_{\text{att}}(q) = -\nabla U_{\text{att}}(q) \qquad (2)$$

In this context, the coefficient factor of the attractive is represented by ξ. The vector $\rho^2(q, q_{\text{goal}})$ extends between the robot's present position q and the target q_{goal}, with its length is equivalent to the Euclidean distance between these two points. The associated attractive force $F_{\text{att}}(q)$ is determined by the opposite direction of the gradient of the potential field.

In the APF method, the repulsive potential field function and the repulsive force are correspondingly defined as Equations (3) and (4):

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q, q_{\text{obs}})} - \frac{1}{\rho_0}\right)^2, & \rho(q, q_{\text{obs}}) < \rho_0 \\ 0, & \rho(q, q_{\text{obs}}) > \rho_0 \end{cases} \qquad (3)$$

$$F_{\text{rep}}(q) = -\nabla U_{\text{rep}}(q) \qquad (4)$$

Where is the repulsive potential field gain coefficient, $\rho(q, q_{\text{obs}})$ is a vector pointing between the obstacle position and the robot's present position, with a magnitude equal to the straight-line distance between the robot's present position q and the obstacle position q_{obs}. The corresponding repulsive force $F_{\text{rep}}(q)$ is the negative gradient of the repulsive field, and ρ_0 is a constant representing the maximum influence range of the obstacle on the vehicle.

2.2 Improved APF Algorithm with Extended Potential Field Range

In the traditional artificial potential field method, the potential field range ρ_0 is set as a fixed constant, which means that in a two-dimensional plane, The repulsive potential field, produced by an obstacle, extends over a circular area with the obstacle's material point as the center and ρ_0 as the radius. However, this approach does not distinguish between dynamic and static obstacles, thus having certain limitations when dealing with dynamic obstacles.

This paper proposes a new method for calculating the potential field ρ_0. This method incorporates the obstacle's material point into the polar equation of an ellipse, thereby enabling the adjustment of the potential field's range based on the movement characteristics of dynamic obstacles, achieving early avoidance of dynamic obstacles. The specific calculation formula is as follows (5):

$$\rho_0 = \frac{a(1-c^2)}{1-e\cos(\theta)} \tag{5}$$

Here, ρ_0 denotes the polar radius, which is the potential field range of the dynamic obstacle; θ is the polar angle, that is, the angle between the vector $\rho(q, q_{obs})$ pointing from the dynamic obstacle's position to the robot's present position and the velocity vector $\vec{v}$; e is the eccentricity of the ellipse. Additionally, the definitions of parameters c and b are as follows (6):

$$\begin{cases} c = k_1 |\vec{v}| \\ b = k_2 \eta \end{cases} \tag{6}$$

By adjusting the coefficients k_1 and k_2, the shape of the ellipse can be changed, thereby altering the range of the potential field, enabling the robot to make evasive actions in advance based on the speed and direction of the dynamic obstacles. When the speed of the dynamic obstacle $\vec{v}$, is 0 that is, when the obstacle is stationary, the range of the repulsive potential field ρ_0, will degenerate into a constant, consistent with the way static obstacles are handled in traditional potential field methods. The complete repulsive force calculation method after improvement is as follows (7):

The enhanced potential field technique not only demonstrates superior adaptability to obstacles in dynamic settings but also enhances the flexibility and safety of robotic path planning. By appropriately selecting the values of k_1 and k_2, the shape of the ellipse and the range of the potential field can be optimized in line with specific operational settings and robotic performance criteria, leading to a more efficient obstacle avoidance strategy.

$$\rho_0 = \frac{a(1-c^2)}{1-e\cos(\theta)}$$
$$\begin{cases} c = k_1 |\vec{v}| \\ b = k_2 \eta \end{cases}$$
$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q,q_{\text{obs}})} - \frac{1}{\rho_0}\right)^2, & \rho(q,q_{\text{obs}}) < \rho_0 \\ 0, & \rho(q,q_{\text{obs}}) > \rho_0 \end{cases}$$
$$F_{\text{rep}}(q) = -\nabla U_{\text{rep}}(q) \tag{7}$$

