

IPA-A*: 带有理想路径面积约束的改进A*算法

曹一波, 杨正东, 范敬文, 赵佳恒

(华南师范大学 软件学院, 广东 佛山 538200)

摘要: 在已知环境中规划最优可行路径仍是自主移动机器人领域一个具有挑战性的问题。传统的A*算法是在全局路径规划领域使用最广泛且知名度最高的算法之一。然而,传统的A*算法存在一些局限性。例如,由于在大规模地图上进行了过多的节点探索,可能导致计算成本过高。此外,即使考虑运行时优化,其也可能生成过于靠近障碍物的路径,并且不能有效地与局部路径规划算法如动态窗口法(DWA)集成。因此,提出一种改进的A*算法,称为理想路径区域A*算法(IPA-A*)。IPA-A*算法通过将理想区域约束纳入传统的启发式搜索函数,解决了过度且不必要的节点探索问题。为了评估其性能,将IPA-A*算法与全局路径规划领域广泛使用的3种算法进行了比较,包括Dijkstra算法、最佳优先搜索(BFS)算法和传统的A*算法。在3种不同复杂程度的地图上进行的仿真实验结果表明,与上述算法相比,IPA-A*算法探索的节点数量、生成路径长度与搜索耗时均显著减少。

关键词: 改进A*算法; 动态窗口法; 路径规划; 移动机器人

DOI: 10.11907/rjtk.241253

中图分类号: TP391.4

文献标识码: A

开放科学(资源服务)标识码(OSID):

文章编号: 1672-7800(2025)004-0075-06



IPA-A* : Improved A* Algorithm with Ideal Path Area Constraint

CAO Yibo, YANG Zhengdong, FAN Jingwen, ZHAO Jiaheng

(School of Software, South China Normal University, Foshan 538200, China)

Abstract: Planning the optimal feasible path in a known environment remains a challenging problem in the field of autonomous mobile robotics. The traditional A* algorithm is one of the most widely used and well-known algorithms in the domain of global path planning. However, the traditional A* algorithm has some limitations. For instance, it may incur high computational costs due to excessive node exploration on large-scale maps. Moreover, even with runtime optimizations, it may generate paths that are too close to obstacles and cannot be effectively integrated with local path planning algorithms, such as the Dynamic Window Approach (DWA) algorithm. In this paper, we propose an improved A* algorithm, called the Ideal Path Area A* (IPA-A*) algorithm. The IPA-A* algorithm addresses the issue of excessive and unnecessary node exploration by incorporating ideal area constraints into the traditional heuristic search function. To evaluate its performance, we compared the IPA-A* algorithm with three algorithms widely used in global path planning: the Dijkstra algorithm, the Best-First Search (BFS) algorithm, and the traditional A* algorithm. Simulation experimental results on maps with three different levels of complexity show that, compared to the aforementioned algorithms, the IPA-A* algorithm significantly reduces the number of explored nodes, the length of the generated path, and the time spent on searching.

Key Words: improved A* algorithm; dynamic window approaches; path planning; mobile robot

0 引言

路径规划算法在各个领域和场景中都得到了广泛应

用^[1]。例如,在导航系统中,用来计算最优路线并提供导航指引,如汽车导航系统和移动地图应用程序等;在无人航空器(Unmanned Aerial Vehicle, UAV)轨迹规划中,用来规划 UAV 飞行路径,以确保安全、高效地完成任务,如

收稿日期: 2024-03-14

扫描二维码阅读全文:

基金项目: 宁波市重点技术研发项目(2021E007)

作者简介: 曹一波(1971-),男,博士,华南师范大学软件学院副教授、硕士生导师,研究方向为机器人自动化技术;杨正东(1998-),男,CCF学生会会员,华南师范大学软件学院硕士研究生,研究方向为轨迹导航;范敬文(1998-),男,CCF学生会会员,华南师范大学软件学院硕士研究生,研究方向为激光视觉融合SLAM;赵佳恒(1999-),男,华南师范大学软件学院硕士研究生,研究方向为轨迹跟踪。本文通讯作者: 杨正东。



UAV 监控和空中摄影等^[2];在智能机器人导航中,用来实现机器人的自主导航,如清洁机器人和仓库机器人等^[3];在搜索和救援行动中,用来计算最优搜索路径,以执行海上救援和山区搜救任务;在城市规划中,用来进行城市道路网络规划和公共交通优化,旨在提高城市交通效率^[4]。

