vehicles

MDPI

Article

Dynamic Obstacle Avoidance Approach Based on Integration of A-Star and APF Algorithms for Vehicles in Complex Mountainous Environments

Changlong Chen [1], Yuejin Lin [2], Lulin Zhan [1], Yuling He [3,*], Yi Zhang [1], Xiqiang Chi [1] and Menghu Chen [1]

[1] Wenzhou Power Construction Co., Ltd., Wenzhou 325000, China; chenchanglong1616@163.com (C.C.); f283111@163.com (L.Z.); zhangyi_0577@163.com (Y.Z.); chixiqiang@163.com (X.C.); happytiger1984@126.com (M.C.)
[2] Wenzhou Tusheng Holdings Group Co., Ltd., Wenzhou 325000, China; tushengjituanljh@163.com
[3] Hebei Engineering Research Center for Advanced Manufacturing & Intelligent Operation and Maintenance of Electric Power Machinery, North China Electric Power University, Baoding 071003, China
* Correspondence: heyuling1@ncepu.edu.cn

Abstract

Complex mountainous environments pose significant challenges for dynamic path planning and obstacle avoidance of transport vehicles. In response, this paper presents an innovative path planning approach that combines an enhanced A* algorithm with the artificial potential field (APF) method. Firstly, the heuristic function of the A* algorithm was improved, and path inflection points were optimized to enhance global path-planning efficiency and smoothness. Secondly, a target distance factor was introduced to modify the APF algorithm's repulsive field function, solving the traditional APF's target-unreachable problem. The integrated algorithm uses the A*-optimized inflection points as sub-target points for the APF, meeting real-time obstacle avoidance requirements in dynamic environments and conducting secondary path planning to avoid local minima. Impressively, static environment simulations demonstrated the integrated algorithm's outstanding path-planning capabilities in complex terrains. Moreover, dynamic obstacle avoidance experiments revealed its remarkable ability to not only detect and evade dynamic obstacles but also maintain a safe distance from static ones. The findings highlight that this method significantly boosts path-planning efficiency while ensuring safety and global optimality in dynamic settings. This breakthrough offers crucial theoretical support for enhancing the navigation of mountain transport vehicles in complex, real-world scenarios, potentially improving their operation.

Keywords: improved A* algorithm; artificial potential field method; path planning; dynamic obstacle avoidance

check for updates

Academic Editors: Sijing Guo, Bin Wang, Quan Zhou and Bin Shuai

Received: 18 May 2025
Revised: 9 June 2025
Accepted: 25 June 2025
Published: 29 June 2025

Citation: Chen, C.; Lin, Y.; Zhan, L.; He, Y.; Zhang, Y.; Chi, X.; Chen, M. Dynamic Obstacle Avoidance Approach Based on Integration of A-Star and APF Algorithms for Vehicles in Complex Mountainous Environments. *Vehicles* **2025**, *7*, 65. https://doi.org/10.3390/vehicles7030065

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Dynamic path planning is a core aspect of mountain transport vehicle research, involving the planning of a driving route from a starting point to a destination in complex terrains with dynamic obstacles. It ensures that the vehicle can safely avoid obstacles and maintain stable passage during movement. Depending on the degree of environmental information available during path planning, path planning algorithms can be categorized into global and local path planning. Currently, common global path planning algorithms include the A* algorithm [1], particle swarm optimization [2], and the ant colony algorithm [3], which primarily optimize the global path based on pre-acquired terrain data.

Local path planning typically employs methods such as the artificial potential field (APF) [4] and fuzzy logic algorithms [5], which focus on real-time processing of sudden dynamic obstacles encountered during travel, adjusting the path dynamically using sensor data. In obstacle avoidance, geometric and reinforcement learning (RL) approaches address dynamic navigation challenges differently. The VFH+T algorithm [6] enhances traditional VFH+ with memory-based trap recognition. Meanwhile, an optimized TD3 [7] allows the robot to take its own actions based on the observations it makes, without defining any trajectory beforehand.

The A* algorithm is a heuristic search algorithm characterized by its concise implementation and higher operational efficiency compared to algorithms such as particle swarm and ant colony algorithms. However, the traditional A* algorithm has notable drawbacks, including an excessive number of inflection points and non-smooth paths in the generated trajectories. Early researchers aimed to reduce pathfinding time by adjusting the weight ratios within the cost function [8]. Nevertheless, this approach exhibited poor environmental adaptability, particularly in complex mountainous terrains. Subsequent scholars focused on modifying the cost function to enhance the heuristics of the heuristic function. For example, Yu Xiang et al. [9] improved the A* heuristic function using Euclidean distance and projection distance, a modification that significantly reduced the number of iterations. Bao Jiusheng et al. [10] proposed incorporating an exponential function as a weighting coefficient for the heuristic function, a strategy that yielded a moderate improvement in the algorithm's search efficiency.

The APF algorithm determines the next movement direction based solely on the surrounding environment and the robot's current state, offering advantages such as high computational efficiency and smooth path generation. However, this method is prone to stagnation at local minima, and traditional APF struggles to meet optimal path criteria when facing dynamic obstacles. To address these limitations, Min Gyu Park et al. [11] addressed the problem that the APF algorithm is prone to getting trapped in local minima during path planning. They proposed a technique that combines the APF algorithm with the simulated annealing algorithm to effectively avoid local minima and achieve path planning. Additionally, Jinseok Lee et al. [12] proposes a random unit total force (RUTF) algorithm and its repulsion-removed variant (RUTF-RR), which dynamically select force strategies based on the positional and directional conditions of the robot, obstacles, and target. This approach is designed to systematically identify and resolve the local minimum trapping issue inherent in potential field methods, particularly under symmetric robot–obstacle–goal configurations.

In complex mountainous environments, unstructured terrains—such as steep slopes, gullies, and irregular obstacles—and dynamic factors including moving obstacles and real-time environmental perception noise pose significant challenges to vehicle path planning. While single global planning algorithms (e.g., the traditional A* algorithm) can generate globally optimal paths based on prior maps, they struggle to respond in real-time to dynamic obstacles and local terrain mutations. Conversely, local obstacle avoidance algorithms (e.g., the artificial potential field (APF) method) can achieve immediate obstacle avoidance but suffer from local minimum traps and lack global path guidance, leading to deviations from optimal paths and even target unreachability. The inherent limitations of these two algorithms are significantly amplified by the complexity of mountainous environments. Therefore, the organic integration of the systematic global planning with the real-time adaptability of local obstacle avoidance to construct a dynamic obstacle avoidance approach that balances global optimality and dynamic adaptability has become a critical breakthrough for solving navigation problems of mountain vehicles in complex environments.

