

基于改进A*的物流机器人路径规划算法

徐 宽, 袁明静, 刘 伟, 温宇骏, 韩树人*

江西理工大学电气工程与自动化学院, 江西 赣州

收稿日期: 2025年4月2日; 录用日期: 2025年5月19日; 发布日期: 2025年5月28日

摘 要

针对物流机器人路径规划中使用的传统A*搜索算法存在的搜索时间长、效率低等问题, 本文提出了一种改进的A*搜索算法。该算法引入动态目标重定向方法解决双向A*搜索算法搜索区域不重合问题, 根据障碍物信息和搜索方向动态调整搜索范围, 将全局八邻域修改为动态n邻域, 有效提高了搜索效率。实验结果表明, 改进算法在随机地图中的平均搜索区域减少60.64%, 平均搜索时间缩短28.88%; 在模拟物流仓库地图中, 平均搜索区域减少59.96%, 平均搜索时间缩短40.50%, 该算法可以有效提高移动物流机器人的路径规划效率。

关键词

A*搜索算法, 路径规划, 搜索策略, 目标重定, 物流机器人

Path Planning Algorithm for Logistics Robots Based on Improved A*

Kuan Xu, Mingjing Yuan, Wei Liu, Yujun Wen, Shuren Han*

School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou Jiangxi

Received: Apr. 2nd, 2025; accepted: May 19th, 2025; published: May 28th, 2025

Abstract

Aiming at the problems of long search time and low efficiency of the traditional A* search algorithm used in the path planning of logistics robots, this paper proposes an improved A* search algorithm. The algorithm introduces a dynamic target redirection method to solve the problem that the search area of the bidirectional A* search algorithm is not coincident. According to the obstacle information and the search direction, the search range is dynamically adjusted, and the global eight neighborhood is modified into a dynamic n neighborhood, which effectively improves the search efficiency. The

*通讯作者。

文章引用: 徐宽, 袁明静, 刘伟, 温宇骏, 韩树人. 基于改进 A*的物流机器人路径规划算法[J]. 人工智能与机器人研究, 2025, 14(3): 647-658. DOI: 10.12677/airr.2025.143064

experimental results show that the average search area of the improved algorithm in random maps is reduced by 60.64%, and the average search time is reduced by 28.88%; in the simulated logistics warehouse map, the average search area is reduced by 59.96%, and the average search time is reduced by 40.50%. This algorithm can effectively improve the path-planning efficiency of mobile logistics robots.

Keywords

A* Search Algorithm, Path Planning, Search Strategy, Target Redefinition, Logistics Robot

Copyright © 2025 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

近年来,随着物流行业快速发展,物流行业对智慧化物流的推崇和追求、人工成本的不断增加,移动物流机器人成为很多企业降本增效的重要手段。机器人在运动过程中必须规避规则摆放的货架、零散的货物及其他移动物流机器人,这就对物流机器人行驶路径提出了更高的要求[1]。路径规划研究一方面有利于促进物流机器人的智能化发展,另一方面可以降低物流行业运送成本,提高运送效率。

目前,机器人路径规划涉及多种算法,包括 A*搜索算法、双向 A*搜索算法[2]等启发式搜索算法,粒子群算法、灰狼算法等智能算法[3]-[8]。近年来,深度强化学习算法[9]、基于 Cornish-Fisher 展开法的最优可靠路径搜索算法[10]等算法也被提出。其中,A*搜索算法[11]因其良好的启发性和快速高效等特点,在机器人路径搜索中被广泛应用。针对 A*搜索算法的局限性,研究人员通过优化扩展策略、改进启发函数、后端轨迹优化等方式提升其规划效率与搜索质量[12]。蒋承杰等通过引入环境障碍率动态调整启发函数权重,提高搜索效率[13]。辛煜、Xiang Dan 等[14] [15]通过优化扩展策略以提高路径质量。张志远等[16]通过预先获取局部障碍率与双向父节点变更,提高路径质量与搜索性能。

综合上述分析,提升物流机器人在进行路径规划时的搜索效率,降低其搜索时间与计算损耗至关重要。本文针对物流系统中的物流仓储环境,提出基于 A*搜索算法的改进与优化。在栅格化地图的基础上针对原有路径,在双向搜索中引入动态目标重定方法,以前向路径与反向路径的最低代价点作为搜索重定目标。同时,采用了动态 n 邻域搜索策略,根据当前迭代点与重定目标点的相对位置及周围障碍物的情况,动态选择搜索范围。最后利用 Python 3.8 在随机地图与模拟物流仓库地图中与 A*搜索算法和双向 A*搜索算法进行对比验证。

2. 问题建模

2.1. 栅格地图搭建

机器人路径规划时,对环境建模的方法主要有栅格地图法、拓扑地图法、代价地图法等。栅格地图法建模相较于拓扑地图法、代价地图法更直观灵活,本文基于栅格法分别建立 32×32 、 64×64 大小的随机地图与模拟物流仓库地图,考虑到物流系统中 AGV 的大小,地图分辨率统一设置为 2.5 m。图 1 表示一个 5×5 大小的栅格地图,使用 $Q = \{(0,0), (0,1), \dots, (4,4)\}$ 表示栅格图坐标集,地图横轴为 X 轴,纵轴为 Y 轴,左下角为坐标原点 $(0,0)$ 。地图分为两类区域,黑色区域表示障碍物区域,白色区域表示无障碍