1 相关工作

全局路径规划问题是移动机器人自主导航的关键问题,也是机器人研究领域的重要问题之一^[5]。全局路径规划需要在一个给定的环境中,根据先验条件如地图的可获得性,找到从起点到终点的可行路径。在很多情况下,这些环境充满了动态障碍物,因而增加了规划的复杂性^[6]。

众多专家学者在全局路径规划领域进行了大量理论研究^[7]。全局路径规划算法可以分为3类:传统路径规划算法、基于采样的路径规划算法以及智能仿生算法。传统路径规划算法包括 Dijkstra 算法^[8-9]、最佳优先搜索(Best First Search, BFS)算法^[10]、带有启发式函数的 A* 算法^[11-12]、BA* 算法^[13]、跳点 A* 算法^[14]以及带有视线检查的 Theta* 算法;基于采样的路径规划算法包括 RRT(Rapidly-exploring Random Trees, RRT)算法^[15-16]和概率路图算法(Probabilistic Road Map, PRM);智能仿生算法包括蚁群优化算法(Ant Colony Optimization, ACO)^[17-19]、遗传算法(Genetic Algorithm, GA)^[20]以及人工神经网络(Artificial Neural Networks, ANNs)。

几何搜索在全局路径规划算法中得到了广泛应用。移动机器人的路径规划算法众多,但在实际应用中,Dijkstra 算法、BFS 算法和 A* 算法使用最为广泛。Dijkstra 算法通过搜索地图上的所有节点获取全局最短路径,其获取最短路径的成功率最高,但搜索效率非常低。A* 算法是基于 Dijkstra 算法的一种带有启发式搜索功能的全局搜索算法,与 Dijkstra 算法相比, A* 算法可以更简单、直接的方式处理最短路径搜索问题,但在搜索效果和效率上仍有改进空间^[21]。在复杂环境中, A* 算法生成的路径紧贴障碍物,可能会对无人机的运行造成很大干扰^[22]。如果地图尺寸过大, A* 算法难以进行路径规划,且搜索效率十分低下^[23]。Fu 等^[24]提出一种改进的 A* 算法,通过确定当前点与目标点之间是否有障碍物以缩短生成路径。Saeed 等^[25]引入一种道路强度障碍检测方法,通过确定相邻路径节点之间的连接线是否穿越障碍物来缩短路径长度。Liu 等^[26]提出一种新的启发式函数,通过结合欧几里得距离和曼哈顿距离,减少 A* 算法需要搜索的节点数量。Yin 等^[27]将删除方法添加到 A* 算法的路径选择中,提高了 A* 算法的搜索范围,减少了大规模网络的负面影响,并将搜索节点数量控制在合理范围内。

上述改进的 A* 方法在搜索效率、鲁棒性和生成路径可获得性方面表现较好,但计算成本仍然较高。本文提出

的理想路径区域约束 A* 算法(Idea Path Area A*)在计算代价函数时加入理想路径面积约束,可大幅降低算力需求,同时生成更优的最短路径。当结合使用动态窗口法(Dynamic Window Approaches, DWA)时,还可缩短运行时间。

2 A* 算法基本理论

A* 算法是一种常见的占用网格地图路径规划算法,也是自主导航机器人中使用最多的算法,由 Hart 等^[28]在 1968 年提出。A* 算法通过计算启发式函数来寻求从起点到目标点的最优路径,算法中的距离估计值越接近实际值,最终的搜索速度将越快。作为一种搜索算法,其本质上是对 BFS 算法的优化。该算法从起点开始,首先遍历起点周围的邻近点,然后遍历已遍历过的点的邻近点,逐渐向外扩散,直到找到终点。

A* 算法的理念与 Dijkstra 算法相似,都采用贪心策略,即如果从起点到终点的最短路径经过点 A,那么从起点到点 A 的线段必然是最短的,以此找到从起点到每个可能到达点的最短路径并记录下来。A* 算法与 Dijkstra 算法的区别在于, A* 算法是一种启发式搜索算法。A* 算法不仅关注已搜索的路径,而且预测未搜索的点或状态。因此, A* 算法相对于 BFS 算法调整了邻近节点的搜索顺序,搜索不太可能出现的点,从而更快地找到目标点的最短路径。此外,由于启发式搜索函数的不同设计, A* 算法找到的路径可能不是最短的,但对准确性的牺牲带来了效率的巨大提升,从而获取全局最优路径。传统 A* 算法的启发式函数如下:

