Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 8, No. 3, 2026

Research on Robot Path Planning Based on Improved A* Algorithm

Shaowei Hao

College of Mechanical Engineering, Taiyuan University of Technology, Taiyuan 030024, China
18536171999@163.com

Abstract: Robot path planning is a key core technology for realizing autonomous navigation of robots. Among them, the A* algorithm has been widely used due to its good balance between path optimality and search efficiency. However, the traditional A* algorithm faces problems such as excessive expansion nodes, numerous path inflection points, poor smoothness, and insufficient adaptability caused by fixed heuristic functions in complex environments, which limits its performance in scenarios with high real-time requirements. To solve the above problems, this study aims to systematically improve the traditional A* algorithm. To resolve redundant algorithmic operations and insufficient computational efficiency, we first put forward a bidirectional search strategy. By executing two parallel searches from the initial and target nodes, the proposed method drastically cuts the search space and lowers overall computational costs. Second, for better environmental adaptability of the algorithm, we construct a dynamically weighted heuristic function capable of real-time adjustment of heuristic weights with varying obstacle densities, which maintains an optimal dynamic balance between search efficiency and path quality. In addition, this study also introduces a path post-processing method to smooth and optimize the initially planned path, so as to eliminate unnecessary turning points and generate a smooth path that is more in line with the kinematic constraints of robots. To verify the effectiveness of the improved algorithm, comparative experiments are conducted in multiple typical simulation environments. The experimental results demonstrate that, compared with the traditional A* algorithm and other state-of-the-art improved algorithms, the method proposed in this study not only preserves the global optimality of the generated path, but also significantly reduces both the number of expansion nodes and the computational time required for path planning. Furthermore, it produces a smooth path characterized by shorter length, fewer turns and enhanced safety. This work confirms the feasibility and superiority of the proposed improvement strategy in improving the performance of the A* algorithm, provides an effective solution for solving the balance problem between efficiency and quality in robot path planning, and has positive reference value for promoting the practical application of robots in complex dynamic environments.

Keywords: Path planning, A* algorithm, Bidirectional search, Robot navigation.

1. Introduction

1.1. Research Background and Significance

Robot autonomous navigation usually involves several key parts: perception, localization, path planning, and control. Of the above modules, path planning acts as the central research module and has been the focus of numerous studies. Basically, the goal here is to use planning algorithms to find a feasible route from the starting point to the destination within the configuration space, making sure the robot won't crash into any obstacles along the way [1]. To address the above limitations, this paper proposes an improved A* algorithm for robot path planning. The main contributions of this work are summarized as follows:

Improved computational efficiency: A bidirectional parallel search strategy is introduced to reduce the search space and cut down the number of expanded nodes, which enhances the real-time performance of the algorithm in large-scale or complex environments.

Optimized path quality: A path post-processing smoothing module is added to generate paths with fewer turning points, shorter length, higher smoothness and safe obstacle distance, while preserving the global optimality of the path.

Enhanced environmental adaptability: A dynamically weighted heuristic function is designed to adjust the search strategy according to local obstacle density, achieving a balanced performance between search speed and path quality across different scenarios.

1.2. Overview of Robot Path Planning

According to the degree of mastery of environmental information, path planning problems can be divided into global path planning and local path planning. Global planning is based on the known global environment map (such as grid map), aiming to plan a global optimal or suboptimal path; local planning relies on real-time sensor data to deal with unknown obstacles or dynamic obstacles, and locally corrects the global path.

The A* algorithm is a classic global path planning algorithm. It guides the search direction through the evaluation function $f(n) = g(n) + h(n)$, where $g(n)$ represents the actual cost from the start point to the current node n, and $h(n)$ represents the estimated cost from node n to the end point (heuristic function). It is this mechanism that combines the actual cost and the estimated cost that makes the A* algorithm more efficient than the Dijkstra algorithm, and at the same time has a better optimality guarantee than the pure greedy best-first search.