In the fields of autonomous driving and unmanned aerial vehicles (UAVs), improvements to the A* and APF algorithms primarily focus on path smoothing, dynamic obstacle avoidance, search efficiency, and constraint integration. In autonomous driving, cubic B-spline/Bezier curve fitting is introduced to optimize the curvature continuity of A*-generated paths, addressing issues of redundant inflection points [13,14]. Weighted heuristic functions and bidirectional search strategies are employed to enhance search efficiency in complex environments and reduce redundant nodes [15,16]. To tackle the local minimum traps of the APF algorithm, repulsive field functions are improved to be distance-sensitive—for example, by incorporating target distance factors or dynamically adjusting repulsive force intensity—and combined with safety distance models or road boundary constraints (e.g., virtual lane line gravitational forces) to enhance global guidance [17,18]. In the UAV domain, repulsive-force optimized heuristic functions enable safe path planning and load stabilization, while lightweight search strategies drastically reduce search time [19].

To address the above challenges, this paper proposes an algorithm that integrates the A* algorithm and the APF algorithm. The algorithm optimizes the set of inflection points in the global path generated by the improved A* algorithm and uses these optimized inflection points as sub-target points for obstacle avoidance in the improved APF algorithm. This approach not only reduces the iteration time but also shortens the path length.

2. Traditional Path Planning Algorithm

2.1. Traditional A Algorithm*

The core idea of the A* algorithm in path planning is to integrate a heuristic search strategy. It calculates two components: the actual cost function g(n) from the start node to the current node, and the heuristic function h(n) estimating the cost from the current node to the target node. The combined evaluation criterion is the total cost f(n) = g(n) + h(n). A priority queue is used to store nodes to be expanded, sorted by f(n). During the search, the algorithm dynamically updates path information and performs effective pruning. Under the condition that h(n) is an admissible heuristic function, A* guarantees finding the optimal path. Typically, the heuristic function h(n) is represented by the Euclidean distance between the current node and the target node:

$$\mathrm{h}(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \tag{1}$$

where x_n and y_n represent the X-axis and Y-axis coordinates of the current node, respectively, while x_g and y_g denote the X-axis and Y-axis coordinates of the target node.

The flowchart of the traditional A* algorithm is shown in Figure 1. However, the traditional A* algorithm has notable drawbacks in its computational results. First, it involves an excessive number of search nodes and long computation time: in complex environments, the search space grows exponentially, and the heuristic function struggles to adapt to most environments. Second, issues such as non-smooth explored paths and suboptimal routes arise: the paths are jagged with frequent turns, and the algorithm cannot adjust the path in a timely manner in dynamic environments.

2.2. Traditional APF Algorithm

The traditional APF algorithm is based on the principle of gradient potential fields. The attractive potential field U_{att} experienced by an object is given by

$$\mathrm{U}_{att} = \frac{1}{2} k_{att} |\rho|^2 \tag{2}$$

where k_{att} denotes the attractive gain coefficient, and ρ represents the spatial vector from the current position of the vehicle to the target. This vector is directed from the current position to the target, with its magnitude equal to the distance between the two points.

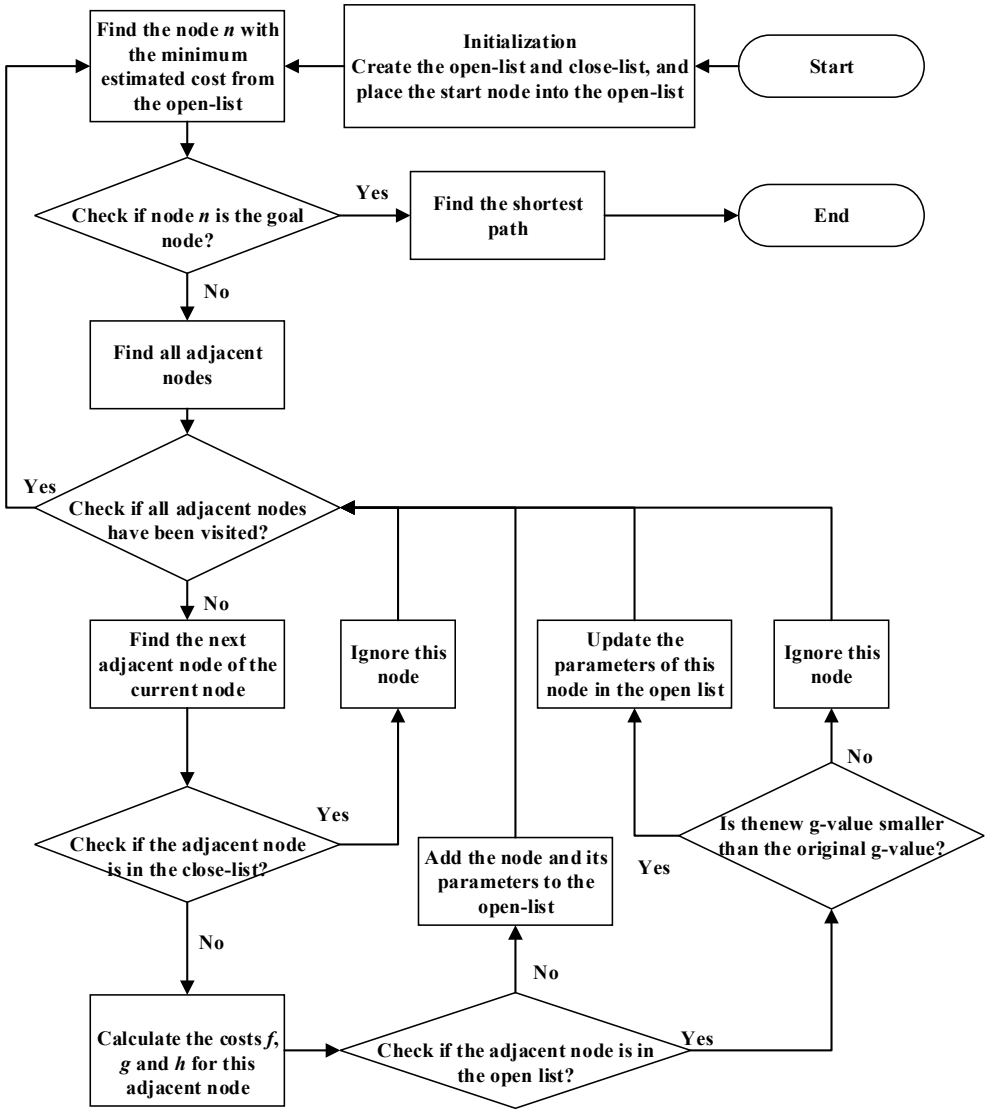


Figure 1. Traditional A* algorithm flowchart.