3 Static Global Path Planning Based on A* Algorithm

3.1 A* Algorithm

The A* algorithm is a classic path search algorithm [8,9]. The basic principle of the algorithm is to designate the initial point as the first parent node, evaluate the $f(n)$ values of the eight surrounding child nodes, and select as the new parent node the one with the minimum $f(n)$ value. The cycle continues in an iterative manner until the endpoint is ultimately achieved. After the search is completed, the final planned path with the lowest cost is obtained by backtracking from the endpoint to the initial point along the parent nodes. The conventional A* algorithm utilizes an assessment function defined as follows:

$$f(n) = g(n) + h(n) \tag{8}$$

In the formula, $\mathrm{g(n)}$ denotes the real cost value between the initial node and the present node n, and $\mathrm{h(n)}$ is the heuristic function, indicating the predict cost value between the present node n and the destination. There are various methods to calculate the heuristic function $\mathrm{h(n)}$, such as Euclidean distance, Manhattan distance, Chebyshev distance, and Mahalanobis distance, etc [10]. In this paper, Euclidean distance is adopted for representation:

$$h(n)=\sqrt{(x_{\text{goal}}-x_{\text{now}})^2+(y_{\text{goal}}-y_{\text{now}})^2} \quad (9)$$

In the formula: x_{goal} denotes the abscissa of the destination, x_{now} epresents the ordinate of the destination, y_{goal} denotes the ordinate of the present node, and y_{now} denotes the ordinate of the present node node.

3.2 Optimized A* Path by Removing Unnecessary Nodes

The paths planned by the A* algorithm often contain many redundant nodes, resulting in many turns and low efficiency. This paper adopts a post-processing step, which uses collision detection to determine whether there are collision obstacles between two nodes. If not, the subsequent node is merged into the previous node until a collision occurs, then the farthest reachable index is updated, and this index is taken as the new starting node to repeat this process before arriving at the end point. Thus, the path length is shortened, the number of turning points is reduced, and the efficiency of the path is improved. As show on Fig. 1.

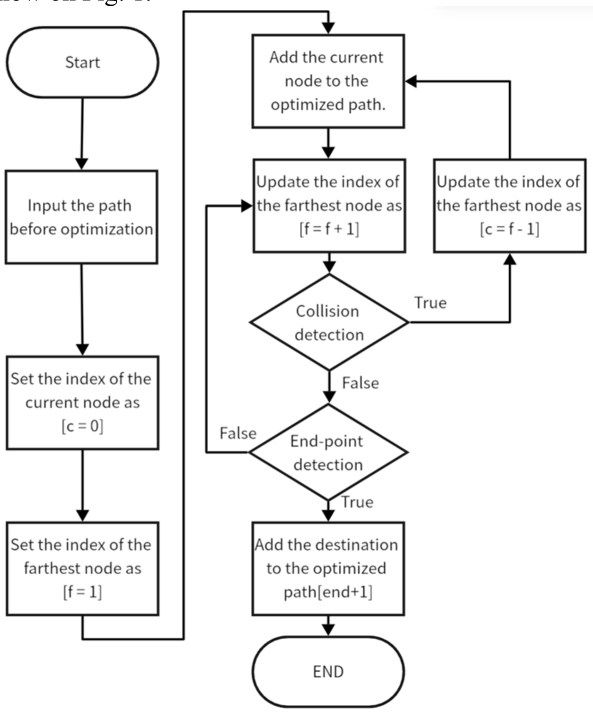

Fig. 1. Flowchart of Path Optimization

4 Path Planning Combining A* and APF

As a classic trajectory planning algorithm, the A* algorithm, after the above optimizations, can effectively decrease the quantity of unnecessary turning points in the path and shorten the path length, thereby planning a globally highest-quality trajectory in known surroundings. It evaluates and sorts the nodes in the open list according to a specific heuristic function, prioritizing the expansion of those nodes that are more likely to approach the target, thereby improving the search efficiency and path quality. This global optimality makes the A* algorithm highly reliable and accurate in path planning, capable of providing a robot with a highest-quality trajectory from the starting point to the destination.

Meanwhile, the APF method has also demonstrated its unique advantages in the above-mentioned modifications, effectively solving the obstacle avoidance problem in dynamic environments. By introducing the elliptical polar coordinate equation to calculate the potential field range ρ_0 of dynamic obstacles, the robot can make evasive actions in advance in accordance with the direction and speed of the obstacle's movement. When the obstacle is stationary, this method can degenerate into the traditional static obstacle handling approach, ensuring its universality and flexibility. Due to the real-time obstacle avoidance capability of the APF method, it can quickly respond to the movement changes of obstacles in dynamic environments, timely adjust the robot's path planning, and prevent collisions from occurring.