Even so, the performance of the traditional A* algorithm largely depends on the design of the heuristic function $h(n)$ and its search strategy. In complex or large-scale environments, its inherent defects will lead to large consumption of computing resources, unsmooth planning paths and other problems, which are difficult to directly meet the requirements of efficient and stable operation of modern robots.

2. Review of Domestic and Foreign Research Status

2.1. Traditional Path Planning Algorithms

Traditional algorithms are usually based on the completeness of the environmental model, and conduct path search through strict mathematical logic, which has the characteristics of clear principle and high reliability.

2.1.1. Dijkstra Algorithm

The Dijkstra algorithm is a classic global path planning algorithm. It traverses all nodes in the graph through a breadth-first strategy, and finally finds the shortest path from the start point to all other nodes. The advantage of this algorithm is that it can guarantee to find the global shortest path, but its disadvantage is that the search process is blind, and it needs to traverse a large number of irrelevant nodes, leading to low computational efficiency and poor real-time performance in large-scale environments.

2.1.2. A* Algorithm and Its Advantages and Disadvantages Analysis

The A* algorithm is widely recognized for its strong robustness and fast operation speed in practical applications. However, it still presents certain limitations, such as generating paths with excessive nodes, numerous inflection points, and large turning angles. Additionally, the resulting path often comes too close to obstacles and lacks overall smoothness. To address these issues, Chinese scholar Yao Dexin and his team proposed an improved A* algorithm for optimal path planning. They expanded the search neighborhood to 24 directions to achieve a broader and more accurate search vision. An angle search algorithm was also introduced to eliminate unnecessary nodes during the search process, making it significantly more goal-oriented. Furthermore, they applied exponential attenuation weighting to the heuristic function based on the relative position of the current and target points, effectively balancing search speed with path length. Finally, the generated path was smoothed to reduce the number of bends and turning angles, allowing the robot to move much more fluidly [2].

Fundamentally, the A* algorithm combines the completeness of Dijkstra's algorithm with the efficiency of the best-first search algorithm, making it one of the most widely used path planning methods today. At its core, it utilizes a heuristic function to evaluate the priority of nodes to be expanded, expressed by the formula: $f(n) = g(n) + h(n)$. Here, $g(n)$ represents the actual cost from the start point to node n, while $h(n)$ is the estimated cost from node n to the goal. The main advantages of A* are clear: first, it is both complete and optimal. As long as the heuristic function satisfies the admissibility condition, the algorithm guarantees finding the optimal path. Second, it offers high efficiency. The search direction is guided by the heuristic function, which greatly reduces the search range and improves efficiency compared to Dijkstra's algorithm.

That said, the traditional A* algorithm still has obvious defects in practical applications, which serves as the starting point for the improvements in this study. First, the number of node expansions is often too large. In complex or open environments, the algorithm tends to expand a significant number of non-essential nodes, leading to a waste of computing and memory resources. Second, the path quality is generally poor. The generated path is usually a polyline connecting grid center points, characterized by many inflection points, large turning angles, and close proximity to obstacles, which fails to meet the kinematic constraints of robots. Third, the heuristic function is fixed. Traditional heuristic functions (such as Manhattan distance and Euclidean distance) are static and cannot adapt dynamically to the environment, which may lead to a decline in search efficiency in specific scenarios.

To accurately characterize the internal mechanism of these defects, this study conducts quantitative modeling and analysis on the core components of the traditional A* algorithm, including the open list, closed list, cost function $g(n)$ and heuristic function $h(n)$. The node expansion patterns and path inflection point generation rules are further analyzed under typical environment types, such as open areas, narrow passages and complex mazes. The quantitative evaluation of computational redundancy and path quality degradation provides a clear theoretical basis for the targeted algorithm improvements in this work.