The magnitude of the attractive force acting on the vehicle is given by the negative gradient of the attractive potential field function:

$$\mathrm{F}_{att} = -\nabla U_{att} = k_{att}\rho \tag{3}$$

The repulsive potential field U_{obs} experienced by the vehicle around an obstacle is given by

$$U_{obs} = \begin{cases} k_{obs}\left(\frac{1}{q} - \frac{1}{\lambda}\right)^2 & q \le \lambda \\ 0 & q > \lambda \end{cases} \tag{4}$$

where k_{obs} represents the repulsive gain coefficient, q denotes the shortest distance from the vehicle to the obstacle, and λ is the radius of the obstacle's repulsive field range.

This means that the obstacle exerts a repulsive force on the vehicle when it enters the repulsive field range, while it has no effect outside this range. Specifically, the repulsive force exerted by the obstacle on the vehicle is given by

$$F_{obs} = -\nabla U_{obs} = \begin{cases} k_{obs}\left(\frac{1}{q} - \frac{1}{\lambda}\right)\frac{1}{q^2} & q \le \lambda \\ 0 & q > \lambda \end{cases} \tag{5}$$

3. Improvement of Path Planning Algorithm

In the operation process of a vehicle, the working environment is a critical factor influencing its path planning. Common methods for constructing environmental maps include the grid method, vector method, and free space method. Among these, the grid method is widely used for its intuitive, simple, and easy-to-establish nature, as well as its good compatibility with the A* algorithm. It works by finely dividing the area where the vehicle operates into numerous grids, thereby constructing an intuitive and easy-to-manipulate representation model.

3.1. Improvement of A Algorithm*

(1) Improvement of Heuristic Function

The A* algorithm uses the cost function f(n) to calculate each reachable node, selecting the node with the lowest cost to determine the next step. By reasonably modifying the combined cost function f(n) of the traditional A* algorithm and increasing the weight of the heuristic function h(n), search efficiency can be improved. Specifically, when h(n) is always smaller than the actual cost g(n) from the current node to the target, the search space expands, leading to a larger number of explored nodes. Adjusting the weight of h(n) to give it a larger proportion in the calculation can more accurately direct the search direction, reducing unnecessary node exploration and enhancing the algorithm's search efficiency. However, an excessive weight on the heuristic function can cause the algorithm to over-explore a single direction during pathfinding, ignoring other possible routes. In this case, the resulting path will no longer be optimal. The improvement method is to modify the heuristic function to not only consider the distance from the current node to the target, but also the distance from the current node to the starting point and the line connecting the target.

Reference [7] introduces an additional component to the heuristic function, which considers not only the distance from the current node to the target but also the distance from the current node to the line connecting the start and target points. This approach can narrow the search area and enable the algorithm to approach the target more quickly. Specifically, it modifies the heuristic function h(n) to

$$h(n) = \frac{|ax_n + by_n + c|}{\sqrt{a^2 + b^2}} + \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \tag{6}$$

where a, b, and c are the three parameters of the linear equation representing the line connecting the start and target points.

Reference [8] does not modify the heuristic function but instead adopts an exponential function as the weighting coefficient for the heuristic function. The optimized cost function f(n) is given by

$$\begin{aligned} f(n) &= g(n) + \exp^{h(\mathrm{n})} \times h(n) \\ &= g(n) + \exp^{h(\mathrm{n})} \times \left(\frac{|ax_n + by_n + c|}{\sqrt{a^2 + b^2}} + \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}\right) \end{aligned} \tag{7}$$

Both of the above methods aim to enhance the heuristics of the cost function by increasing the proportion of the heuristic function. Thus, this paper integrates these two approaches to form a new cost estimation function. Taking an L-shaped obstacle as an example, Figure 2 compares the specific path results of four methods. In Figure 2, white grids represent unexplored nodes; blue grids denote examined nodes; black grids indicate obstacles; the red and blue grids mark the start and target nodes, respectively; and the red solid line shows the final path found by the algorithm.

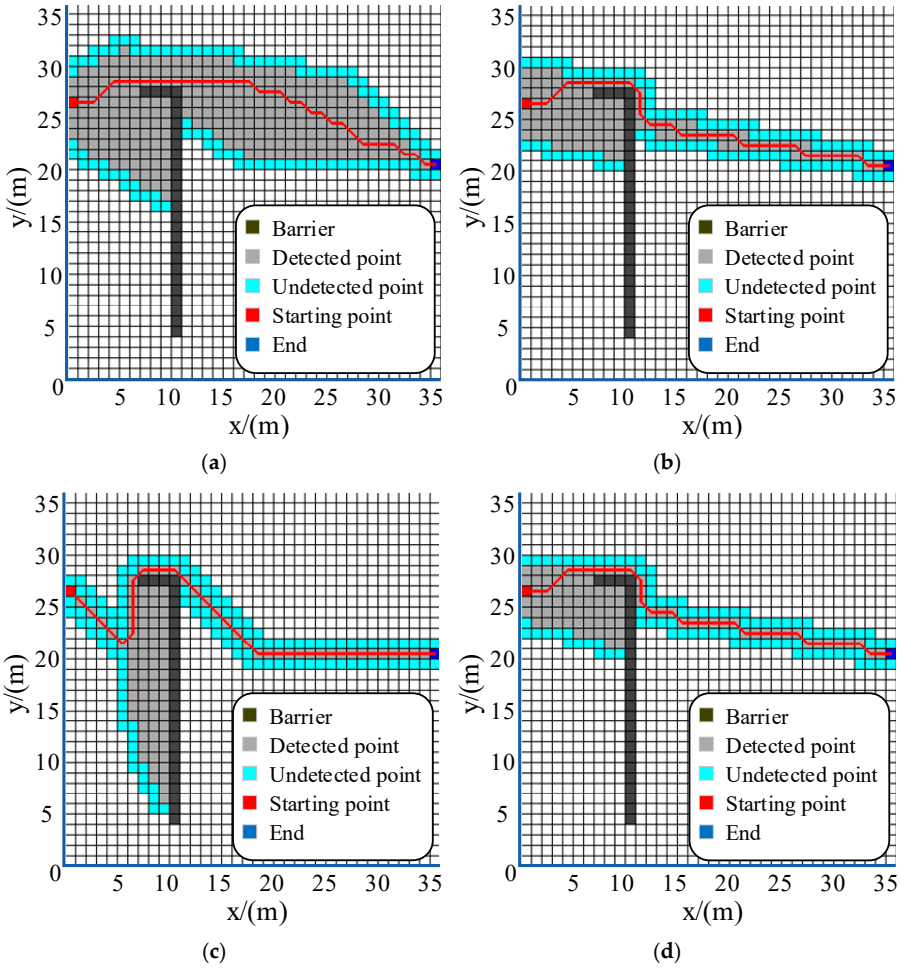


Figure 2. Schematic diagram of path planning under different algorithms: (**a**) traditional A* algorithm path; (**b**) A* algorithm path in reference [7]; (**c**) the A* algorithm path in reference [8]; (**d**) improving the weighted A* algorithm path.

