

# 基于全局引导力的分数阶人工势场的路径规划算法研究

曹志成, 杨旗\*

沈阳理工大学机械工程学院, 辽宁 沈阳

收稿日期: 2025年12月7日; 录用日期: 2025年12月30日; 发布日期: 2026年1月7日

## 摘要

本文针对传统A\*算法在路径规划中存在扩展节点多、搜索时间长、路径转折频繁及易陷入局部最优等问题, 提出一种融合全局引导的分数阶人工势场法(APF)的改进A\*路径规划算法。该算法通过引入分数阶微积分增强势场力的连续性与记忆性, 并结合全局路径趋势引导力, 动态约束搜索邻域, 减少冗余搜索方向。在MATLAB平台上进行仿真实验, 结果表明: 在 $30 \times 30$ 和 $50 \times 50$ 栅格地图中, 改进算法相较于传统A\*算法, 搜索时间分别减少38.67%和47.31%, 扩展节点数降低22.97%和33.46%, 路径转折次数减少15.38%和18.18%, 路径长度缩短1.94%和2.95%。实验验证了所提算法在复杂环境中具有更高的规划效率和路径质量。

## 关键词

分数阶, 人工势场法, A\*算法, 路径规划, 改进人工势场法, MATLAB, 全局引导, 局部极小值

# Research on Path Planning Algorithm Based on Fractional-Order Artificial Potential Field with Global Guiding Force

Zhicheng Cao, Qi Yang\*

School of Mechanical Engineering, Shenyang Ligong University, Shenyang Liaoning

Received: December 7, 2025; accepted: December 30, 2025; published: January 7, 2026

## Abstract

To address the issues of excessively expanded nodes, long search time, frequent path turns, and

\*通讯作者。

susceptibility to local optima in traditional A\* algorithm-based path planning, this paper proposes an improved A\* path planning algorithm that integrates globally-guided fractional-order artificial potential field (APF). By introducing fractional calculus to enhance the continuity and memory of potential field forces, and incorporating global path trend guidance to dynamically constrain the search neighborhood, the algorithm effectively reduces redundant search directions. Simulation experiments conducted on the MATLAB platform show that in  $30 \times 30$  and  $50 \times 50$  grid maps, the improved algorithm reduces search time by 38.67% and 47.31%, decreases the number of expanded nodes by 22.97% and 33.46%, and reduces the number of path turns by 15.38% and 18.18%, and shortens the path length by 1.94% and 2.95%, respectively, compared to the traditional A\* algorithm. The results verify that the proposed algorithm achieves higher planning efficiency and path quality in complex environments.

## Keywords

Fractional Order, Artificial Potential Field (APF), A\* Algorithm, Path Planning, Improved Artificial Potential Field, MATLAB, Global Guidance, Local Minimum

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着科技的进步和社会的不断发展，自主移动机器人在工业、农业、医学、航空航天等许多领域发挥日益重要的作用，广为人知的无人车、无人机和水下机器人等越来越多地进入了人们的生活。在移动机器人相关技术研究中，路径规划是一个重要的环节和组成部分。所谓的路径规划就是指机器人根据各种传感器对周围环境进行感知，自主探索出一条从起点到目标点的无碰撞路径[1]。该路径应满足安全、光滑、高效等一系列要求，并且能够避开沿途的静态和动态障碍物。良好的移动机器人路径规划技术不仅可以节省大量的时间，还可以减少移动机器人的磨损和资本投资[2]。

根据环境信息是否先前已知和规划的范围，路径规划可以分为全局路径规划和局部路径规划[3]。全局路径规划算法被广泛应用于解决在已知的静态环境中的路径规划问题，常见的算法有 Dijkstra 算法、A\*算法、RRT 算法、遗传算法、蚁群算法等；局部路径规划算法常被应用在动态的未知或部分已知环境中的路径规划问题，常见的算法有动态窗口法、人工势场法和时间弹性带等[4]。

以上路径规划算法各有优缺点，在全局路径的规划算法中，A\*算法使用较为广泛，具有简单且速度相对较快[5]、稳定性较高等优点。但也存在实时性差、扩展节点多、计算量大、路径不平滑、结果不一定为最优解等问题。迟旭等[6]将 A\*算法原本固定的 8 个搜索方向改进为 5 个搜索方向，提高了搜索效率，但是可能错过一些更优节点；彭哲等[7]对评价函数进行优化，虽然搜索效率提升了许多，但搜索出的路径经常不是最优解。