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

在公式(1)中, $f(n)$ 为节点 n 的总优先值, A* 算法搜索过程如图 1 所示。当要选择搜索下一拓展节点时,算法总是选择拥有最小预估代价(即最小 $f(n)$ 值)的邻节点, $g(n)$ 为节点从起点出发的遍历代价,如图 2 所示。 $h(n)$ 为节点相较于终点的预估代价,也是 A* 算法的启发搜索函数。

传统 A* 算法常用切比雪夫距离、曼哈顿距离和欧式距离作为启发函数。3 种距离函数如图 3 所示。

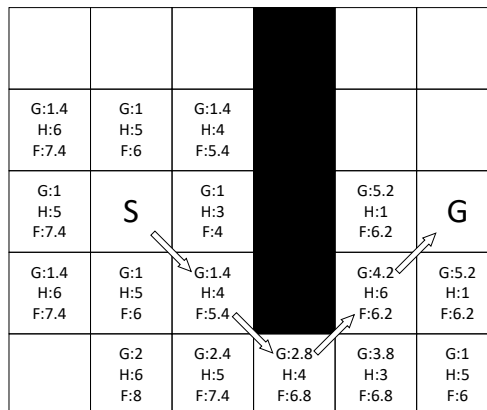


Fig. 1 A* algorithm search process

图 1 A* 算法搜索过程

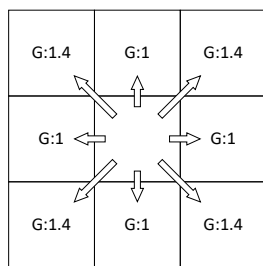
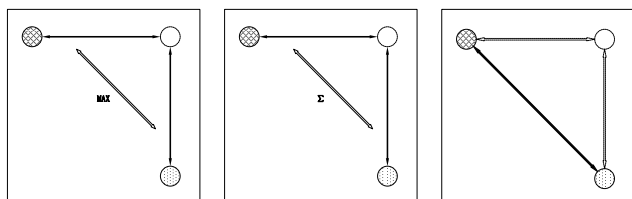
Fig. 2 The calculation process of $g(h)$ value图2 $g(h)$ 值计算过程

Fig. 3 Chebyshev, Manhattan and Euclidean distances

图3 切比雪夫距离、曼哈顿距离与欧氏距离

3 A*算法改进

A*算法是一种基于启发式搜索函数的全局路径规划算法,而启发式搜索函数的优劣对于整个系统性能至关重要。假设估计启发代价值为 $h(n)$,而最佳启发代价值为 $H(n)$,则会出现以下3种情况:

(1)如果 $h(n) < H(n)$,代表当前搜索过程存在大量不必要的节点,且有更好路径存在。

(2)如果 $h(n) = H(n)$,表示当前搜索效率已达到最高,且能够找到最优路径,但该情况发生的概率很小。

(3)如果 $h(n) > H(n)$,表示最优路径已被找到,且使用的搜索节点更少,搜索效率更高。

3.1 理想路径约束

给定一幅地图和一个目的地,最短的理想路径应当是连接起点与终点的直线段。然而在现实情境中,这样的理想条件是稀少的。尽管如此,最短理想路径可以作为寻找最优可行路径的重要参考,因为最优可行路径总是可以在理想路径所在的合理大小区域内找到。

A*算法的核心思想是一种贪婪策略,该策略将寻找全局最优路径转换为在每一步搜索局部最优路径,最终获得全局最优路径。A*算法的贪婪策略具体体现在启发式搜索函数设计上。启发式搜索函数的设计直接决定了A*算法的搜索结果和效率。最常用的启发式搜索函数是欧几里得距离,而IPA-A*算法的启发式搜索函数也是基于欧几里得距离的。欧几里得距离直接计算两点之间的直线距离。作为一个启发式函数,它能够有效地指导下一步搜索的方向,并提供可能以最短距离到达目标的最近节点。然而,欧几里得距离在搜索过程中只提供一个局部搜索方向,而没有考虑当前搜索节点位于地图的哪个区域。这可能导致当前搜索节点偏离真实的全局最优路径,造成路径