物区域。

(0,4)	(1,4)	(2,4)	(3,4)	(4,4)
(0,3)	(1,3)	(2,3)	(3,3)	
	(1,2)	(2,2)	(3,2)	(4,2)
	(1,1)			(4,1)
	(1,0)	(2,0)	(3,0)	

Figure 1. Grid map

图 1. 栅格地图

2.2. A*搜索算法

A*搜索算法是一种经典的启发式图搜索算法，兼具高效性与可行性。其代价函数如下：

$$F(n) = h(n) + g(n) \quad (1)$$

其中， $F(n)$ 为节点 n 的总代价， $g(n)$ 表示从起始节点到节点 n 的实际代价， $h(n)$ 为从节点 n 到目标节点的启发式估计，常见估计方法包括欧式距离估计、曼哈顿距离估计和切比雪夫距离估计，具体形式分别表示为：

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

$$h(n) = |x_n - x_g| + |y_n - y_g| \quad (3)$$

$$h(n) = \max(|x_n - x_g|, |y_n - y_g|) \quad (4)$$

式中， (x_n, y_n) 为当前节点的坐标， (x_g, y_g) 为目标点的坐标，本文采用欧式距离估计。

A*算法的算法流程如图 2 所示，搜索区域转换成二维数组，栅格的中心被定义为节点 n ，二维方格定义为可行域(OPENLIST)或不可行域(CLOSE LIST)。 $F(n)$ 评估 OPEN LIST 中的每个节点的代价值并确定节点的顺序。OPEN LIST 用于记录已计算代价值但并未展开的节点，CLOSE LIST 用于储存已展开的节点通过循环判断待检测节点的总代价值来选择下一个节点。

本文采用如图 1 所示的栅格地图构建地图模型，A*搜索算法示意图如图 3 所示，坐标为机器人运动起点，标记为蓝色；坐标为机器人运动终点，标记为绿色；灰色栅格为 A*搜索算法的全部搜索节点。