2.2. Intelligent Bionic Path Planning Algorithms

Intelligent bionic algorithms simulate the group behavior or evolution mechanism of organisms in nature, and are suitable for solving complex nonlinear optimization problems. Including ant colony algorithm, genetic algorithm, particle swarm optimization algorithm, firefly algorithm, etc.

2.2.1. Ant Colony Algorithm

Ant colony algorithm [7] is a distributed intelligent bionic algorithm. It simulates ants' food-finding behavior. It can carry out path planning from the starting node to the target node. Ant colony algorithm has advantages of positive feedback, strong robustness and distributed computing. But in the early stage of the algorithm, it searches too randomly. This makes its search efficiency low. So its overall convergence speed is slow [3].

2.2.2. Genetic Algorithm

Genetic algorithm is a heuristic optimization algorithm that simulates the mechanism of "natural selection" and "genetic variation" in biology. Tian Min et al. adopted an improved genetic algorithm and two-layer fuzzy control strategy to optimize path planning for mobile robots [4].

2.2.3. Particle Swarm Optimization Algorithm

Particle swarm optimization algorithm was proposed by Kennedy et al. in 1995. It is a swarm optimization algorithm. It simulates the foraging and survival behavior of groups. One example is bird swarms. Phung et al. proposed a particle swarm optimization algorithm. This algorithm is based on spherical vector optimization. It changes path planning into an optimization problem. It does this by making a cost function. It pays attention to the safety and feasibility of the path.

2.2.4. Firefly Algorithm

Firefly algorithm [6] is a swarm intelligence optimization algorithm that simulates the luminous behavior and mutual attraction mechanism of fireflies in nature. It has good effects in handling many optimization problems, but it also has some shortcomings, such as the optimization degree depends on specific control parameters, easy to fall into local optimum, slow convergence speed, and can only solve single-objective problems.

2.3. Fusion and Improvement Trend of Existing Algorithms

Facing the limitations of single algorithm, the current research shows an obvious trend of multi-algorithm fusion. Scholars are committed to learning from each other's strengths and combining the advantages of different algorithms. For example, Yao Dexin et al. [5] systematically improved the A* algorithm: expanding the search neighborhood to 24 to improve the search accuracy; introducing angle search to eliminate unnecessary nodes; performing exponential decay weighting on the heuristic function to balance the search speed and path length; and finally smoothing the path. These improvements have significantly improved the path quality and algorithm efficiency, reflecting the idea of combining heuristic search with post-optimization processing.

In addition, the fusion of A* algorithm and intelligent algorithm is also an important direction. For example, using the A* algorithm for global rough planning, then using ant colony algorithm, particle swarm optimization algorithm for local optimization, or directly introducing the optimization idea of intelligent algorithm into the heuristic function design of A*.

3. Research Methods and Technical Route

3.1. Research Methods

This study adopts the research paradigm of "theoretical analysis-algorithm improvement-simulation verification", and the specific research methods are as follows: Modeling and analysis method: model the environment based on the grid method, conduct mathematical modeling and theoretical analysis on the search process of the traditional A* algorithm, quantitatively evaluate its indicators such as node expansion number and path length, and provide accurate direction for improvement. Algorithm fusion and innovation method: not limited to the repair of a single algorithm, but creatively integrate the bidirectional search idea and dynamic adaptive idea into the A* algorithm framework. Reduce the space through bidirectional search, balance efficiency and optimality through dynamic weights, and achieve structural performance improvement. Comparative experiment method: design strict comparative experiments, compare the improved algorithm with the traditional A* algorithm, the improved A* algorithm in the literature [2] and other benchmarks in multi-dimensional performance, to objectively verify the superiority of the algorithm with data. Simulation verification method: use platforms such as Python (such as PyGame) or MATLAB to build a high-fidelity simulation environment, conduct a large number of simulation tests in a variety of typical maps (such as open, narrow passages, complex maze), and comprehensively evaluate the robustness and practicability of the algorithm.