As shown in Figure 2, the search area of the traditional A* algorithm is significantly larger than that of the other three methods. Modifying the heuristic function clearly restricts the search scope: the inclusion of the distance from the current node to the line connecting the start and target points increases the heuristic value for nodes deviating from this line, thereby reducing their likelihood of being selected. Additionally, a comparison between Figure 2c,d reveals that the improved heuristic function exhibits a pronounced convergent effect, narrowing the search focus effectively. Comparing Figure 2a,c shows that while the exponential weighting method enhances heuristic guidance, it struggles to navigate across rectangular obstacles. In mountainous environments with gullies and fallen trees, relying solely on exponential function improvements is clearly insufficient to adapt to the complex terrain.

The specific data for each algorithm in Figure 2 are shown in Table 1. In the "Running Time" column, the first column is the time the algorithm is running, and the second column

is the percentage of time saved compared to traditional algorithms In terms of search efficiency, whether by modifying heuristic functions or applying exponential weighting, it has excellent optimization effects compared to traditional algorithms, reducing search time by 57.7% and 56.2%, respectively. However, due to the limitations of exponential weighting, the path generated by this method is significantly longer than those of the other three approaches. Combining the heuristic function modification with exponential weighting overcomes this shortcoming. The results also indicate that the number of nodes has a greater impact on runtime than algorithm complexity to some extent. Since the traditional A* algorithm has weak heuristics and conducts more thorough exploration of surrounding nodes, it is reasonable that it achieves the shortest path among the methods compared.

Table 1. Statistical results of path planning under different algorithms.

Algorithm	Number of Searched Nodes	Path Length	Running Time	
Traditional A* algorithm	267	39 m	5.994 s	--
Improved heuristic function	99	40.314 m	2.536 s	↓ 57.7%
Exponential weighted A* algorithm	105	46.21 m	2.626 s	↓ 56.2%
Improved exponential weighted A* algorithm	90	40.314 m	2.491 s	↓ 58.4%

(2) Path optimization

In the scenario depicted in Figure 2, even after completing path planning, the resulting path still suffers from issues such as an excessive number of inflection points and a relatively long total length. Therefore, it is necessary to optimize this path. First, all inflection points in the path are extracted in sequence. The inflection point is the path inflection points in the global path generated by the improved A* algorithm, which can serve as sub objectives of the APF algorithm to facilitate dynamic obstacle avoidance. Notably, both the start node and the target node are treated as special inflection points and included in the extraction process. After extracting these points, the algorithm traverses each inflection point starting from the start node. If there are no obstacles between the start node and the current inflection point, the algorithm skips this point. If obstacles exist, it retains the previous inflection point and resumes traversal from that point to the subsequent ones, continuing until reaching the target node. Through this series of operations, the final generated path becomes the theoretically shortest path.

To generate random obstacles more consistent with mountainous environments, as shown in Figure 3, four algorithms were used for path planning in this environment, where the yellow solid line represents the optimized path. As indicated by Figure 3, after path optimization, the number of inflection points in the path is significantly reduced. Through statistical analysis of various data before and after algorithm optimization, Table 2 is obtained. The results show that this path optimization can substantially reduce the number of inflection points and effectively shorten the path length.

Table 2. Data before and after path optimization.

Algorithm	Number of Searched Nodes	Number of Inflection Points		Path Length	
		Before	After	Before	After
Traditional A* algorithm	184	16	5	38.31 m	36.19 m
Improved heuristic function	66	13	5	38.31 m	36.21 m
Exponential weighted A* algorithm	40	19	6	44.46 m	40.07 m
Improved exponential Weighted A* algorithm	41	13	5	38.31 m	36.21 m

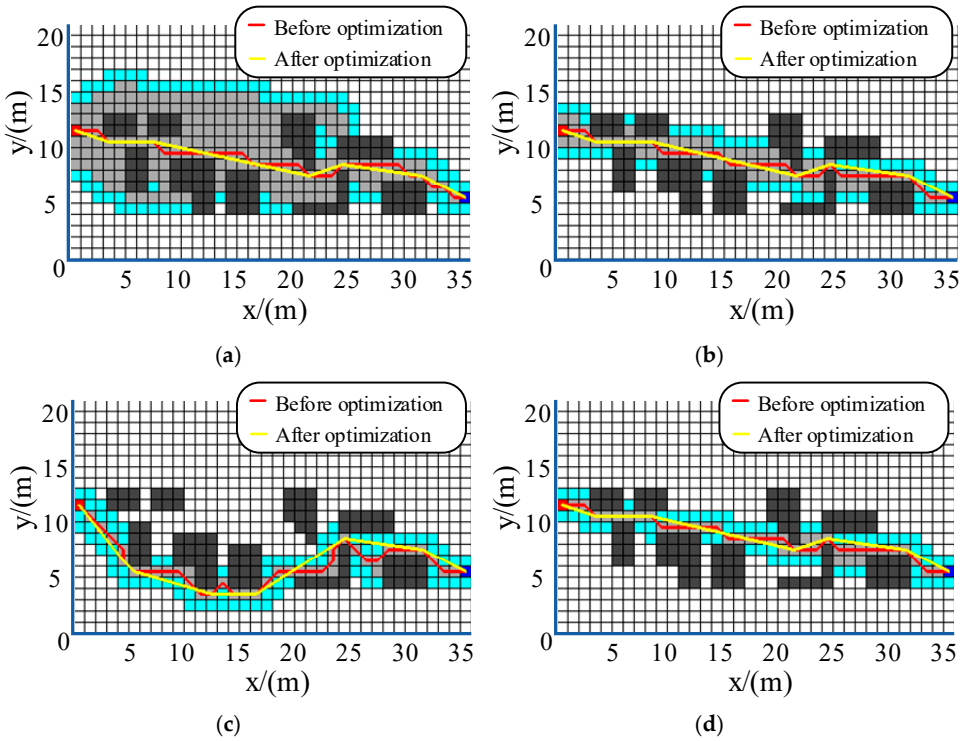


Figure 3. Schematic diagram of path optimization: (**a**) traditional A* algorithm path; (**b**) A* algorithm path in Reference [7]; (**c**) the A* algorithm path in reference [8]; (**d**) improving the path of the index weighted A* algorithm.