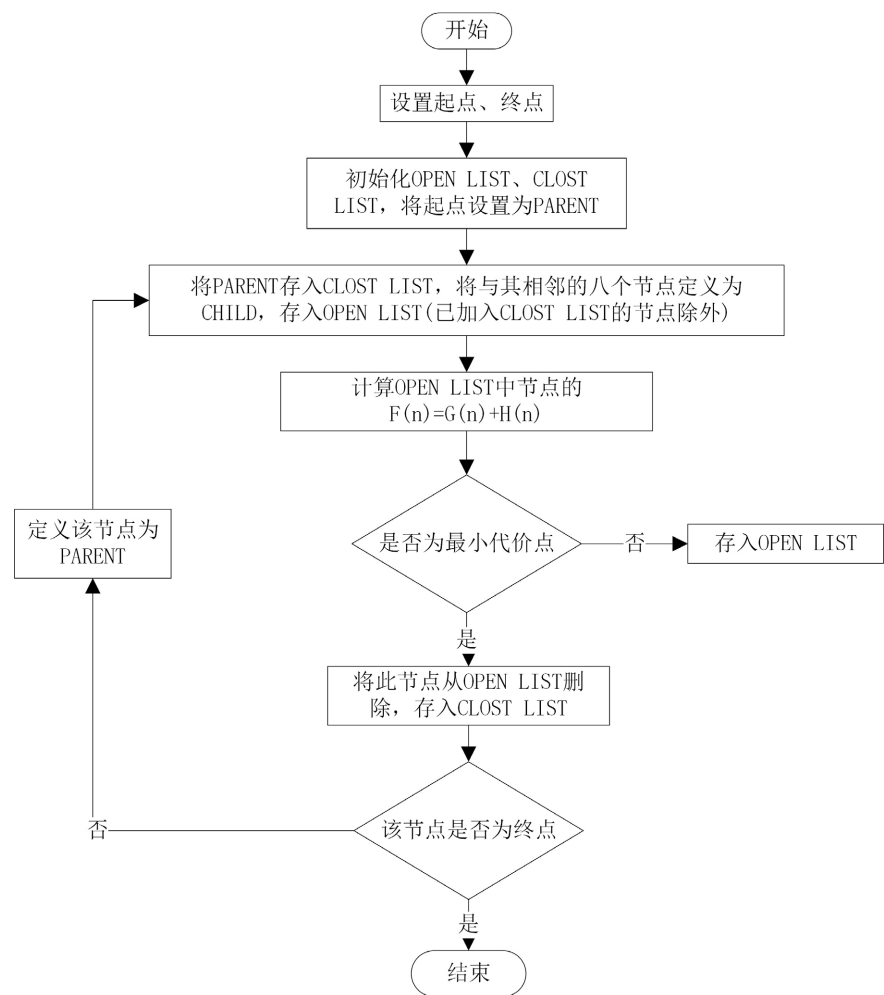


Figure 2. A* search algorithm flowchart
图 2. A*搜索算法流程

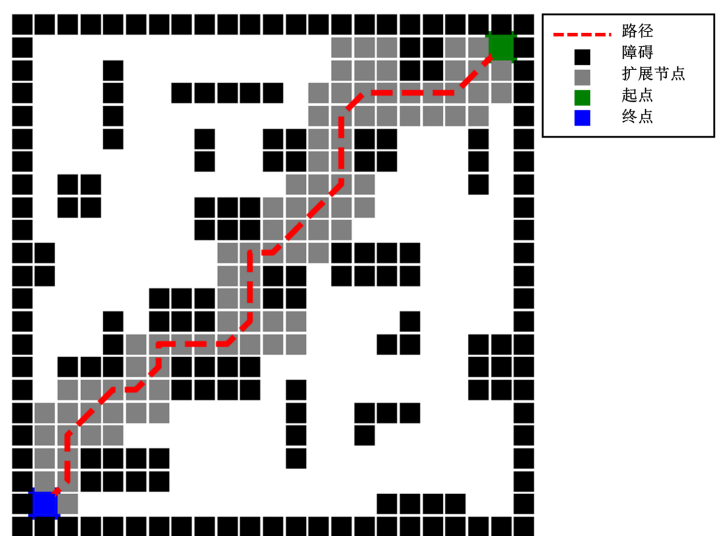


Figure 3. Schematic diagram of A* search algorithm
图 3. A*搜索算法示意图

3. 改进算法

3.1. 双向 A* 搜索算法

传统 A* 搜索算法以起点为开始节点，朝目标点搜索靠近，仅需要建立一个 OPEN 集，双向 A* 搜索算法则分别以起点、终点为开始节点，一条路径朝终点靠近，一条路径朝起点靠近，即起点与终点二者互为起点，也互为终点。需要建立前后两个 OPEN 集。双向 A* 搜索算法的代价函数由前向代价函数与后向代价函数构成，具体可表示为：

$$F_f(n) = G_f(n) + H_f(n) \quad (5)$$

$$F_b(n) = G_b(n) + H_b(n) \quad (6)$$

$$F(n) = F_f(n) + F_b(n) \quad (7)$$

其中， $F_f(n)$ 表示前向搜索中从起始节点到节点 n 的估计总代价， $F_b(n)$ 表示反向搜索中从节点 n 到目标节点的估计总代价； $G_f(n)$ 代表从起始节点到节点 n 的实际代价， $G_b(n)$ 代表从目标节点到节点 n 的实际代价； $H_f(n)$ 代表从节点 n 到目标节点的启发式估计代价， $H_b(n)$ 代表从节点 n 到起始节点的启发式估计代价， $F(n)$ 表示总代价函数。

相较于 A* 搜索算法，双向 A* 搜索算法最大的优点在于，在大空间内，双向搜索可大大减少搜索空间、提高搜索效率。但双向 A* 搜索算法仅仅是将 A* 搜索算法由一维扩充为二维，其前向开放集与后向开放集之间不存在信息交换。理想情况下，两个搜索前沿会在某个节点相遇，从而构成一条从起点到终点的路径。但有时可能会出现两个搜索前沿不相交直至前向搜索包含终点或后向搜索包含起点的情况，即

$$U_0 \cap U_1 = \{Start, End\} \quad (8)$$

相当于对起点与终点分别进行了一次完整的 A* 搜索，搜索节点数增加、搜索效率较 A* 搜索算法也更低。

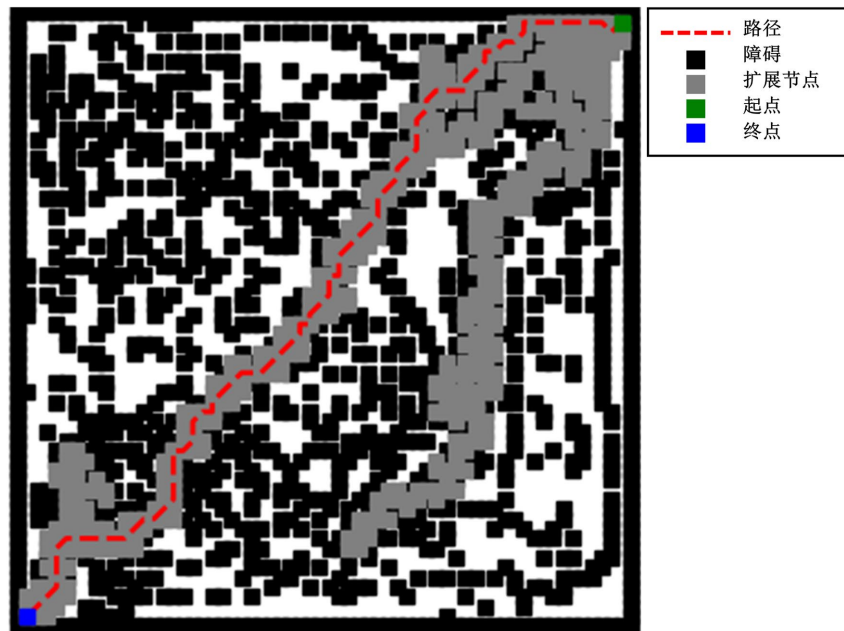


Figure 4. Schematic diagram of bidirectional A* search algorithm
图 4. 双向 A* 搜索算法示意图

如图 4 所示, 前向与反向搜索呈现出两片不存在交集的搜索区域, 最终前向搜索完成一次 A* 搜索, 形成路径解结束双向搜索。此时双向 A* 搜索算法得出的路径与 A* 搜索算法得出的路径相似, 但搜索区域接近于 A* 搜索算法的 2 倍。

3.2. 动态目标重定

针对上述出现的双向搜索区域不相交而导致搜索节点过多、搜索效率下降的问题, 本文引入动态目标重定方法, 通过动态调整前向搜索和反向搜索的目标点来改变单向搜索方向, 从而有效控制双向 A* 搜索的方向, 减少双向 A* 搜索算法的搜索节点, 缩短搜索花费时间。