针对 A\*算法的局限性，本文结合改进分数阶人工势场法优化搜索邻域，减少了搜索邻域的数量，解决了易陷入 U 型陷阱的问题；改进后的 A\*算法大幅提高了路径规划效率和质量，进一步加强了 A\*算法的实用性。

## 2. 传统 A\*算法

A\*算法在保证搜索最短路径的情况下，通过综合考虑节点到起始点的实际代价和对目标点的估计代

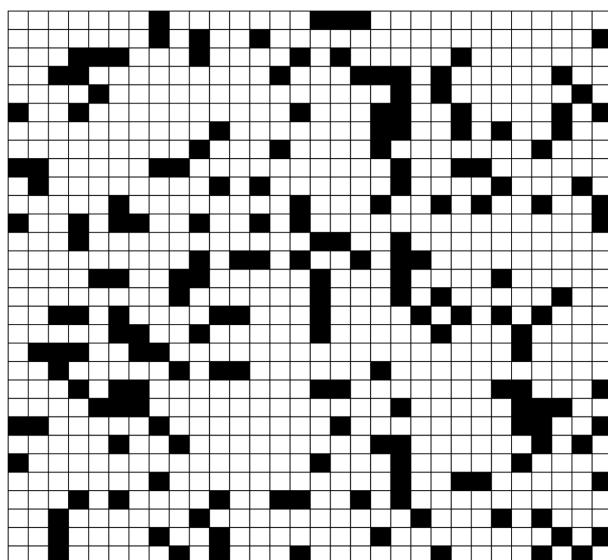
价，引入了启发函数的概念，提高路径搜索效率。其实现过程可分为场景建模与路径规划两个部分。

## 2.1. 场景建模

在 A\*算法中，场景建模模块扮演着至关重要的角色，它能够将实际场景转化为易于计算机读取的栅格地图，从而为路径规划提供了坚实的基础。场景建模方法有栅格法、拓扑法、可视图法和单元分解法等。与其他建模方法相比，栅格法在操作上更为简单且容易实现，因而被广泛应用于描述机器人环境地图[8]。

## 2.2. 储存系统组成及流程

在建立地图时，根据实际场景栅格法将环境划分为规则的网格，并在每个网格单元中均存储相应的信息，每一个栅格代表一个节点，这种简单的结构使得栅格法能够更加方便地对相应地图上的区域进行处理与计算。如图 1 所示，白色栅格表示可通行，黑色栅格对应障碍物。映射标记完成后，就可以开始进行路径规划了。



**Figure 1.** Grid map

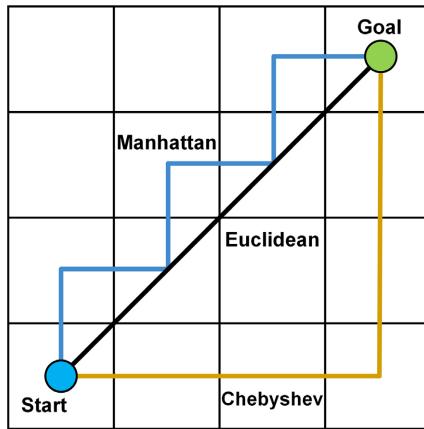
**图 1.** 栅格地图

## 2.3. 系统的控制要求

A\*算法作为 Dijkstra 算法的扩展，继承了 Dijkstra 算法的优点，具有完备性和最优性。同时，A\*算法又类似于贪婪最佳优先搜索算法，使用启发式函数引导搜索[9]。传统的 A\*算法常用的节点扩展方式一般为八邻域，利用评价函数来评估当前节点周围的八个节点到目标点的最短距离，并依据这个评估值来选择接下来要搜索的节点，通过综合考虑实际距离和预估距离，最大限度地提高最短路径的寻找速度[10]。利用估价函数对计算空间中每个节点的总代价进行计算。估价函数如式(1)所示：

$$F(n) = G(n) + H(n) \quad (1)$$

式中： $n$  为当前节点； $G(n)$  为起点到节点  $n$  的实际代价， $H(n)$  为节点  $n$  到目标点的启发式预估代价， $F(n)$  为节点  $n$  的总代价。启发函数  $H(n)$  常见的计算方式有曼哈顿距离、切比雪夫距离和欧氏距离，如图 2 所示。