长度增加,并加入许多不必要的节点搜索。为了防止局部最优路径过于偏离理想路径,并减少搜索节点数量,本文提出将理想路径约束纳入欧几里得距离作为新的启发式搜索函数。

在设定起点和终点时,可以获得一条连接起点和终点的直线段,将这条直线段定义为 L_{idea} 。本文定义当前搜索节点P、起点S和终点G所形成的三角形面积为 A_{idea} ,当前搜索节点P、起点S和终点G的坐标分别为 $P_x, P_y, S_x, S_y, G_x, G_y$ 。理想面积计算公式如下:

$$A_{idea} = | [S_x * (P_y - G_y) + P_x * (G_y - S_y) + G_x * (S_y - P_y)] / 2 | \quad (2)$$

欧氏距离函数表示空间中两点之间的距离。在二维空间中,两点之间的欧氏距离公式如下:

$$D_{euler} = \sqrt{(G_x - P_x)^2 + (G_y - P_y)^2} \quad (3)$$

为了有机融合理想面积与欧氏距离,需要重新解读欧氏距离函数为以两点之间横纵坐标之差为边长的两个正方形面积之和的平方根。理想路径面积如图4所示。

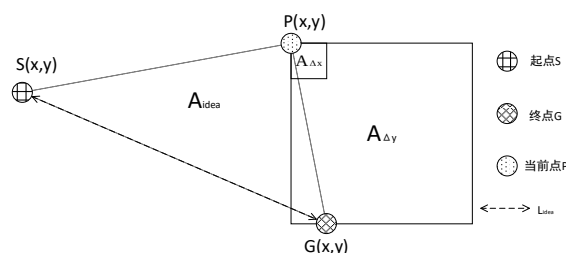


Fig. 4 Ideal path area

图4 理想路径面积

以当前搜索节点与目标点横坐标差值为边长的正方形面积和上述理想面积 A_{idea} 共同组成新的启发函数。新的启发函数公式如下:

$$\begin{cases} H_{ideaEuler}(n) = \sqrt{(G_x - P_{nx})^2 + (G_y - P_{ny})^2} + A_{idea} \\ F(n) = G(n) + H_{ideaEuler}(n) \end{cases} \quad (4)$$

改进启发函数的A*算法伪代码如下:

算法1 改进A*算法

输入: 占位栅格地图 M , 起点 S , 终点 G

输出: 一个连接起点与终点的链表

1. 初始化开放列表 O ;
2. 初始化关闭列表 C ;
3. 将起点 S 加入 O , 设置 f, g, h 值为0;
4. while O 非空 do
5. 提取 O 中 f 值最小节点, 称为 Q ;
6. 从 O 中弹出 Q ;
7. 取出与 Q 相邻的至多8个节点, 并设置其父节点为 Q ;
8. for 每个相邻节点 P do
9. if P 为 G then
10. 搜索结束, 返回链表
11. End
12. 计算 P 的 $A_{idea} = | [S_x * (P_y - G_y) + P_x * (G_y - S_y) + G_x * (S_y -$

$$P_y) \bigr] / 2l ;$$

13. $P.g = Q.g + \text{dis}(P, Q) ;$

14. $P.h = \sqrt{(G_x - P_x)^2 + (G_y - P_y)^2} + A_{\text{idea}}$

15. $P.f = P.g + P.h ;$

16. if 节点在 O 中, 且 f 值小于 P then

17. 跳过;

18. End

19. if 节点在 C 中, 且 f 值小于 P then

20. 跳过;

21. else

22. 将节点加入 O ;

23. end

24. end

25. 将 P 加入 C ;

26. end

3.2 改进A*算法验证

为了验证改进的有效性, 以下提供了3张地图复杂度由简单到复杂的栅格地图。除改进A*算法外, 还分别在3张地图上运行Dijkstra算法、BFS算法和A*算法进行对比。本文采用3个指标评价改进效果, 分别是搜索节点数(Number of Nodes Searched, NONS)、生成路径长度(Path Length, PL)和搜索时间(Plan Time, PT)^[30]。

简单难度地图对比结果如图5所示。在简单难度地图中, Dijkstra算法搜索节点数达8 175栅格, 生成路径长度为107栅格, 搜索时间为4 235 μs ; BFS算法搜索节点数达7 256栅格, 生成路径长度为107栅格, 搜索时间为3 561 μs ; A*算法搜索节点数达1 403栅格, 生成路径长度