3.2. Improvement of APF Algorithm

In the process of numerous practical applications, the traditional APF algorithm has two pressing issues to be resolved. The first is the local minimum problem, and the second is the target unreachable problem. Figure 4 shows the basic flowchart of the APF algorithm. The main idea of this paper is to divide the path into multiple sub-target points under the guidance of the A* algorithm, and when a local optimum point is encountered, the path is replanned. Since these are the two major problems of the APF algorithm, the improvement of the APF in this paper mainly focuses on the target unreachable problem.

In certain special environments, such as when obstacles are close to the target point, the traditional APF algorithm may prevent the robot from reaching the target. This is primarily because when the robot approaches the target, the repulsive force remains unchanged while the attractive force decreases significantly, leading to the attractive force being smaller than the repulsive force from obstacles, thus making it impossible for the robot to reach the target. To address this issue, the repulsive field function is modified by incorporating the distance between the robot and the target point. The improved repulsive field function, i.e., the repulsive force function, is given by

$$U'_{obs} = \begin{cases} \frac{1}{2}k_{obs}\left(\frac{1}{q} - \frac{1}{\lambda}\right)\rho^n & q \le \lambda \\ 0 & q > \lambda \end{cases} \tag{8}$$

$$\vec{F}_{obs} = -\nabla U'_{obs} = \begin{cases} \vec{F}_{obs1} + \vec{F}_{obs2} & q \le \lambda \\ 0 & q > \lambda \end{cases} \tag{9}$$

$$\begin{cases} |F_{obs1}| = k_{obs}\left(\frac{1}{q} - \frac{1}{\lambda}\right)\frac{1}{q^2}\rho^n \\ |F_{obs2}| = \frac{1}{2}k_{obs}\left(\frac{1}{q} - \frac{1}{\lambda}\right)\rho^{n-1} \end{cases} \tag{10}$$

where n is a real number greater than 0, and F_{obs1} and F_{obs2} represent the repulsive force components. Specifically, F_{obs1} is directed from the obstacle to the vehicle, while F_{obs2} is directed from the vehicle to the target. This separation mirrors the potential field's dual functional form, enabling independent tuning of distance-based repulsion and goal-oriented navigation behaviors.

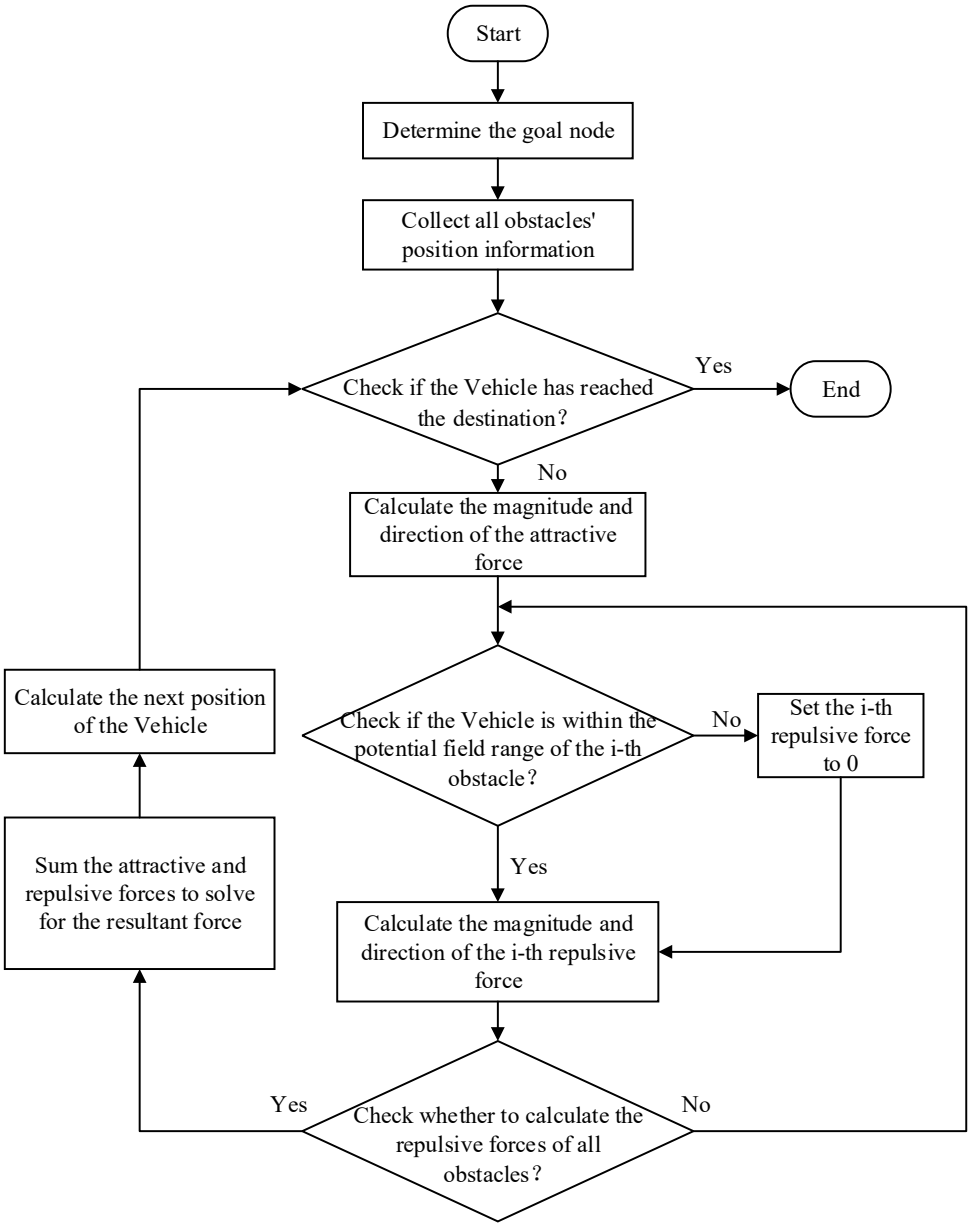


Figure 4. Traditional APF algorithm flowchart.