**Figure 2.** Schematic diagram of three types of heuristic function calculation methods  
**图 2.** 三种启发函数计算方式示意图

### 3. 提出基于全局引导的分数阶人工势场 A\*算法

#### 3.1. 融合势场法优化搜索邻域

当移动机器人在栅格地图上进行搜索时,通常会采用 $3\times 3$ 邻域搜索方式,即以当前节点为中心,从周围的8个节点中挑选出可供选择的备选节点,然后从中筛选出下一步将进行扩展的节点[11]。受局部路径规划人工势场法的启发,可以将目标点对机器人产生的引力和障碍物对机器人产生的斥力合成一个引导力,根据引导力的方向来确定保留的5个搜索方向和舍弃的3个搜索方向。

##### 3.1.1. 分数阶人工势场法的推广

人工势场法是在移动机器人路径规划避障中广泛应用的方法,基本思想是引入人工势场的概念,将势场分为引力势场和斥力势场[11],人工势场定义为:

$$U = U_{att} + U_{rep} \quad (2)$$

其中引力势场函数定义为:

$$U_{att} = \frac{1}{2} K \|R - G_g\|^2 \quad (3)$$

式中,  $K$  为引力势场增益系数,  $\|R - G_g\|^2$  为机器人到目标点的欧氏距离。机器人所受引力为:

$$F_{att} = -\nabla(U_{att}) = -K \|R - G_g\| \quad (4)$$

斥力势场函数定义为:

$$U_{rep} = \begin{cases} \frac{1}{2} q \left( \frac{1}{d} - \frac{1}{d_0} \right)^2, & d \leq d_0 \\ 0, & d > d_0 \end{cases} \quad (5)$$

式中,  $q$  为斥力势场增益系数;  $d$  为当前位置与障碍物之间的距离;  $d_0$  为常数,表示障碍物影响距离。机器人所受斥力为:

$$F_{rep} = -\nabla(U_{rep}) = \begin{cases} q \left( \frac{1}{d} - \frac{1}{d_0} \right) \frac{1}{d}, & d \leq d_0 \\ 0, & d > d_0 \end{cases} \quad (6)$$

然而, 传统整数阶 APF 仅仅是一种局部瞬时的描述, 无法刻画势场力的历史累积效应与机器人的运动惯性。针对这一问题, 本文将 APF 算法在阶次上从整数阶推广至分数阶, 分数阶算子具有记忆性和遗传性, 能够更精细地描述势场的渐变过程, 从而使势场力的变化更加平滑连续, 有效克服传统 APF 的缺陷。

引入了分数阶微积分后的 APF 算法, 引力势场重新定义为:

$$U_{att}(q) = \frac{1}{2} k_{att} \cdot \left( D^\beta [d(q, q_{goal})] \right) \quad (7)$$

式中,  $D^\beta$  表示 Caputo 型分数阶微分算子;  $\beta (0 < \beta < 2)$  为引力场的分数阶阶数;  $d(q, q_{goal})$  为当前点到目标点的欧氏距离。机器人所受分数阶引力为该势场的负梯度:

$$F_{att} = -\nabla U_{att} = -k_{att} \cdot D^\beta [d(q, q_{goal})] \quad (8)$$

斥力势场为:

$$U_{req} = \begin{cases} \frac{1}{2} k_{req} \cdot \left( D^\alpha \left[ \frac{1}{d(q, q_{obs})} - \frac{1}{\rho} \right] \right)^2 & d(q, q_{obs}) \leq \rho \\ 0 & d(q, q_{obs}) > \rho \end{cases} \quad (9)$$