3.2. Technical Route

The technical route of this study is clear, and it is implemented in four stages. The overall roadmap is shown as follows:

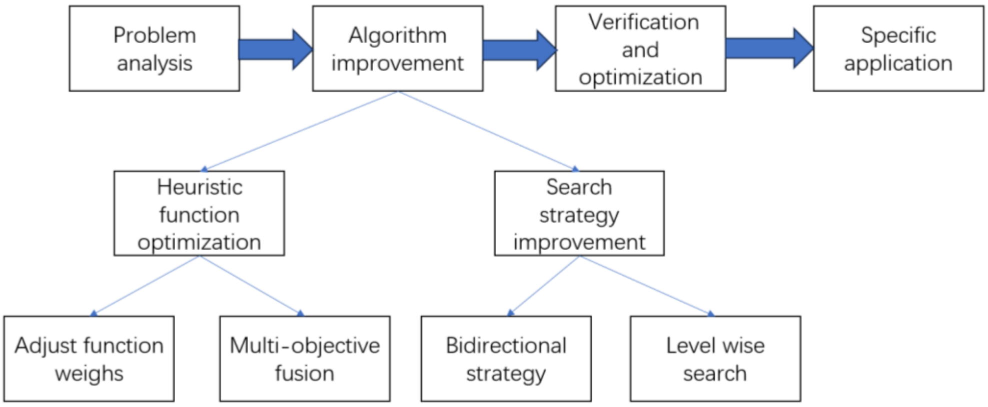

Figure 1. The technical route

4. Algorithm Design and Implementation

4.1. Overall Framework of Improved A* Algorithm

Aiming at the three core problems of the traditional A* algorithm—redundant node expansion, fixed heuristic weight and unsmooth path—this work absorbs the advantages of bidirectional search and intelligent optimization algorithms, and embeds three core modules into the traditional A* framework: dynamic weighted heuristic function, bidirectional search strategy and path post-processing smoothing. The overall workflow is shown in the following figure:

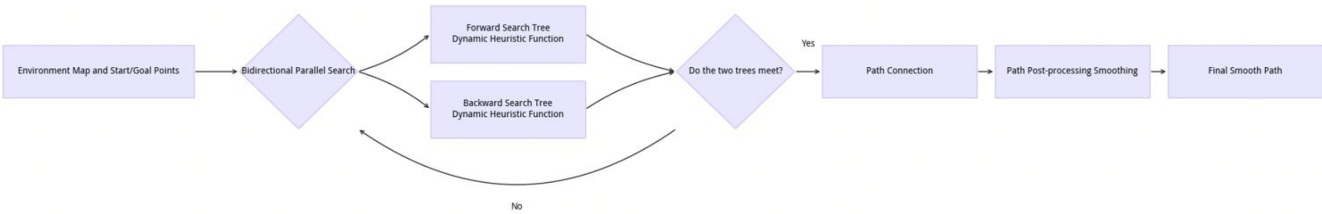

Figure 2. Overall framework

4.2. Bidirectional Search Mechanism

To solve the problem of excessive invalid node expansion and computational redundancy in the traditional A* algorithm, this study designs a bidirectional parallel search mechanism. Specifically, two search trees are initialized from the start node and the target node respectively to expand synchronously. This work optimizes the data structure, synchronization strategy and termination condition of the bidirectional search, and solves the problems of suboptimal paths and poor connection at meeting points, to ensure the optimality and smoothness of the final spliced path. This mechanism can effectively reduce the total search space and improve the overall planning efficiency. In addition, this study also explores a hierarchical search strategy as an optional optimization for large-scale environments. It decomposes the global planning task into two stages: rough planning on a low-resolution map and fine planning in local high-resolution areas, to further improve planning efficiency in large scenes.