3.3. Improved A Algorithm Combined with APF Algorithm*

Through improvements, the A* algorithm effectively addresses the issue of low search efficiency. After path optimization, it can plan an optimal path. The APF algorithm, built upon the paths generated by the improved A* algorithm, is used to solve obstacle avoidance problems in dynamic environments. Meanwhile, when the APF algorithm encounters a local optimum, it replans the path to avoid this limitation. This paper integrates the two algorithms, enabling the combined approach to exhibit strong path-planning capabilities in both static environments and dynamic scenarios, as shown in Figure 5.

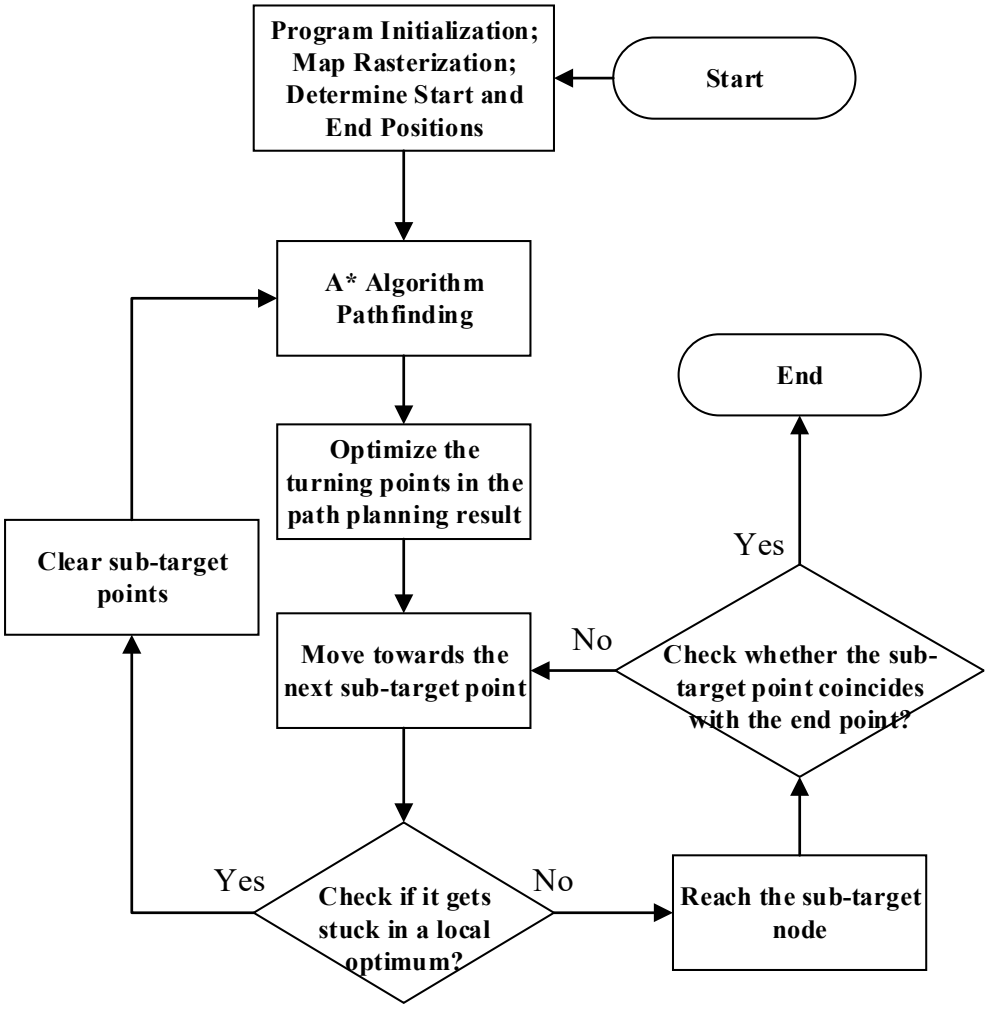


Figure 5. Flowchart of the fusion of the A* algorithm and APF algorithm.

4. Simulation Verification of Fusion Algorithm

4.1. Static Planning

In mountainous environments, the distribution of obstacles is often complex, with narrow areas where planned paths may appear passable but are actually unnavigable in reality. Random obstacles were created on a 36 × 36 grid map, with the vehicle starting at coordinates (1, 27) and ending at (36, 21). Figure 6 shows the pathfinding results of the traditional APF algorithm and the APF algorithm guided by the improved A* algorithm, where blue dashed lines represent the vehicle's movement trajectory.

As shown, the traditional APF algorithm failed to bypass large obstacles in complex setups (Figure 6a), struggling to navigate long strip-shaped barriers due to its lack of global path prediction. In contrast, the A*-guided APF algorithm generated a more reasonable and shorter path (Figure 6b), leveraging optimized inflection points from the improved A* as sub-targets for dynamic obstacle avoidance. However, single-pass planning may still trap the vehicle in local minima. To address this, the algorithm marked the current position as an obstacle and initiated secondary path planning upon detecting stagnation, ensuring successful destination reachability (Figure 6c). This validation highlights that integrating A*'s global optimization (via heuristic function modification and path pruning) with APF's local adaptability (via target-distance factor adjustment) effectively balances real-time obstacle avoidance and global path optimality in complex terrains.