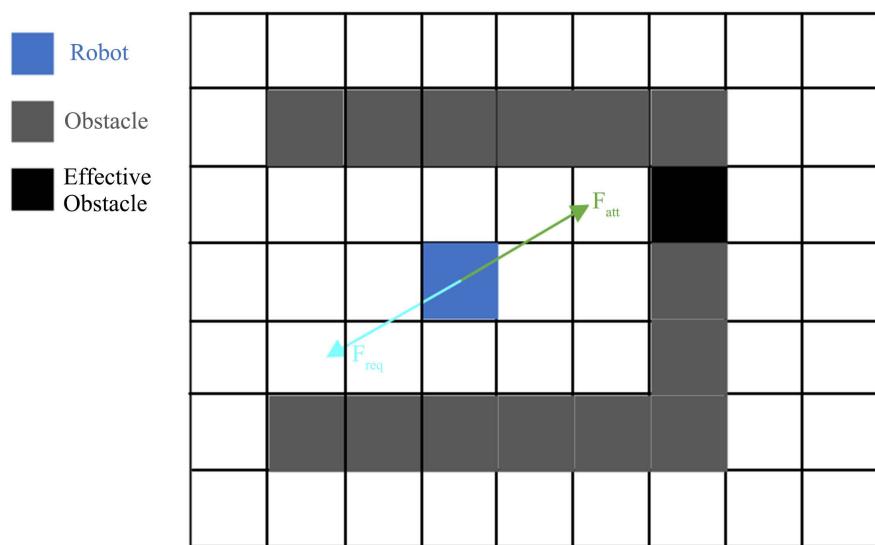
式中,  $\alpha$  为斥力场的分数阶阶数;  $d(q, q_{obs})$  为当前点到障碍物的距离;  $\rho$  为障碍物影响距离。对应的分数阶斥力为:

$$F_{req} = -\nabla U_{req} \quad (10)$$

通过调节分数阶阶数( $\beta, \alpha$ ), 可以更精细地控制引力与斥力随距离变化的非线性关系, 从而为不同场景下的路径规划提供了更灵活、更强大的参数优化空间, 从根本上增强了模型的适应性和鲁棒性。

### 3.1.2. 提出引入全局路径趋势引导力的 APF 改进

传统 APF 算法在面对 U 型障碍物环境时, 引力与斥力可能在非目标点处达到平衡, 即  $F_{att} + F_{req} = 0$ , 导致机器人陷入局部最优点, 无法脱身(图 3)。



**Figure 3.** Trapped in local optimum under traditional APF

图 3. 传统 APF 下陷入局部最优

针对这一问题, 本文提出了一种结合全局路径趋势信息的优化势场法。我们在改进的 APF 算法中引入了全局路径趋势引导力, 使得机器人不仅受到目标点的引力和障碍物的斥力影响, 还能够根据全局路径的方向来调整搜索方向。

改进后的势场法将目标点的引力、障碍物的斥力和全局路径趋势的引导力综合在一起, 形成总的引导力  $F$ , 使得机器人能够沿着更合理的路径前进。具体而言, 目标点的引力、障碍物的斥力的计算同上一小节, 全局路径的引导力定义其方向为从当前节点指向全局路径参考方向, 大小与当前节点偏离全局路径的程度相关计算公式如下:

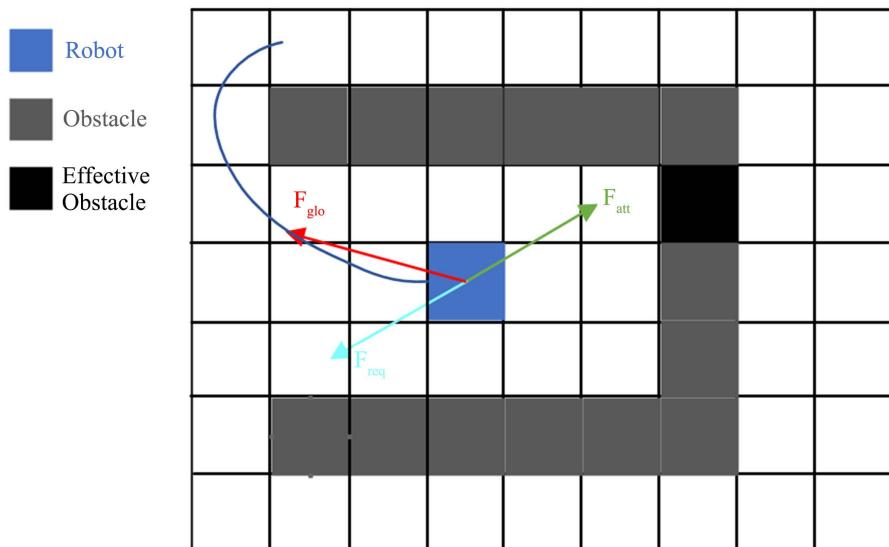
$$F_{glo} = k_g \cdot \frac{P_{goal} - P_{start}}{\|P_{goal} - P_{start}\|} \quad (11)$$