引入动态目标重定方法后, 在双向搜索过程中, 定义某个时间前向搜索开放集为 $F_b(n)$, 后向搜索开放集为 $F_f(n)$, 则前向搜索目标点与后向搜索目标点为:

$$n_{fore} = \arg \min_{n \in U_1(t)} F_b(n) \quad (9)$$

$$n_{back} = \arg \min_{n \in U_0(t)} F_f(n) \quad (10)$$

其中, n_{fore} 为前向搜索目标点, n_{back} 为后向搜索目标点。随着算法迭代, 后向搜索结果驱动前向搜索, 使前向搜索主动向反向搜索前沿靠近; 同时后向搜索又被前向搜索驱动, 反向搜索同样也向前向搜索前沿靠近, 最终在中心汇合, 形成一条较优解路径。

引入动态目标重定方法后路径规划结果如图 5 所示。与图 4 相比, 引入动态目标重定方法解决了双向搜索区域无交集的问题, 使得双向 A* 搜索算法的搜索节点得到明显减少, 搜索时间缩短。但由于双向 A* 搜索算法与 A* 搜索算法搜索策略都使用八邻域搜索, 始终有部分节点对路径生成影响较小, 存在搜索节点冗余问题。

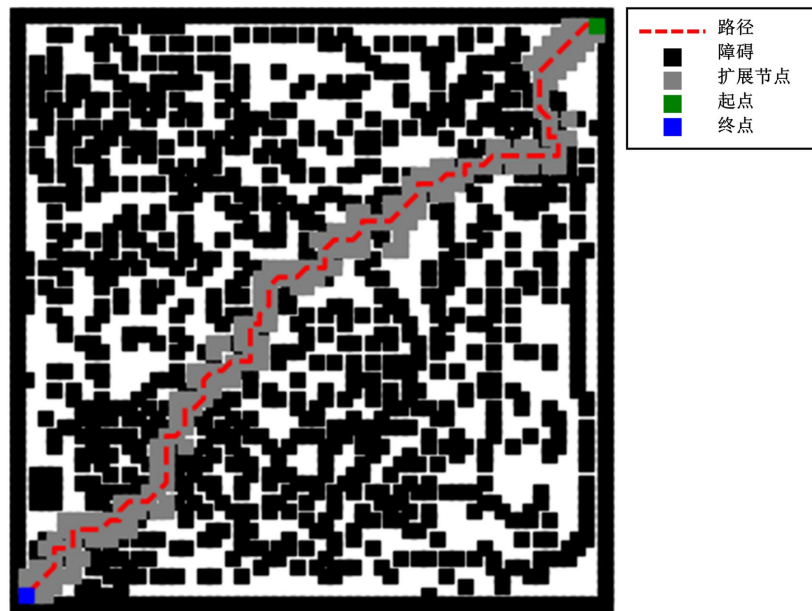


Figure 5. Schematic diagram of bidirectional A* algorithm for dynamic target redefinition
图 5. 动态目标重定双向 A* 算法示意图

3.3. 动态 n 邻域

在 A* 搜索算法与双向 A* 搜索算法中, 节点通常采用八邻域进行扩展搜索, 八邻域扩展搜索范围可

用坐标集表示为:

$$E_n = \left\{ \begin{array}{l} (x_n \pm 1, y_n), \\ (x_n, y_n \pm 1), \\ (x_n \pm 1, y_n \pm 1) \end{array} \right\} \quad (11)$$

如图 6 所示, 灰色区域为父节点(PARENT)即当前节点, 白色区域为子节点(CHILD)即当前节点的邻节点, 子节点的坐标值是父节点的相对坐标。

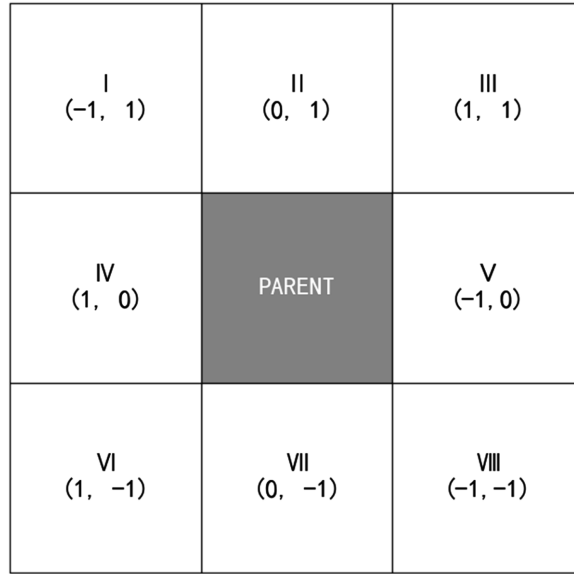


Figure 6. Schematic diagram of eight neighborhood search
图 6. 八邻域搜索示意图

采用八邻域搜索策略既可以保证搜索路径的安全性, 也可以确保搜索方向全面, 但缺点是当存在较大搜索空间时, 采用八邻域搜索策略会造成搜索冗余。

针对扩展搜索策略优化问题, 研究人员提出的方向分为扩展增加与扩展减少, 其中虞立斌、吴飞龙等[17][18]通过优化搜索策略减少搜索区域提升搜索效率; 崔宝侠等[19]则通过增加搜索邻域进而增加更多的搜索方向。本文针对此类问题对搜索策略进行改进, 提出动态 n 邻域搜索策略。动态 n 邻域搜索策略以八邻域搜索为基础, 定义障碍物集合

$$O = \{obs_1 \quad obs_2 \quad \cdots \quad obs_n\} \quad (12)$$

所有可搜索节点为:

$$S = \left\{ \begin{array}{l} (x_n \pm 1, y_n), \\ (x_n, y_n \pm 1), \\ (x_n \pm 1, y_n \pm 1) \end{array} \right\} \quad (13)$$