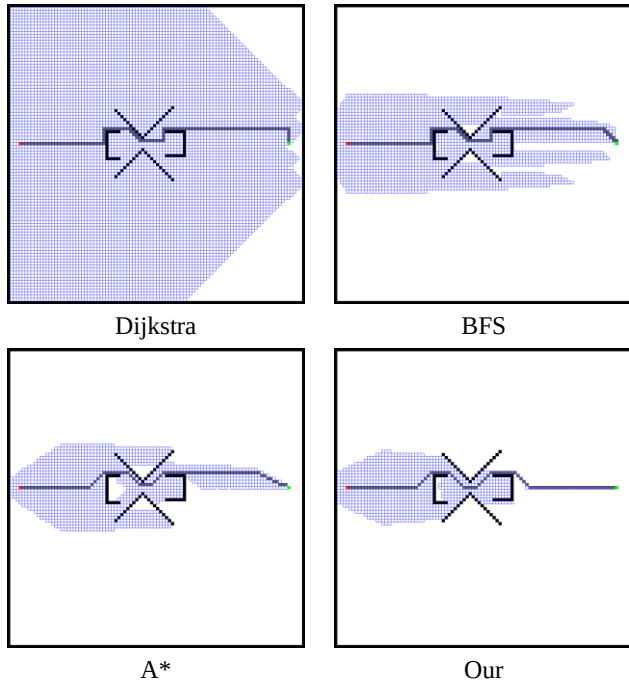


Fig. 5 Comparison results of simple difficulty maps

图5 简单难度地图对比结果

为89栅格, 搜索时间为494 μs ; 改进A*算法搜索节点数达951栅格, 生成路径长度为87栅格, 搜索时间为301 μs 。

根据对比数据, 由图6分析得出, 本文改进算法与另外3种算法中效果最好的相比, 搜索节点数减少了32.2%, 生成路径长度缩短了2.2%, 搜索时间减少了39.1%。

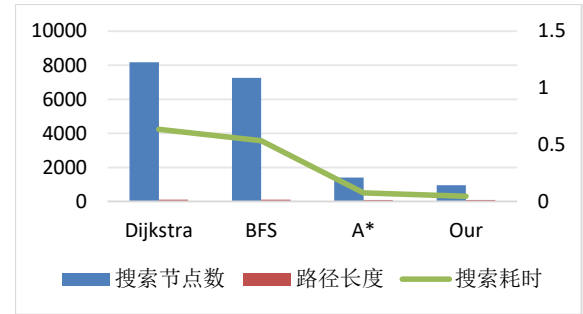


Fig. 6 Comparison of simple difficulty map data

图6 简单难度地图数据对比

中等难度地图对比结果如图7所示。在中等难度地图中, Dijkstra算法搜索节点数达8 106栅格, 生成路径长度为171栅格, 搜索时间为4 122 μs ; BFS算法搜索节点数达6 886栅格, 生成路径长度为171栅格, 搜索时间为3 027 μs ; A*算法搜索节点数达3 025栅格, 生成路径长度为114栅格, 搜索时间为1 075 μs ; 改进A*算法搜索节点数达2 290栅格, 生成路径长度为116栅格, 搜索时间为717 μs 。

根据对比数据, 由图8分析得出, 本文改进算法与另外3种算法中效果最好的相比, 搜索节点数减少了24.30%, 生成路径长度增加了1.75%, 搜索时间减少了33.31%。