其中  $P_{goal}$ 、 $P_{start}$  分别为目标点位置和起点位置,  $k_g$  为全局引导力增益系数, 用于调节全局趋势的影响程度。

此时当前节点受到的合力为:

$$F = F_{att} + F_{req} + F_{glo} \quad (12)$$

其中  $F_{att}$  和  $F_{req}$  为传统的引力和斥力势场,  $F_{glo}$  为全局路径趋势引导力。通过综合这些引导力, 机器人能够在复杂环境中更有效地找到最优路径, 避免了局部最优解的产生(图 4)。



**Figure 4.** Improved APF algorithm avoids local minimum  
**图 4.** 改进 APF 算法避免局部最小

为了证明该改进算法的有效性, 我们首先考虑优化后的路径规划算法的收敛性。假设机器人在路径规划过程中不断调整其位置以满足目标点引力、障碍物斥力和全局引导力的作用。我们引入 Lyapunov 稳定性理论来证明算法的收敛性。

$$V(P_{current}) = \frac{1}{2} \|P_{current} - P_{goal}\|^2 \quad (13)$$

对其进行求导得:

$$\dot{V} = (P_{current} - P_{goal})^T \cdot \dot{P}_{current} \quad (14)$$

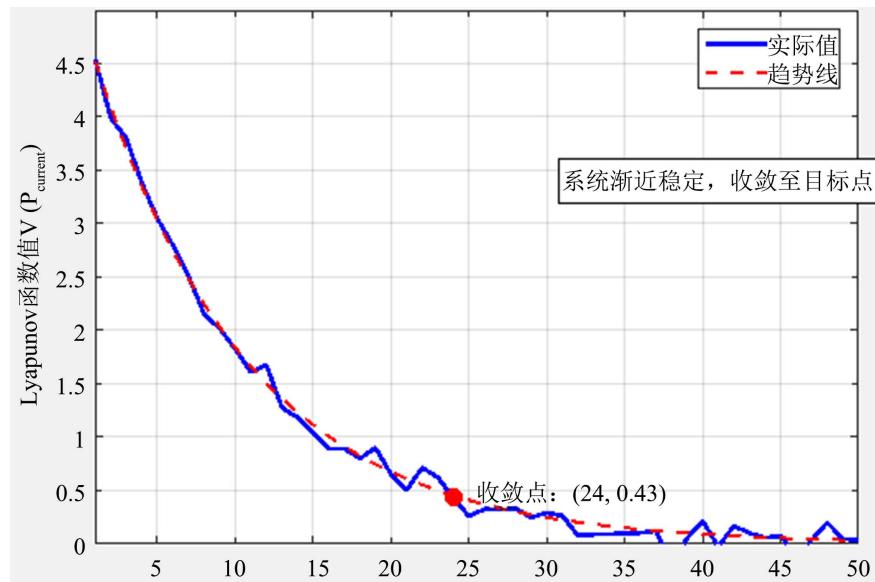
假设机器人运动方向与总引导力方向一致, 即:

$$P_{current} = \alpha \cdot F \quad (15)$$

代入得:

$$\dot{V} = \alpha (P_{current} - P_{goal})^T \cdot (F_{att} + F_{req} + F_{glo}) \quad (16)$$