定义节点 n 与目标节点 E 相对位置为:

$$\left\{ \begin{array}{l} \Delta x = x_E - x_n \\ \Delta y = y_E - y_n \end{array} \right\} \quad (14)$$

根据相对位置关系, 引入

$$\begin{cases} \Delta_x = \text{sgn}(\Delta x) \\ \Delta_y = \text{sgn}(\Delta y) \end{cases} \quad (15)$$

初始化优先搜索方向集合

$$P = \begin{cases} (x_n + \Delta_x, y_n), \\ (x_n, y_n + \Delta_y), \\ (x_n + \Delta_x, y_n + \Delta_y) \end{cases} \quad (16)$$

其中, $\text{sgn}(\Delta x)$ 为符号函数, 取值分别为-1、0、1。

定义拓展搜索集, 当搜索方向不存在障碍物或障碍物对搜索不影响时,

$$\begin{cases} P \cap O = \emptyset \\ U_{et} = \emptyset \end{cases} \quad (17)$$

搜索方向存在障碍物且影响搜索时, 搜索区域扩展到八邻域, 此时搜索节点集合为:

$$\begin{cases} P \cap O \neq \emptyset \\ U_{et} = S \setminus P \end{cases} \quad (18)$$

动态 n 邻域搜索节点集合

$$N(n, E) = \{p \in P \mid p \notin O\} \cup \{s \in U_{et} \mid s \notin O\} \quad (19)$$

4. 仿真验证

本文在 Python 3.8 中进行对比实验, 设立两组对比实验, 第一组实验对比 A*搜索算法、双向 A*搜索算法与本文中改进 A*搜索算法规划路径的各项数据; 第二组实验将改进 A*搜索算法与 A*搜索算法中采用其他邻域搜索方式的算法规划路径的各项数据进行对比, 通过两组对比实验验证改进双向 A*搜索算法的优越性。本次实验仿真栅格图大小为 32×32 与 64×64 , 分为随机地图与物流地图, 模拟地图障碍物由随机数根据地图大小进行取点, 模拟物流地图设置区域型长条障碍模拟货架与服务台, 并随机布置障碍物模拟货物掉落; 给定同一起点和目标点, 规划起点位于左下角(蓝色方块), 目标点位于右上角(绿色方块), 重复 10 次实验。

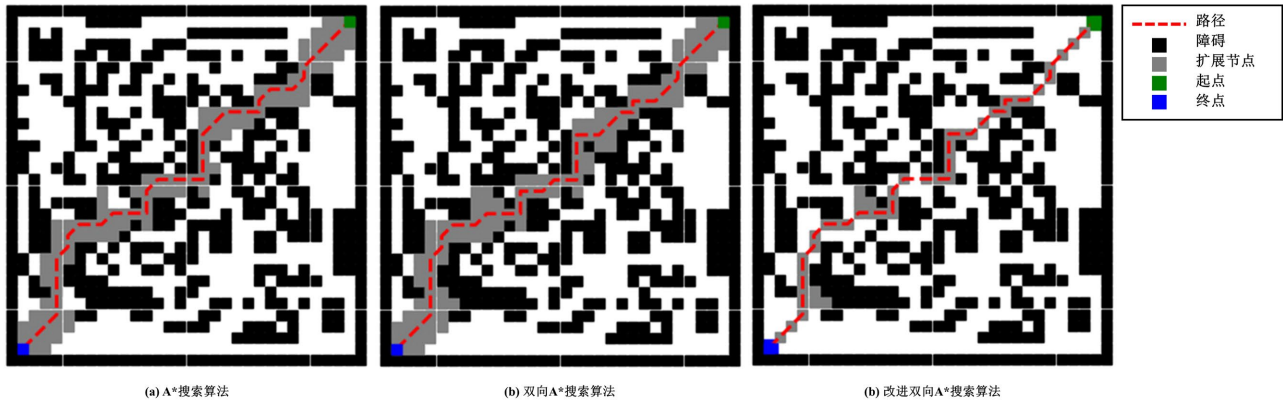


Figure 7. Group 1: Comparison diagram of random map results under 32×32 size

图 7. 对比组一: 32×32 大小下随机地图结果对比图

第一组仿真实验结果如图 7~10 所示, 依次为 32×32 大小下随机地图结果对比图、 32×32 大小下模拟物流仓库地图结果对比图、 64×64 大小下随机地图结果对比图、 64×64 大小下模拟物流仓库地图结果对比图。各个算法在不同栅格图中的路径规划数据如表 1 所示, 数据为十次实验数据均值。