This paper combines these two improved algorithms to fully leverage their respective advantages. The A* is used for macro-level navigation planning in a known obstacle environment. By analyzing and processing map information, it generates an optimized path list. The key points in this path list serve as the segmented gravitational anchor points for the navigation planning of the APF method, providing clear goals and directions for local planning. In the local planning stage, the APF method plans along the optimized path list, taking advantage of its real-time capabilities to adjust the movement direction and speed of robot in real time based on the dynamic obstacle conditions in the environment, guaranteeing the robot safely and efficiently achieves the destination. As show on Fig. 2.

This combination approach not only fully exploits the global optimality of the A* algorithm, ensuring the overall direction and optimality of the entire path planning, but also leverages the real-time nature of the artificial potential field method, enabling the robot to flexibly respond to various unexpected situations in dynamic environments. By integrating global planning with local planning, it not only avoids the limitations of pure global planning in dynamic environments but also overcomes the potential local optimality issues of pure local planning, thereby achieving a more efficient and reliable path planning strategy.

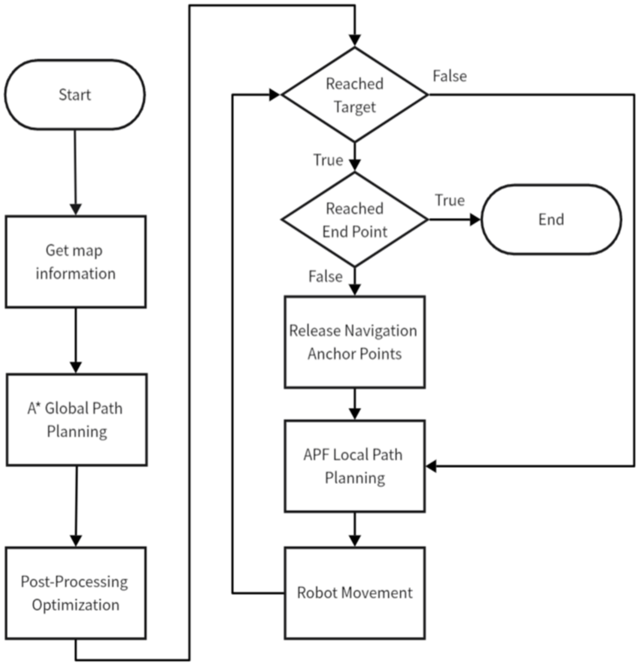

Fig. 2. Flowchart of Hybrid Algorithm

5 Results and Discussion

5.1 Performance Analysis of A* Optimization

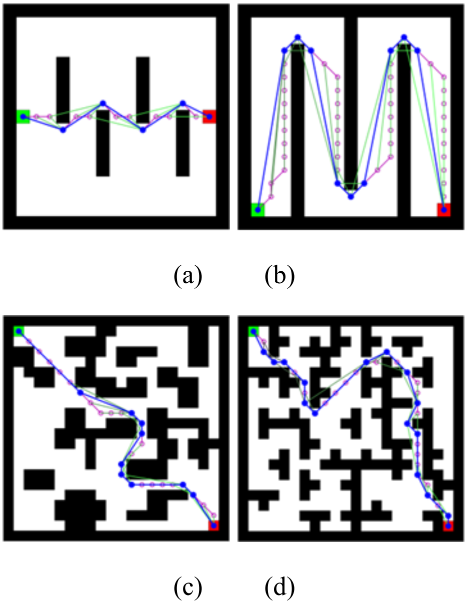

Fig. 3. Generation results of four types of maps. (a) Map with multiple unnecessary turning points, (b) Map with long optimizable paths, (c) and (d) Two random maps

Table 1. Optimized A* Path Lengths

	Original Path Length (m)	Optimize Path Length (m)
a	17.314	16.215
b	55.799	53.212
c	40.87	39.906
d	33.213	32.324

Table 2. Optimized A* Number of Turns

	Original Number of Turns	Optimized Number of Turns
a	12	4
b	12	9
c	18	15
d	9	9

For the path optimization of A*, this paper selects two typical maps and two randomly generated maps, namely MAP1 (Fig. 3.a map with multiple unnecessary turning points), MAP2 (Fig. 3.b map with a long optimizable path), and MAP3 and MAP4 (Fig. 3.c, d two randomly generated maps).

In the experiment, the purple path in the above figure is the path determined by the traditional A*, while the blue path is the better path obtained by greedily optimizing the original path based on collision detection. As evident from the figure, the optimized path demonstrates greater conciseness in managing turning points, the overall path is smoother, and the unnecessary turning points are effectively removed, making the movement trajectory of the robot or intelligent agent more natural and efficient.