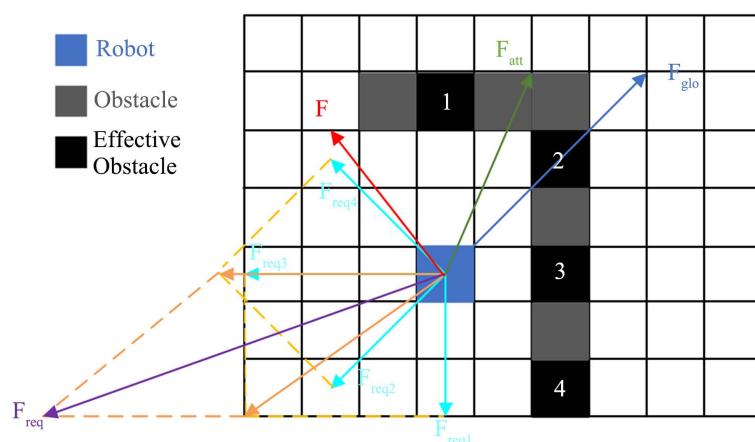
由于  $F_{att}$  指向目标点,  $F_{glo}$  也朝向全局方向, 合力方向总体朝向目标点, 因此  $\dot{V} < 0$ , 系统渐近稳定, 机器人最终收敛至目标点(图 5)。



**Figure 5.** Lyapunov function convergence  
**图 5.** Lyapunov 函数收敛

### 3.1.3 改进 APF 后的邻域搜索

融合改进 APF 算法来确定搜索邻域的准确性。如图 6 所示, 利用势场法计算出的  $F$  就是当前节点的受到的引导力, 再通过引导力的方向选择出下一步要搜索的 5 个邻域。

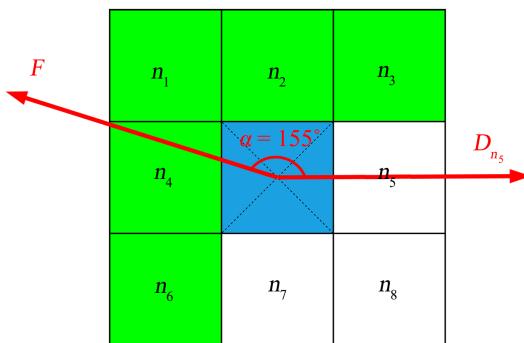


**Figure 6.** Guiding force calculation based on potential field method  
**图 6.** 势场法求引导力

为了减小计算量, 我们只考虑当前节点的 8 邻域方向是否有障碍物, 设引导力  $F$  与  $n_5$  方向的夹角为  $\alpha$ , 夹角  $\alpha$  与搜索邻域的关系如表 1 所示。根据图 7 所示,  $\alpha = 155^\circ$ , 所以当前节点需要搜索的邻域是  $n_1$ 、 $n_2$ 、 $n_3$ 、 $n_4$ 、 $n_6$ 。

**Table 1.** Comparison table of MPC and traditional PID**表 1.** 搜索方向与夹角  $\alpha$  的关系

$\alpha$	保留的 5 个方向	舍弃的 3 个方向
$-22.5^\circ \leq \alpha < 22.5^\circ$	$n_2 n_3 n_5 n_7 n_8$	$n_1 n_4 n_6$
$22.5^\circ \leq \alpha < 67.5^\circ$	$n_1 n_2 n_3 n_5 n_8$	$n_4 n_6 n_7$
$67.5^\circ \leq \alpha < 112.5^\circ$	$n_1 n_2 n_3 n_4 n_5$	$n_6 n_7 n_8$
$112.5^\circ \leq \alpha < 157.5^\circ$	$n_1 n_2 n_3 n_4 n_6$	$n_5 n_7 n_8$
$157.5^\circ \leq \alpha$ 或 $\alpha < -157^\circ$	$n_1 n_2 n_4 n_6 n_7$	$n_3 n_5 n_8$
$-157.5^\circ \leq \alpha < -112.5^\circ$	$n_1 n_4 n_6 n_7 n_8$	$n_2 n_3 n_5$
$-112.5^\circ \leq \alpha < -67.5^\circ$	$n_4 n_5 n_6 n_7 n_8$	$n_1 n_2 n_3$
$-67.5^\circ \leq \alpha < -22.5^\circ$	$n_3 n_5 n_6 n_7 n_8$	$n_1 n_2 n_4$