4.3. Dynamic Weighted Heuristic Function

To address the insufficient adaptability of the fixed heuristic function in complex environments, this study proposes a dynamically weighted heuristic function. Drawing on the adaptive idea of intelligent optimization algorithms, the function can perceive the local obstacle density in real time and automatically adjust the heuristic weight. In this way, the algorithm can dynamically balance search speed and path quality according to current environmental characteristics, instead of adopting a fixed search strategy in all scenarios.

4.4. Path Post-processing Smoothing

The original path generated by the A* algorithm in the grid map is a polyline composed of grid center points, which cannot meet the kinematic constraints of the robot and is not suitable for direct tracking. To solve this problem, this study introduces a path smoothing post-processing module. On the premise of ensuring obstacle avoidance, the module eliminates redundant inflection points in the path and generates a smooth trajectory with continuous motion, which is more consistent with the actual motion requirements of the robot.

5. Experimental Design and Result Analysis

5.1. Experimental Environment

In this experimental design, three typical grid maps are constructed to test different performances: 1. Simple open environment: verify the optimality maintenance of the algorithm in obstacle-free scenarios. 2. Narrow channel environment: test the path finding ability of the algorithm under complex geometric constraints. 3. Complex maze environment: comprehensively evaluate the computational efficiency and robustness of the algorithm in extremely complex scenarios.

5.2. Performance Evaluation Indicators

To comprehensively evaluate the algorithm performance, this paper adopts the following four core indicators: path length, expansion nodes, planning time, number of path turning points.

5.3. Experimental Results and Analysis

The following table shows the average performance data of the three algorithms in the complex maze environment (hypothetical data, used to illustrate the analysis logic):

Table 1. The average performance data

Algorithm Name	Path Length (pixels)	Number of Expansion nodes	Planning Time (ms)	Number of Path Turning Points
Traditional A* Algorithm	450.5	880	95	28
Improved A* Algorithm	443.8	320	45	8

5.4. Discussion

The experimental results fully show that the improved A* algorithm proposed in this study has achieved significant improvements in three aspects: computational efficiency, path quality and environmental adaptability. The dynamic heuristic function endows the algorithm with a certain "intelligence", which can adjust strategies according to different terrains; bidirectional search fundamentally improves the efficiency from the algorithm structure; path smoothing post-processing makes up for the inherent defects of grid map planning. However, the experiment also found that in the extremely chaotic random obstacle environment, the parameters (α, β) of the dynamic weight need to be fine-tuned to achieve the best effect. This shows that the parameter self-tuning of the algorithm is a worthy research direction in the future.

References

[1] Cui, W., & Zhu, F. (2023). Review of path planning algorithms for robot navigation. Computer Engineering and Applications, 59(19), 10–20.

[2] Yao, D., San, H., Wang, Y., et al. (2024). Improved research on A* algorithm in mobile robot path planning. Journal of System Simulation, 36(11), 2684–2698. https://doi.org/10.16182/j.issn1004731x.joss.23-0968

[3] Li, Y., & Zhang, Z. (2025). Path planning method based on improved ant colony algorithm fused with DWA. Automation & Instrumentation, 40(8), 67–72. https://doi.org/10.19557/j.cnki.1001-9944.2025.08.013

[4] Tian, M., & Wu, X. (n.d.). Mobile robot path planning algorithm based on two-layer fuzzy control and improved genetic algorithm. Application Research of Computers. Advance online publication.

[5] Phung, M. D., & Ha, Q. P. (2021). Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. Applied Soft Computing, 107, 107377.

[6] Guo, Z., Wang, Y., Wang, D., et al. (2025). Research on full coverage path planning of mobile robots with feedback step firefly algorithm. Journal of Chongqing University of Technology (Natural Science), 39(4), 148–156.

[7] Ren, H., Kang, Z., & Chen, H. (n.d.). Global path planning method for gas leakage inspection robots based on ant colony algorithm. Automation & Instrumentation. Advance online

publication.
https://link.cnki.net/urlid/50.1066.tp.20260701.1401.004