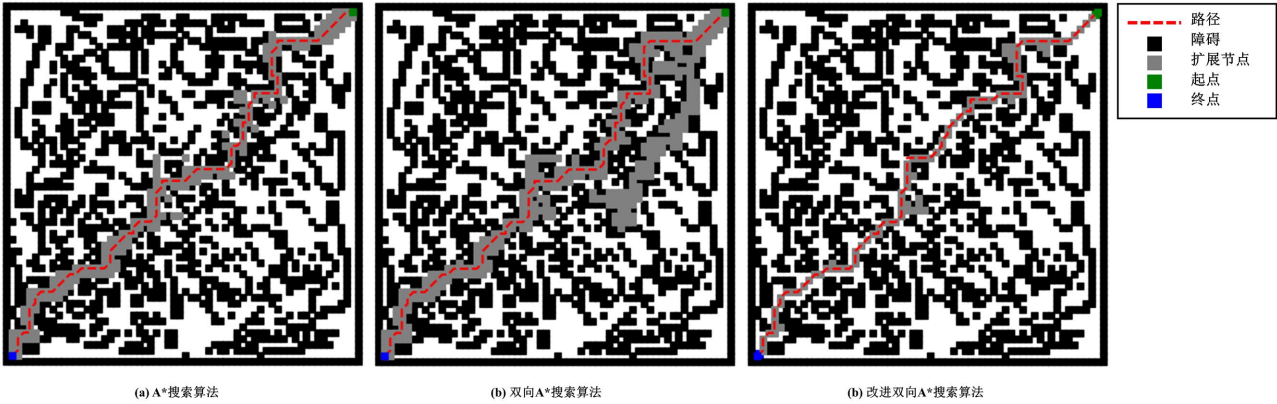


Figure 8. Group 1: Comparison diagram of simulated logistics warehouse map results under 32×32 size
图 8. 对比组一: 32×32 大小下模拟物流仓库地图结果对比图

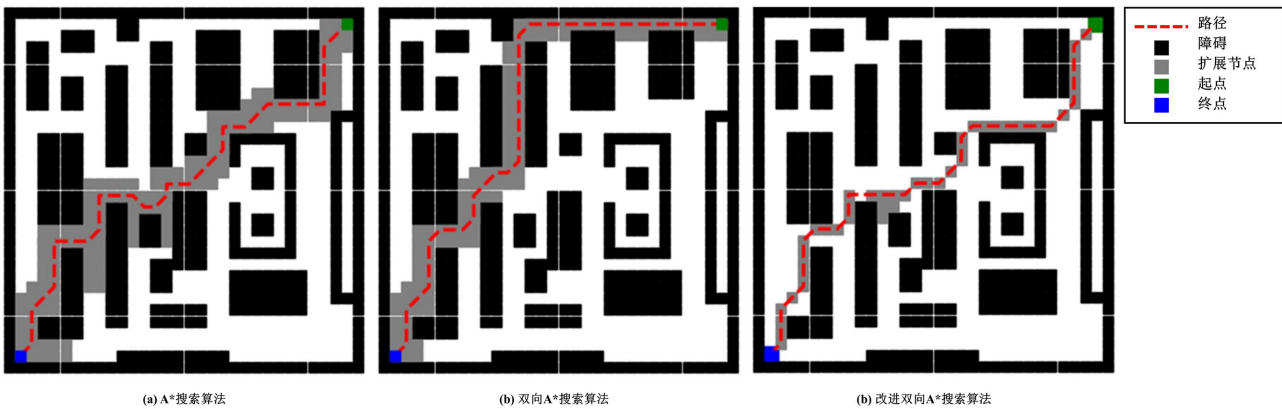


Figure 9. Group 1: Comparison diagram of random map results under 64×64 size
图 9. 对比组一: 64×64 大小下随机地图结果对比图

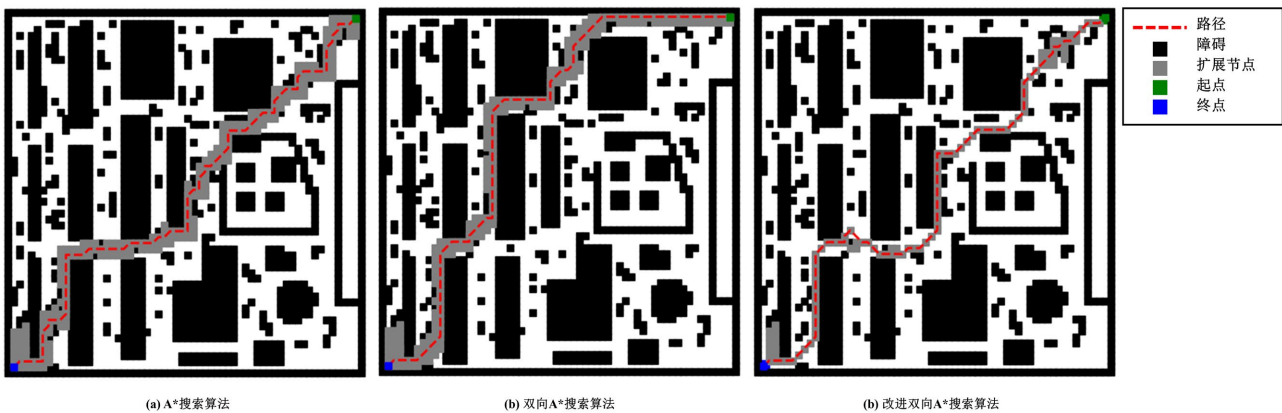


Figure 10. Group 1: Comparison diagram of simulated logistics warehouse map results under 64×64 size
图 10. 对比组一: 64×64 大小下模拟物流仓库地图结果对比图

Table 1. Path planning data of various algorithms in different raster graphs
表 1. 各个算法在不同栅格图中的路径规划数据