**Figure 7.** The included angle between guiding force and  $n_5$   
**图 7.** 引导力与  $n_5$  的夹角

因此,  $\alpha$  取不同值时对应的保留和舍弃方向如表 2 所示。

#### 4. 仿真实验与分析

为了验证改进方法的有效性, 本文使用 MatlabR2022b 平台, 在  $30 * 30$  和  $50 * 50$  大小的地图上进行了算法验证。分别对 Dijkstra 算法、传统 A\*算法、文献[5]邻域改进方法、文献[6]权重调整方法和本文改进的 A\*算法进行实验和分析, 对比了在两地图下的搜索时间、扩展节点数、转折次数和路径长度。实验数据如表 2 所示, 关键数据的对比图如图 8 所示。

**Table 2.** Comparison of experimental data  
**表 2.** 实验数据对比

地图	算法	搜索时间(s)	扩展节点数	转折次数	路径长度
$30 * 30$	Dijkstra 算法	13.2259	807	10	47.6985
	传统 A*算法	2.5340	209	11	47.6985

续表

	文献[5]算法	2.1729	191	11	47.6985
	文献[6]算法	1.5706	<b>159</b>	14	49.3553
	本文算法	<b>1.5542</b>	161	<b>9</b>	<b>46.7731</b>
	Dijkstra 算法	40.0258	2228	26	98.7696
	传统 A*算法	12.8468	813	26	98.7696
50 * 50	文献[5]算法	11.5177	736	26	98.7696
	文献[6]算法	7.0954	<b>507</b>	36	111.1960
	本文算法	<b>6.7689</b>	541	<b>22</b>	<b>95.8513</b>

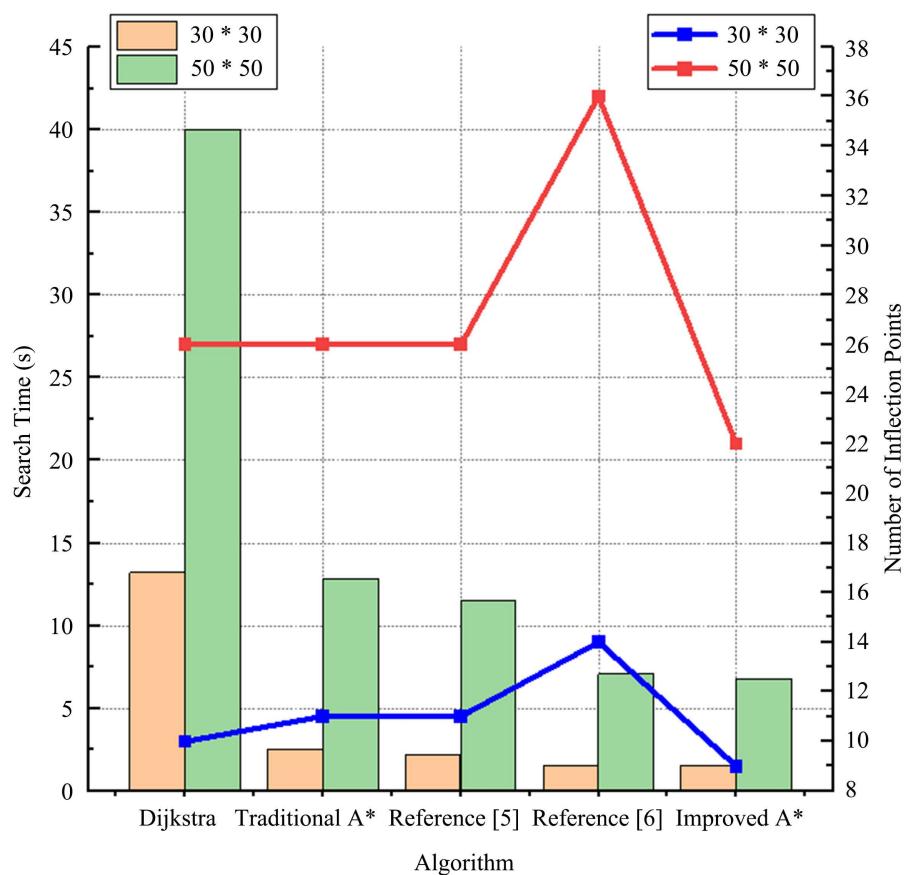
**Figure 8.** Comparison chart of experimental data