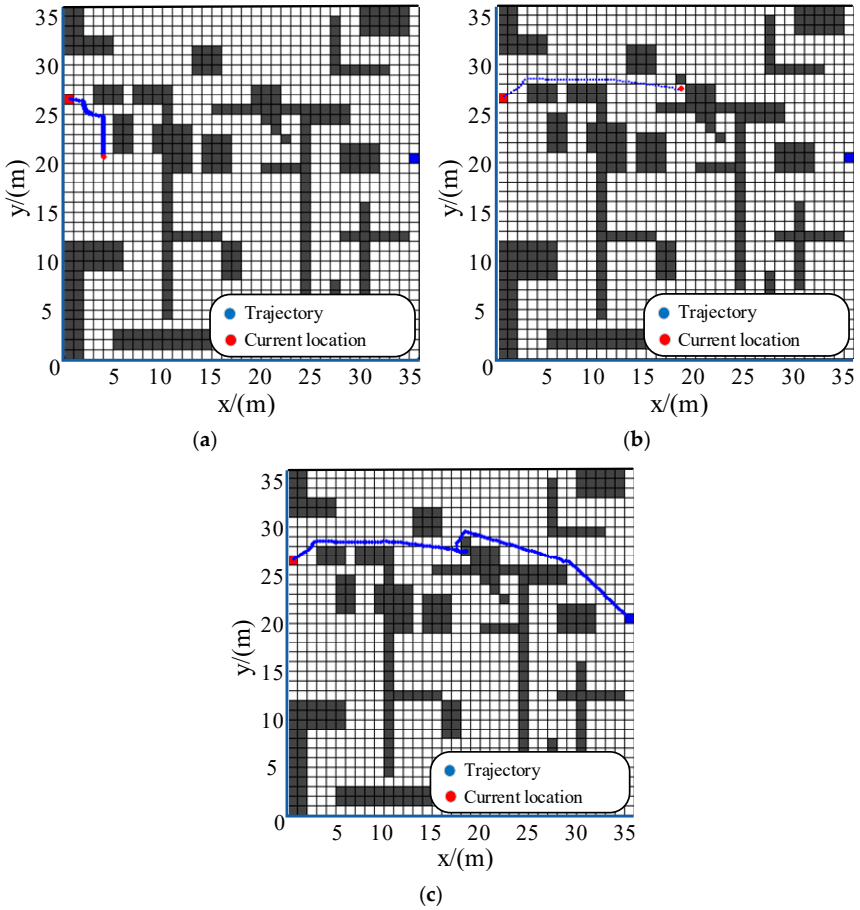


Figure 6. Comparison of static environment pathfinding: (**a**) traditional APF algorithm; (**b**) A* and APF fusion algorithm; (**c**) second planning.

4.2. Dynamic Obstacle Avoidance

In mountainous environments, there are not only numerous static obstacles but also a small number of dynamic obstacles. To simplify calculations, this section adds dynamic obstacles to the static environment. Dynamic obstacles were modeled using a constant velocity framework with linear trajectories, spanning velocities of 0.5–1.0 m/s to simulate slow-moving debris or wildlife. Reciprocal linear motion served as the primary dynamic model, while sinusoidal trajectories were included in supplementary simulations to emulate erratic movements. Safety margins were $\geq$0.5 m for static obstacles and $\geq$1.0 m for dynamic ones, enforced via adaptive modifications to the APF repulsive field. Among them, three dynamic obstacles that were colored dark green moved reciprocally along diagonal, longitudinal, and lateral directions, respectively. The red line represents the path planned by the global algorithm, while the blue line denotes the real-time route adjusted for dynamic obstacles.

As shown in Figure 7, when the vehicle was approaching dynamic obstacles, the APF algorithm detected whether these obstacles affect its movement. When dynamic obstacles posed a threat, the bike initiated obstacle avoidance in advance. Meanwhile, while evading dynamic obstacles, it maintained a safe distance from static obstacles. After completing the avoidance maneuver, the vehicle returned to the route planned by the global path planning algorithm and eventually reached the target safely, verifying the effectiveness and safety of the local path planning. From the above simulation tests, it can be seen that during several instances of evading dynamic obstacles, the vehicle made significant deviations but successfully returned to the original route and reached the destination smoothly.

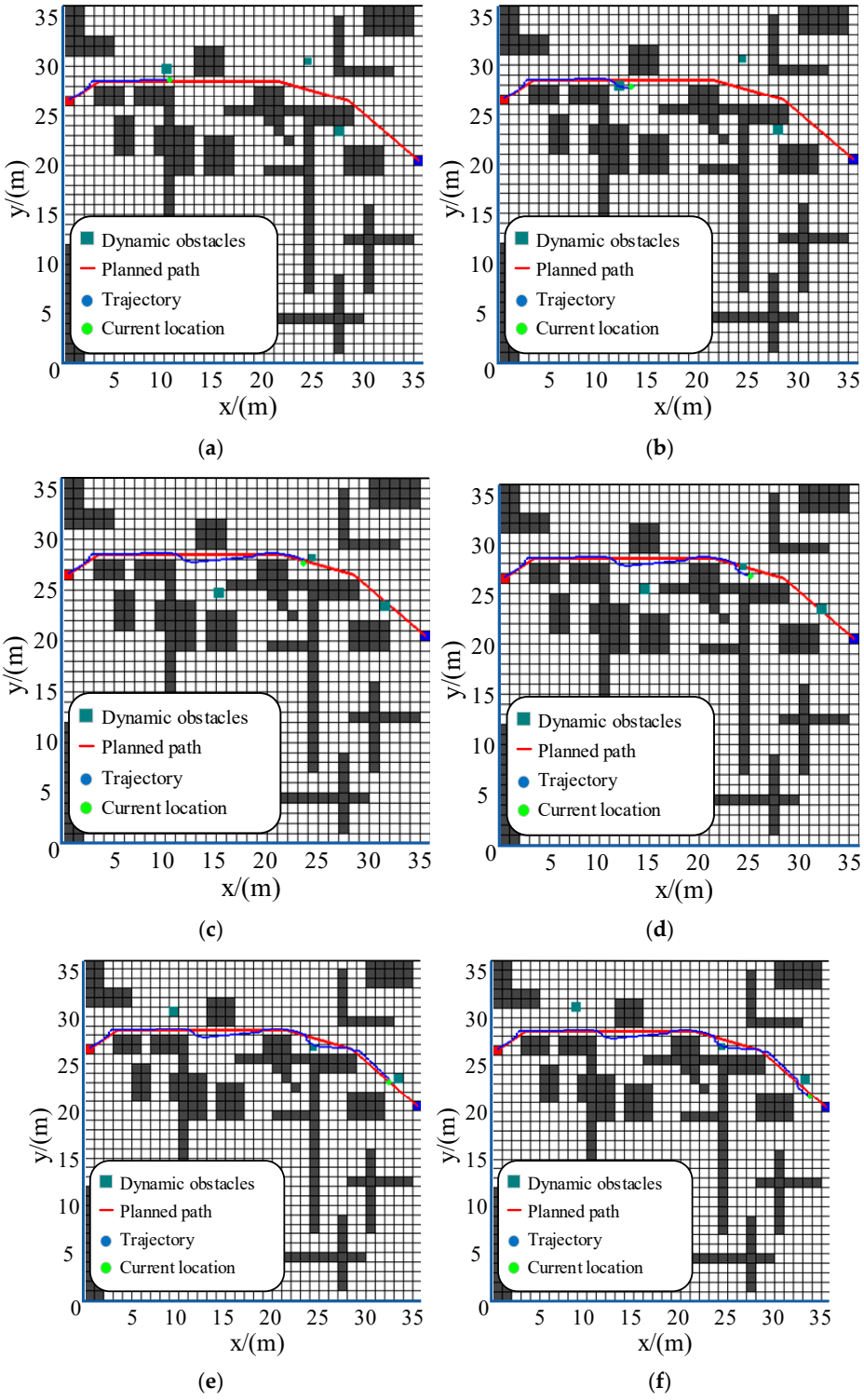


Figure 7. Fusion algorithm dynamic simulation raster graph simulation results: (**a**) the first encounter with an obstacle; (**b**) avoiding the obstacle for the first time; (**c**) the second encounter with an obstacle; (**d**) avoiding the obstacle for the second time; (**e**) the third encounter with an obstacle; (**f**) avoiding the obstacle for the third time.