栅格地图大小	栅格地图类型	算法	路径平均节点个数	平均搜索节点个数	平均花费时间
32 × 32	随机地图	A*搜索算法	48.6	166.5	0.098563
		双向 A*搜索算法	49.2	150.7	0.056849
		改进 A*搜索算法	52.6	55.9	0.048425
	模拟物流仓库地图	A*搜索算法	52.3	159.2	0.088782
		双向 A*搜索算法	51.9	182.1	0.099689
		改进 A*搜索算法	56.4	48.4	0.058425
64 × 64	随机地图	A*搜索算法	108.6	411.6	0.256984
		双向 A*搜索算法	109.1	376.5	0.233257
		改进 A*搜索算法	116.7	172.0	0.183580
	模拟物流仓库地图	A*搜索算法	118.5	352.4	0.236657
		双向 A*搜索算法	117.2	323.8	0.238794
		改进 A*搜索算法	126.5	152.7	0.207458

第二组实验如图 11、图 12 所示。

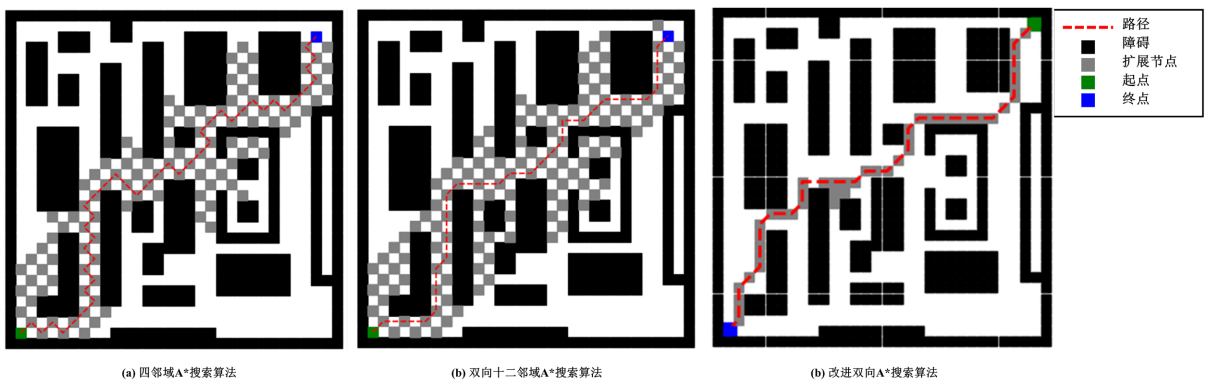


Figure 11. Group 2: Comparison diagram of simulated logistics warehouse map results under 32 × 32 size
图 11. 对比组二：32 × 32 大小下模拟物流仓库地图结果对比图

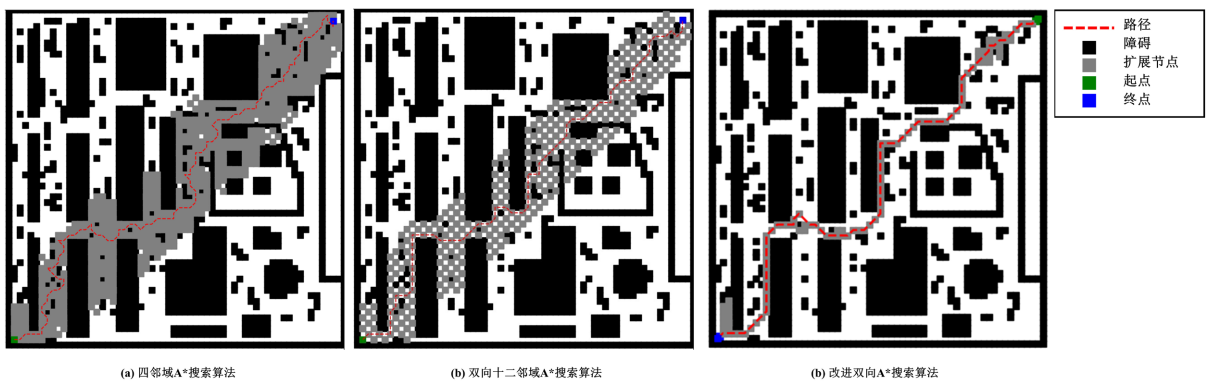


Figure 12. Group 2: Comparison diagram of simulated logistics warehouse map results under 64 × 64 size
图 12. 对比组二：64 × 64 大小下模拟物流仓库地图结果对比图

Table 2. Path planning data in different grid graphs using different neighborhood expansion methods
表 2. 采用不同邻域扩展方法在不同栅格图中的路径规划数据

栅格地图大小	邻域类型	路径平均节点个数	平均搜索节点个数	平均花费时间
32 × 32	四邻域	62.7	127.6	0.076895
	双向十二邻域	48.9	182.1	0.087929
	动态双向 n 邻域	56.4	48.4	0.058425
64 × 64	四邻域	136.1	1068.6	0.446657
	双向十二邻域	109.7	732.9	0.238794
	动态双向 n 邻域	126.5	152.7	0.207458

在实验中，通过对比组一的四组实验，改进 A*搜索算法在随机地图与结构化物流仓库地图中均表现稳定，算法对不规则障碍物(零散货物)和规则障碍物(货架)均具有适应性。通过对比图 9 中的(b)、(c)，可明显观察到引入动态目标重定向后，改进 A*搜索算法在控制双向路径汇合方面得到明显改善；对比图 7~12 中的(a)、(b)、(c)，观察到引入动态 n 邻域搜索策略后，改进 A*搜索算法的搜索节点数明显减少，搜索效率显著提高。

从表 1、表 2 的数据中可以分析得出，在给定障碍物地图环境下，无论是随机地图还是模拟物流仓库地图，在不同地图尺寸下，相较于其他四种搜索算法，改进 A*搜索算法在有双向信息方向提供指引、动态选择邻域范围的前提下，搜索节点数与搜索花费时间均有不同程度的优化，搜索节点个数平均减少 60.12%，搜索时间平均缩短 36.63%。