图 8. 数据对比图

从实验结果可以看出, 相对于另外四种算法, 本文改进的 A\*算法搜索时间、扩展节点数、转折次数、和路径长度均得到有效减少, 效果较为理想。与传统 A\*算法相比, 本文改进的 A\*算法搜索时间分别缩短了 38.67% 和 47.31%, 扩展节点数分别减少了 22.97% 和 33.46%, 转折次数分别减少了 15.38% 和 18.18%, 路径长度分别缩短了 1.94% 和 2.95%, 可以看出场景越复杂, 地图越大, 算法改进的效果会越明显。由此得出实验结论, 本文改进后的 A\*算法, 有效提高了算法的搜索效率和路径平滑度, 降低了算法运行所需内存和路径长度。

## 5. 结论

本文针对传统 A\*算法计算量大、时间长、路径不平滑、结果不一定为最优解等问题, 进行分析并提出改进方法。对搜索方向进行改进, 结合势场法, 考虑当前节点的位置和周围障碍物的分布, 减少了搜索邻域的数量; 对预估代价函数进行改进, 使估价函数自适应调整权重; 对搜索线程进行改进, 改进传统双向搜索策略, 提高计算效率; 最后删除冗余节点并利用贝塞尔曲线对路径进行平滑处理。利用 MATLAB 进行仿真, 通过分析数据和实验结果, 验证了本文改进的 A\*算法, 在保证路径最短的同时, 缩短了路径规划时间, 减少了转弯次数, 提升了路径平滑度, 降低了内存消耗。

## 参考文献

- [1] 刘志荣, 姜树海. 基于强化学习的移动机器人路径规划研究综述[J]. 制造业自动化, 2019, 41(3): 90-92.
- [2] 林韩熙, 向丹, 欧阳剑, 等. 移动机器人路径规划算法的研究综述[J]. 计算机工程与应用, 2021, 57(18): 38-48.
- [3] Wang, Z., Li, G. and Ren, J. (2021) Dynamic Path Planning for Unmanned Surface Vehicle in Complex Offshore Areas Based on Hybrid Algorithm. *Computer Communications*, **166**, 49-56. <https://doi.org/10.1016/j.comcom.2020.11.012>
- [4] Li, F., Du, Y. and Jia, K. (2022) Path Planning and Smoothing of Mobile Robot Based on Improved Artificial Fish Swarm Algorithm. *Scientific Reports*, **12**, Article No. 659. <https://doi.org/10.1038/s41598-021-04506-y>
- [5] Erke, S., Bin, D., Yiming, N., Qi, Z., Liang, X. and Dawei, Z. (2020) An Improved A-Star Based Path Planning Algorithm for Autonomous Land Vehicles. *International Journal of Advanced Robotic Systems*, **17**, 1-13. <https://doi.org/10.1177/1729881420962263>
- [6] 迟旭. 基于改进 A\*算法与动态窗口法融合的移动机器人路径规划研究[D]: [硕士学位论文]. 大连: 大连交通大学, 2021.
- [7] 彭哲, 杨兴果. 基于改进 A\*算法的仓库物料车路径规划[J]. 电工技术, 2023(16): 20-25.
- [8] 王凡, 李铁军, 刘今越, 等. 基于 BIM 的建筑机器人自主路径规划及避障研究[J]. 计算机工程与应用, 2020, 56(17): 224-230.
- [9] 陈家伟, 许颖婷, 陈淑婧, 等. 基于动态启发的双向搜索 A\*路径规划算法[J]. 现代信息科技, 2024, 8(2): 86-91.
- [10] Xu, D., Yue, H., Zhao, Y., Yang, F., Wu, J., Pan, X., et al. (2024) Improved A\* Algorithm for Path Planning Based on Cubesats In-Orbit Electromagnetic Transfer System. *Aerospace*, **11**, Article 394. <https://doi.org/10.3390/aerospace11050394>
- [11] Xie, G., Fang, L., Su, X., Guo, D., Qi, Z., Li, Y., et al. (2024) A Path-Planning Approach for an Unmanned Vehicle in an Off-Road Environment Based on an Improved A\* Algorithm. *World Electric Vehicle Journal*, **15**, Article 234. <https://doi.org/10.3390/wevj15060234>