5. Conclusions and Discussion

This study focused on developing an efficient path planning solution for mountain transport vehicles in intricate environments. By integrating an improved A* algorithm with the APF algorithm, significant progress was achieved in both static and dynamic path planning scenarios.

In the improvement of the A* algorithm, the newly designed heuristic function, which combines the distance-based and line-distance-based factors, and the exponential weighting strategy effectively reduced the number of searched nodes and running time. The subsequent path optimization step further streamlined the path by minimizing inflection points and shortening its length. These enhancements contribute to a more efficient global path-planning process.

For the APF algorithm, the modification of the repulsive field function by incorporating the target distance factor successfully overcame the target-unreachable issue. When integrated with the improved A* algorithm, the combined approach used the optimized inflection points from the A* algorithm as sub-target points for the APF, enabling real-time obstacle avoidance in dynamic environments.

In the simulation verification, in static environments, the integrated algorithm outperformed the traditional APF algorithm. It can plan paths around complex obstacles in mountainous terrains, and the secondary path-planning mechanism effectively resolves local-minimum problems. In dynamic obstacle-avoidance scenarios, the algorithm can accurately detect approaching dynamic obstacles and initiate timely avoidance maneuvers while maintaining a safe distance from static obstacles. This indicates its high adaptability and reliability in dynamic environments.

However, there are still areas for improvement. For example, the computational complexity of the integrated algorithm may increase in extremely large-scale and highly complex environments. Future research could explore ways to further optimize the algorithm's computational efficiency without sacrificing path-planning quality. Additionally, more real-world experimental data is needed to fully validate the algorithm's performance in diverse mountainous terrains, and we plan to conduct prototype tests in future work.

Author Contributions: Conceptualization, C.C. and L.Z.; methodology, Y.L.; software, Y.Z.; validation, L.Z., X.C. and M.C.; formal analysis, Y.H.; investigation, L.Z.; resources, Y.H.; data curation, X.C.; writing—original draft preparation, C.C. and Y.L.; writing—review and editing, Y.H.; visualization, L.Z.; supervision, Y.Z.; project administration, C.C., X.C. and M.C.; funding acquisition, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Wenzhou Tusheng Holding Group Co., Ltd., Science and Technology project (CF058807002023003).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Authors Changlong Chen, Lulin Zhan, Yi Zhang, Xiqiang Chi, and Menghu Chen are employed by the company Wenzhou Power Construction Co., Ltd. Author Yuejin Lin is employed by the company Wenzhou Tusheng Holdings Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
2. Tang, X.R.; Zhu, Y.; Jiang, X.X. Improved A-star algorithm for robot path planning in static environment. *J. Phys. Conf. Ser.* **2021**, *1792*, 012067. [CrossRef]
3. Shao, S.; Peng, Y.; He, C.; Du, Y. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Trans.* **2020**, *97*, 415–430. [CrossRef] [PubMed]
4. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [CrossRef]
5. Wenbin, C.; Qingbao, Z.; Jun, H. Path planning based on biphasic ant colony algorithm and fuzzy control in dynamic environment. In Proceedings of the 2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics, Nanjing, China, 26–28 August 2010; Volume 1, pp. 333–336.

6. Neamah, H.A.; Donát, E.; Korondi, P. Optimizing autonomous navigation in unknown environments: A novel trap avoiding vector field histogram algorithm VFH + T. *Results Eng.* **2024**, *23*, 102625. [CrossRef]
7. Neamah, H.A.; Mayorga, O.A.M. Optimized TD3 algorithm for robust autonomous navigation in crowded and dynamic human-interaction environments. *Results Eng.* **2024**, *24*, 102874. [CrossRef]
8. Zhao, Z.; Meng, Z. Path planning of service mobile robot based on adding-weight A* algorithm. *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* **2008**, *10*, 197–198.
9. Yu, X.; Jiang, C.; Duan, S.; Deng, Q. Path Planning for Improvement of A* Algorithm and Artificial Potential Field Method. *J. Syst. Simul.* **2024**, *36*, 782–794.
10. Bao, J.S.; Zhang, M.Y.; Ge, S.R.; Liu, Q.; Yuan, X.M.; Wang, M.S. Underground driverless path planning of trackless rubber tyred vehicle based on improved A* and artificial potential field algorithm. *J. China Coal Soc.* **2022**, *47*, 1347–1360.
11. Park, M.G.; Jeon, J.H.; Lee, M.C. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. In Proceedings of the 2001 IEEE International Symposium on Industrial Electronics Proceedings, Pusan, Republic of Korea, 12–16 June 2001; Volume 3, pp. 1530–1535.
12. Lee, J.; Nam, Y.; Hong, S. Random force based algorithm for local minima escape of potential field method. In Proceedings of the 2010 11th International Conference on Control Automation Robotics & Vision, Singapore, 7–10 December 2010; pp. 827–832.
13. Liu, W.; Chen, L.; Wang, R.; Wan, Y. Trajectory planning for AGV based on the improved artificial potential field-A* algorithm. *Meas. Sci. Technol.* **2024**, *35*, 096312. [CrossRef]
14. Shan, S.; Shao, J.; Zhang, H.; Xie, S.; Sun, F. Research and validation of self-driving path planning algorithm based on optimized A*-artificial potential field method. *IEEE Sens. J.* **2024**, *24*, 24708–24722. [CrossRef]
15. Wu, Z.; Liu, F.; Zhou, J. Path planning for autonomous vehicles under multi-road condition based on improved artificial potential field with A* algorithm. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2025**. [CrossRef]
16. Meng, X.; Fang, X. A ugv path planning algorithm based on improved a* with improved artificial potential field. *Electronics* **2024**, *13*, 972. [CrossRef]
17. Yang, Y.; Luo, X.; Li, W.; Liu, C.; Ye, Q.; Liang, P. AAPF*: A safer autonomous vehicle path planning algorithm based on the improved A* algorithm and APF algorithm. *Clust. Comput.* **2024**, *27*, 11393–11406. [CrossRef]
18. 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. [CrossRef]
19. Rao, J.; Xiang, C.; Xi, J.; Chen, J.; Lei, J.; Giernacki, W.; Liu, M. Path planning for dual UAVs cooperative suspension transport based on artificial potential field-A* algorithm. *Knowl.-Based Syst.* **2023**, *277*, 110797. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.