As indicated in Table 1 and Table 2, the length of the optimized path and the number of turns has been effectively reduced, especially in typical cases. In MAP a, due to the existence of many unnecessary turning points, the optimization algorithm significantly shortened the length of path and reduced quantity of curvature points by eliminating unnecessary turns, thereby reducing the energy consumption and time cost of the robot's steering during path execution. In MAP b, the long path provided a greater space for optimization. The optimized path not only had a significant reduction in length but also improved the degree of path tortuosity, enabling the robot to reach the target position more quickly.

For randomly generated MAP c and MAP d, the optimization algorithm also demonstrated its effectiveness. Despite the varying distributions of obstacles and path shapes in the random maps, a better solution than that generated by the traditional A* algorithm could still be found through the greedy optimization strategy based on collision detection. This indicates that the optimization method has good universality and flexibility and capable of operating in across various settings.

5.2 Dynamic Obstacle Avoidance Analysis of APF Algorithm

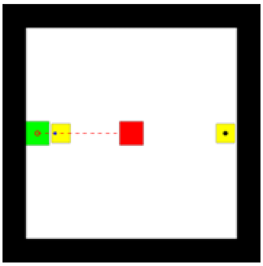

Fig. 4. Obstacle Avoidance Environmental Model

As shown on Fig. 4, in the APF dynamic obstacle avoidance analysis experiment, this paper sets up an experimental scenario: the robot (blue particle) commences its journey at the starting point (green area) and makes its way heading for the destination (red area). After the robot traverses the yellow area, the dynamic barrier (black particle) moves towards the robot at high speed. This setting aims to simulate the possible sudden situations in a unstable scenario and check the capabilities of the improved APF algorithm under extreme conditions. As shown in Table 3 above, the experimental results indicate even when the speed of the obstacle reaches ten times that of the robot, the improved APF algorithm can still effectively guide the robot to avoid the obstacle.

This result verifies that the improvement measures of the potential field range in this paper significantly enhance the robustness of the local planning algorithm.

Table 3. Dynamic Obstacle Avoidance Capabilities of the Improved APF Algorithm

Robot Speed: Dynamic Obstacle Speed	APF Response Time	APF (Improved) Response Time	APF Avoidance Distance	APF (Improved) Avoidance Distance
0.1 : 0.2	3	2	5.08	5.64
0.1 : 0.4	2.2	1.6	2.88	4.41
0.1 : 0.8	1.6	1.2	1.8	2.95
0.125 : 1.25	none	0.125	none	2.31

5.3 Performance Analysis of Improved APF and A* Hybrid Planning

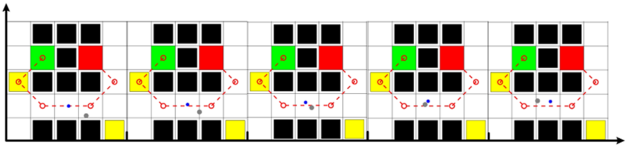

Fig. 5. Traditional APF Obstacle Avoidance

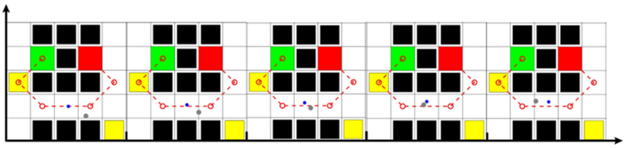

Fig. 6. Improved APF Obstacle Avoidance

The above figure shows the verification of the correspondingly planning of A* algorithm and APF algorithm. From Fig. 5-6, it is evident that the hybrid planner can effectively guide the robot (blue particle) to complete the navigation and obstacle avoidance tasks. Moreover, from the comparison between Fig. 5 and Fig. 6, it is evident that the APF algorithm after improving the potential field range can guide the robot to make evasive reactions in advance to approaching obstacles. This means that the robot can adjust its movement trajectory before the obstacle enters its safety area, thereby avoiding sharp turns or speed changes in emergency situations. In addition, the improved algorithm also enables the robot sustains a safe distance from obstacles, reducing the risk of collision and improving the navigation safety of the robot in complex environments.