但由于大量减少搜索节点导致搜索信息部分缺失，搜索路径较其他四种搜索算法均有不同程度加长，(平均增加 8%~10%)无法保证搜索路径最短，在物流场景下，快速搜索响应较严格最优路径更具实际意义。

5. 结论

本文针对物流机器人路径规划中存在的搜索时间长、效率低等问题，提出了一种改进的 A*搜索算法，该算法引入动态目标重定向方法与动态 n 邻域搜索策略相结合。实验证明，改进 A*搜索算法大幅减少搜索节点，缩短搜索花费时间，适合部署在嵌入式设备中，具有较强的实用性。但在减少搜索节点后对全局的实时感知能力减弱，无法保证路径的最优，动态 n 邻域搜索策略进一步优化后，有望继续提升机器人自主移动工作效率。

基金项目

2023 年省级大学生创新创业训练计划项目(s202310407080)。

参考文献

[1] 陈楠, 杜鹏, 乔立春. 基于势场蚁群算法的仓储搬运机器人避障控制方法[J/OL]. 计算机测量与控制: 1-7. <http://kns.cnki.net/kcms/detail/11.4762.TP.20240423.1529.016.html>, 2024-05-06.

[2] 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.

[3] Li, S., Zhao, G. and Yue, W. (2021) Research on Path Planning for Mobile Robot Based on Improved Ant Colony Algorithm. *Journal of Physics: Conference Series*, **2026**, Article ID: 012049. <https://doi.org/10.1088/1742-6596/2026/1/012049>

[4] 朱喜明, 倪恒欣, 赵建鹏. 基于改进灰狼算法的移动机器人三维路径规划[J]. 科学技术与工程, 2025, 25(3): 1125-1132.

-
- [5] Nazarahari, M., Khanmirza, E. and Doostie, S. (2019) Multi-Objective Multi-Robot Path Planning in Continuous Environment Using an Enhanced Genetic Algorithm. *Expert Systems with Applications*, **115**, 106-120. <https://doi.org/10.1016/j.eswa.2018.08.008>
- [6] Wang, L.P., Zhang, Z., Zhu, Q.D. and Ma, S. (2020) Ship Route Planning Based on Double-Cycling Genetic Algorithm Considering Ship Maneuverability Constraint. *IEEE Access*, **8**, 190746-190759.
- [7] Phung, M.D. and Ha, Q.P. (2021) Safety-Enhanced UAV Path Planning with Spherical Vector-Based Particle Swarm Optimization. *Applied Soft Computing*, **107**, Article ID: 107376. <https://doi.org/10.1016/j.asoc.2021.107376>
- [8] Thabit, S. and Mohades, A. (2019) Multi-Robot Path Planning Based on Multi-Objective Particle Swarm Optimization. *IEEE Access*, **7**, 2138-2147. <https://doi.org/10.1109/access.2018.2886245>
- [9] 李松柏. 基于深度强化学习的物流车队配送路径规划及库内分拣作业路径优化研究[J]. 互联网周刊, 2024(2): 28-30.
- [10] 牛曦辰. 基于 Cornish-Fisher 展开法的最优可靠路径搜索算法[J]. 微型电脑应用, 2024, 40(11): 302-306.
- [11] Hart, P., Nilsson, N. and Raphael, B. (1968) A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, **4**, 100-107. <https://doi.org/10.1109/tssc.1968.300136>
- [12] 孟凡齐, 孙潇潇, 朱金善, 等. 基于双向 A*-APF 算法的船舶路径规划研究[J/OL]. 大连海洋大学学报: 1-14. <https://doi.org/10.16535/j.cnki.dlhyxb.2023-293>, 2024-05-06.
- [13] 蒋承杰, 朱华, 谢瑶, 等. 基于改进 A*算法的移动机器人的路径规划[J/OL]. 制造技术与机床: 1-5. <http://kns.cnki.net/kcms/detail/11.3398.TH.20240314.1512.008.html>, 2024-05-06.
- [14] 辛煜, 梁华为, 杜明博, 等. 一种可搜索无限个邻域的改进 A*算法[J]. 机器人, 2014, 36(5): 627-633.
- [15] Xiang, D., Lin, H., Ouyang, J. and Huang, D. (2022) Combined Improved A* and Greedy Algorithm for Path Planning of Multi-Objective Mobile Robot. *Scientific Reports*, **12**, Article No. 13273. <https://doi.org/10.1038/s41598-022-17684-0>
- [16] 张志远, 陈海进, 章一鸣. 基于局部障碍率预获取和双向父节点变更的 A*算法优化[J]. 计算机工程与科学, 2023, 45(9): 1661-1669.
- [17] 虞立斌, 张亿, 黄磊, 等. 双向 A*路径规划算法的邻域改进方法研究[J/OL]. 小型微型计算机系统: 1-9. <http://kns.cnki.net/kcms/detail/21.1106.tp.20240511.1031.025.html>, 2024-05-27.
- [18] 吴飞龙, 郭世永. 融合改进 A~*和动态窗口法的 AGV 动态路径规划[J]. 科学技术与工程, 2020, 20(30): 12452-12459.
- [19] 崔宝侠, 王淼弛, 段勇. 基于可搜索 24 邻域的 A*算法路径规划[J]. 沈阳工业大学学报, 2018, 40(2): 180-184.