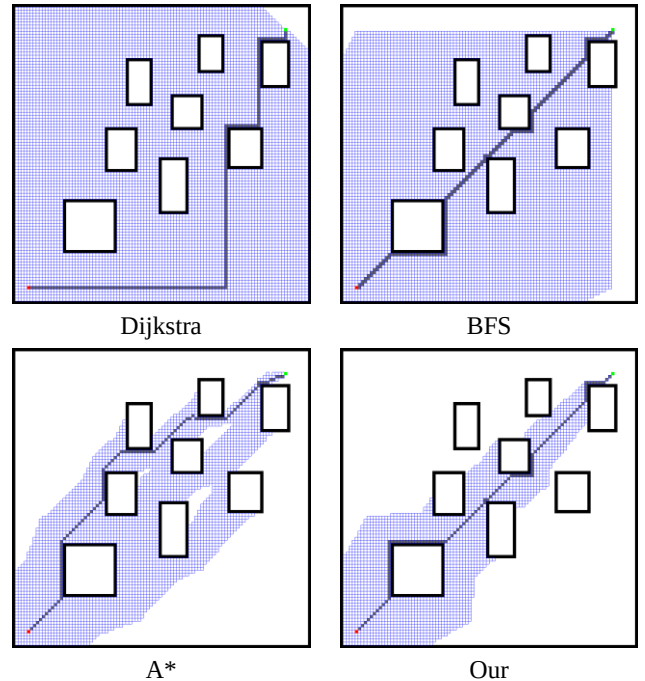


Fig. 7 Comparison results of medium difficulty maps

图7 中等难度地图对比结果

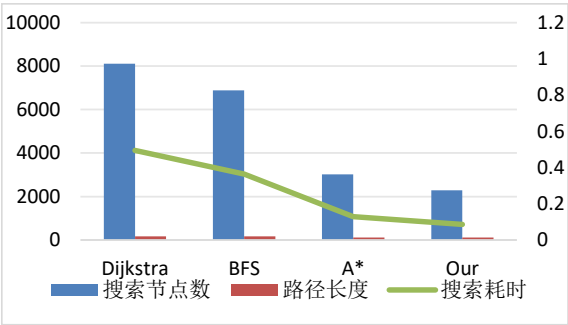


Fig. 8 Comparison of medium difficulty map data
图 8 中等难度地图数据对比

复杂难度地图对比结果如图 9 所示。在复杂难度地图中,Dijkstra 算法搜索节点数达 8 720 栅格,生成路径长度为 167 栅格,搜索时间为 4 659 μs ;BFS 算法搜索节点数达 7 263 栅格,生成路径长度为 167 栅格,搜索时间为 3 489 μs ;A*算法搜索节点数达 4 931 栅格,生成路径长度为 127 栅格,搜索时间为 1 645 μs ;改进 A*算法搜索节点数达 3 225 栅格,生成路径长度为 127 栅格,搜索时间为 1 031 μs 。

根据对比数据,由图 10 分析得出,本节改进算法与另外 3 种算法中效果最好的相比,搜索节点数减少了 34.60%,生成路径长度持平,搜索时间减少了 37.33%。

4 种算法对比结果如表 1 所示。

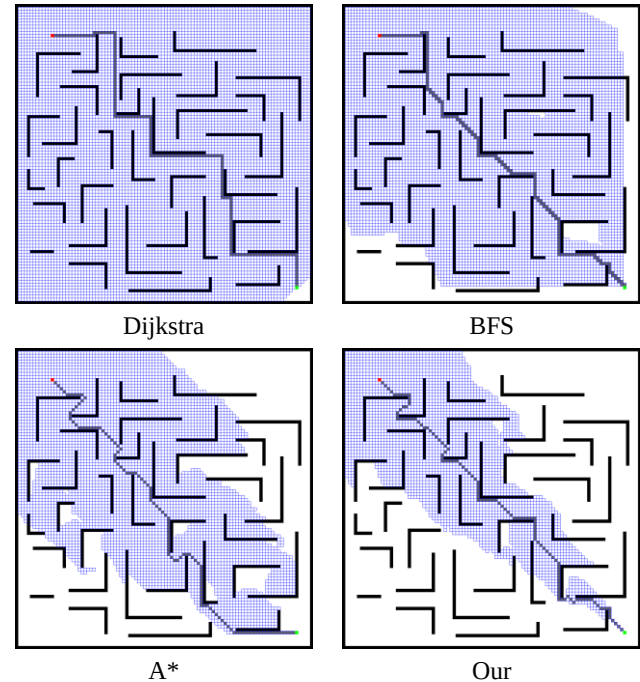


Fig. 9 Comparison results of complex difficulty maps
图 9 复杂难度地图对比结果

4 实验结果分析

本文提出的改进 A*算法在 3 张地图中具有良好表现,具体体现在:

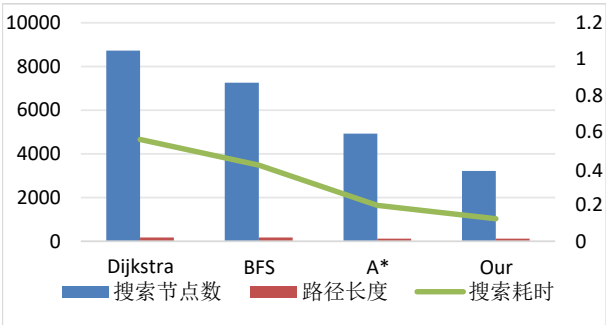


Fig. 10 Complex map data comparison
图 10 复杂地图数据对比

Table 1 Comparison result of algorithms
表 1 算法对比结果

算法名	参数名	简单场景	中等场景	复杂场景
Dijkstra	NONT(cells)	8 175	8 106	8 720
	PL(cells)	107	171	167
	PT(μs)	4 235	4 122	4 659
BFS	NONT(cells)	7 256	6 886	7 263
	PL(cells)	107	171	167
	PT(μs)	3 561	3 027	3 489
A*	NONT(cells)	1 403	3 025	4 931
	PL(cells)	89	<u>114</u>	<u>127</u>
	PT(μs)	494	1 075	1 645
Our	NONT(cells)	<u>951</u>	<u>2 290</u>	<u>3 225</u>
	PL(cells)	<u>87</u>	116	<u>127</u>
	PT(μs)	<u>301</u>	<u>717</u>	<u>1 031</u>

注:cells 为遍历格数,下划线并加粗数据代表最优结果

(1)在简单难度地图场景中,本文提出的改进 A*算法相比其他 3 种算法,在搜索节点数、生成路径长度和搜索耗时上分别减少了 32.22%~88.37%、2.25%~18.70%、39.07%~92.90%。

(2)在中等难度地图场景中,本文提出的改进 A*算法相比其他 3 种算法,在搜索节点数、生成路径长度和搜索耗时上分别减少了 24.30%~71.75%、-1.75%~32.17%、33.31%~82.61%。

(3)在复杂难度地图场景中,本文提出的改进 A*算法相比其他 3 种算法,在搜索节点数、生成路径长度和搜索耗时上分别减少了 34.60%~63.02%、0.00%~23.06%、37.33%~77.88%。

5 结语

本文针对传统 A*算法存在的搜索效率较低、搜索节点过多的问题,提出带有理想路径约束的改进 A*算法。改进算法通过将连接起点与终点的线段作为理想路径,并将其与每次搜索迭代中当前搜索点围成的面积加入传统欧式距离作为新的启发函数,在 3 张不同复杂程度的地图上分别运行 Dijkstra 算法、BFS 算法、传统的 A*算法和改进 A*算法,通过结果数据分析得到本文提出的改进 A*算法在搜索节点数和搜索耗时两项关键指标上有显著改进,在

生成路径长度方面还有进步空间。

然而,A*算法作为全局路径规划算法无法指导实际的移动机器人移动,如何将A*算法与能指导实际机器人移动的局部路径规划算法相结合,取长补短,是下一步研究的方向。

参考文献:

- [1] VAGALE A, BYE R T, OUCHEIKH R, et al. Path planning and collision avoidance for autonomous surface vehicles II: a comparative study of algorithms[J]. *Journal of Marine Science and Technology*, 2021, 26: 1307–1323.
- [2] FANG Y X, YAO Y P, ZHU F, et al. Piecewise-potential-field-based path planning method for fixed-wing UAV formation [J]. *Scientific Reports*, 2023, 13: 2234.
- [3] CHEN Y, WU J F, HE C S, et al. Intelligent warehouse robot path planning based on improved ant colony algorithm [J]. *IEEE Access*, 2023, 11: 12360–12367.
- [4] RAGOTHAMAN S, MAAREF M, KASSAS Z M. Autonomous ground vehicle path planning in urban environments using GNSS and cellular signals reliability maps: models and algorithms [J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2021, 57(3): 1562–1580.
- [5] LIU H X, ZHANG Y H. ASL-DWA: an improved A-star algorithm for indoor cleaning robots [J]. *IEEE Access*, 2020, 10: 99498–99515.
- [6] TESO-FZ-BETOÑO D, ZULUETA E, FERNANDEZ-GAMIZ U, et al. A free navigation of an AGV to a non-static target with obstacle avoidance [J]. *Electronics*, 2019, 8(2): 159.
- [7] NAZARAHARI M, KHANMIRZA E, DOOSTIE S. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm [J]. *Expert Systems with Applications*, 2019, 115: 106–120.
- [8] SUN Y H, FANG M, SU Y X. AGV path planning based on improved Dijkstra algorithm [J]. *Journal of Physics: Conference Series*, 2021, 1746(1): 012052.
- [9] BAILEY T, NIETO J, GUIVANT J, et al. Consistency of the EKF-SLAM algorithm [C]//2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006: 3562–3568.
- [10] JESUTHAS N J A, SOMASKANDAN S. Path-finding and planning in a 3D environment an analysis using bidirectional versions of Dijkstra's, weighted A*, and greedy best first search algorithms [C]//2022 2nd Asian Conference on Innovation in Technology, 2022: 1–8.
- [11] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths [J]. *IEEE Transactions on Systems Science and Cybernetics*, 1968, 4(2): 100–107.
- [12] SEDIGHI S, NGUYEN D V, KUHNERT K D. Guided hybrid A-star path planning algorithm for valet parking applications [C]//2019 5th International Conference on Control, Automation and Robotics, 2019: 570–575.
- [13] YU D X, WANG L C, WU X C, et al. Implementation and visualization of weighted A-star algorithm and bidirectional weighted A-star algorithm under large-scale road network [C]//International Conference on Computer Graphics, Artificial Intelligence, and Data Processing, 2023: 12604.
- [14] LIU L S, YAO J X, HE D W, et al. Global dynamic path planning fusion algorithm combining jump-A* algorithm and dynamic window approach [J]. *IEEE Access*, 2021, 9: 19632–19638.
- [15] HAUER F, TSOTRAS P. Deformable rapidly-exploring random trees [C]//Cambridge: Robotics: Science and Systems, 2017.
- [16] HIRAKAWA T, YAMASHITA T, FUJIYOSHI H. Scene context-aware rapidly-exploring random trees for global path planning [C]//2019 IEEE International Conference on Pervasive Computing and Communications Workshops, 2019: 608–613.
- [17] MAHESHWARI P, SHARMA A K, VERMA K. Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization [J]. *Ad Hoc Networks*, 2021, 110: 102317. 1–102317. 15.
- [18] NING J X, ZHANG Q, ZHANG B, et al. A best-path-updating information-guided ant colony optimization algorithm [J]. *Information Sciences*, 2018, 433: 142–162.
- [19] KUMAR P M, MANOGARAN G, SUNDARASEKAR R, et al. Ant colony optimization algorithm with internet of vehicles for intelligent traffic control system [J]. *Computer Networks*, 2018, 144: 154–162.
- [20] ELHOSENY M, THARWAT A, HASSANIEN A E. Bezier curve based path planning in a dynamic field using modified genetic algorithm [J]. *Journal of Computational Science*, 2018, 25: 339–350.
- [21] GOLDENBERG M. The heuristic search research framework [J]. *Knowledge-Based Systems*, 2017, 129: 1–3.
- [22] LIU J H, YANG J G, LIU H P, et al. An improved ant colony algorithm for robot path planning [J]. *Soft Computing*, 2017, 21: 5829–5839.
- [23] MAC T T, COPOT C, TRAN D T, et al. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization [J]. *Applied Soft Computing*, 2017, 59: 68–76.
- [24] FU B, CHEN L, ZHOU Y, et al. An improved A* algorithm for the industrial robot path planning with high success rate and short length [J]. *Robotics and Autonomous Systems*, 2018, 106: 26–37.
- [25] SAEED R A, RECUPERO D R, REMAGNINO P. A boundary node method for path planning of mobile robots [J]. *Robotics and Autonomous Systems*, 2020, 123: 103320.
- [26] LIU H X, ZHANG Y H. ASL-DWA: an improved A-star algorithm for indoor cleaning robots [J]. *IEEE Access*, 2022, 10: 99498–99515.
- [27] YIN W, YANG X G. A totally astar-based multi-path algorithm for the recognition of reasonable route sets in vehicle navigation systems [J]. *Procedia-social and Behavioral Sciences*, 2013, 96: 1069–1078.
- [28] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths [J]. *IEEE Transactions on Systems Science and Cybernetics*, 1968, 4(2): 100–107.
- [29] WANG X D, ZHANG H W, LIU S, et al. Path planning of scenic spots based on improved A* algorithm [EB/OL]. <https://www.nature.com/articles/s41598-022-05386-6.pdf>.
- [30] LIU L S, LIANG J R, GUO K Q, et al. Dynamic path planning of mobile robot based on improved sparrow search algorithm [EB/OL]. <https://www.mdpi.com/2313-7673/8/2/182>.

(责任编辑:黄健)