5.4 Discussion

The outcomes of the simulation reveal that using the node sequence planned by A* algorithm as the anchor points for the potential field direction can effectively address the local minima and convergence failure issues faced by conventional APF algorithm in static maps. The global path produced by the A* algorithm provides clear direction guidance for APF, helping the robot avoid local minima regions and ensuring its smooth arrival at the target point. However, when the movement path of dynamic obstacles coincides with the path planned by A*, a so-called "dynamic trap" occurs. In such cases, even though the improved APF algorithm incorporates a vertical repulsive force direction to guide the robot out of the predicament, the dynamic trap problem persists in certain situations and has not been fundamentally resolved, as shown in the following figure.When dynamic obstacles move towards the robot along the path planned by A*, the robot might become trapped in a local minimum and be unable to adjust its path promptly to evade the obstacles. Although the improved APF algorithm helps the robot get out of this predicament by introducing a force in the vertical repulsion direction, this method may not always be effective in certain specific situations. For instance, when the speed of the obstacle is very fast or completely opposite to the speed of the robot, the robot may not be able to detect the threat of the obstacle in time and respond accordingly. Additionally, if the shape or movement trajectory of the obstacle is complex, it may also make it difficult for the robot to accurately judge and avoid the obstacle.

6 Conclusion

The paper presents a path planning approach that combines a modified A* algorithm with an optimized APF algorithm. It has been verified that this method plans the global path through the A* algorithm and guides the APF algorithm for local dynamic obstacle avoidance, effectively solving the problems of easy trapping in local minima and convergence failure of the traditional APF algorithm in static maps, significantly improving the overall performance of path planning. Meanwhile, by segmenting and optimizing the A* path through collision detection, the quantity of nodes and changes of direction in the path is reduced, further shortening the path length and enhancing the efficiency and simplicity of the path. Additionally, by constructing an elliptical potential field that varies with speed in the polar coordinate system, the improved APF algorithm performs well in dynamic obstacle avoidance, especially in high-speed dynamic obstacle scenarios, demonstrating a faster response speed and a larger safe obstacle avoidance distance. It also maintains compatibility with the traditional APF algorithm in static obstacle handling without the need for additional parameter adjustments, enhancing the adaptability and practicality of the algorithm.

Of course, there is still room for improvement in the computational complexity and algorithmic responsiveness in intricate scenarios. Future work will focus on optimizing the algorithm's computational efficiency to further reduce the computational cost of path planning and adapt to more complex dynamic environments. Additionally, the algorithm is planned to be applied in practical scenarios such as robot navigation and autonomous driving, to verify its robustness and reliability in real environments through experiments, and to explore how to dynamically adjust algorithm parameters in different scenarios to further enhance its adaptability and practicality.

References

1. Li, S., Wang, J.. ‘Parallel parking path planning in narrow space based on a three-stage curve interpolation method’. IEEE Access, 2023, 11, 93841.
2. Ziegler, J., Bender, P., Dang, T., et al.. ‘Trajectory planning for Bertha—a local, continuous method’. In IEEE Intelligent Vehicles Symposium Proceedings. IEEE, 2014, 450–457.
3. Gao, J., Xu, X., Pu, Q., Petrovic, P.B., Rodić, A., Wang, Z.. ‘A Hybrid Path Planning Method Based on Improved A* and CSA-APF Algorithms’. IEEE Access, 2024, 12, 39139–39151.
4. Sang, H., You, Y., Sun, X., Zhou, Y., Liu, F.. ‘The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations’. Ocean Engineering, 2021, 223, 108709.
5. Liang, C., Zhang, X., Watanabe, Y., Deng, Y.. ‘Autonomous collision avoidance of unmanned surface vehicles based on improved A* and minimum course alteration algorithms’. Applied Ocean Research, 2021, 113, 102755
6. Seo, C., Noh, Y., Abebe, M., Kang, Y.J., Park, S., Kwon, C.. ‘Ship collision avoidance route planning using CRI-based A* algorithm’. International Journal of Naval Architecture and Ocean Engineering, 2022, 15, 100551
7. Moradi, M.H., Brutsche, M., Wenig, M., Wagner, U., Koch, T.. ‘Marine route optimization using reinforcement learning approach to reduce fuel consumption and consequently minimize CO_2 emissions’. Ocean Engineering, 2022, 259, 111882.
8. Labakhua, L., Nunes, U., Rodrigues, R., et al.. ‘Smooth trajectory planning for fully automated passenger vehicles: Spline and clothoid-based methods and its simulation’. In Informatics in Control Automation and Robotics: Selected Papers from the International Conference on Informatics in Control Automation and Robotics. Springer, Heidelberg, 2008, 169–182.
9. Bergman, K., Ljungqvist, O., Axehill, D ‘Improved path planning by tightly combining lattice-based path planning and optimal control’. IEEE Transactions on Intelligent Vehicles, 2020, 6(1), 57–66.
10. Chai, R., Tsourdos, A., Savvaris, A., et al. ‘Six-DOF spacecraft optimal trajectory planning and real-time attitude control: a deep neural network-based approach’. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(11), 5005–5013.