

基于自适应启发函数和逆向寻优策略的改进 A* 移动机器人路径规划算法

宋卫猛¹, 王毅²

(1. 重庆理工大学 两江人工智能学院, 重庆 400054; 2. 重庆理工大学 机械工程学院, 重庆 401135)

摘要: 移动机器人大多数情况都是在室外和室内障碍物环境下进行移动; 因此, 在这些障碍物环境中, 高效率、短路径和少转折点的路径规划算法对移动机器人导航至关重要; 针对在室外和室内障碍物环境下 A* 算法无法同时保持高效率、短路径和少转折点的问题, 提出了一种基于自适应启发函数和逆向寻优策略的改进 A* 算法; 通过增加自适应权重系数、引入父节点的影响力并对搜索方向进行筛选, 减少了搜索面积, 提高了搜索效率; 采用逆向寻优策略对路径进行进一步优化, 缩短了路径长度, 减少了转折点数量; 为了评估改进 A* 算法的性能, 在仿真实验中设置常见的室外和室内障碍物环境并与 A* 算法对比; 仿真实验结果表明, 改进 A* 算法在效率、路径长度和转折点数量方面具有显著优势, 能够有效地应用于移动机器人的导航中。

关键词: 移动机器人; 路径规划; A* 算法; 自适应启发函数; 筛选搜索方向; 路径优化

Improved A* Algorithm for Mobile Robot Path Planning Based on Adaptive Heuristic Function and Reverse Optimization Strategy

SONG Weimeng¹, WANG Yi²

(1. School of Liangjiang Artificial Intelligence, Chongqing University of Technology, Chongqing 400054, China;
2. School of Mechanical Engineering, Chongqing University of Technology, Chongqing 401135, China)

Abstract: Mobile robots often move in both outdoor and indoor environments with obstacles. Therefore, in these environments, path planning algorithms have the characteristics of efficient, short, and fewer turning points, which are crucial for mobile robot navigation. To address this problem, an improved A* algorithm based on adaptive heuristic function and reverse optimization strategy is proposed. By increasing the adaptive weight coefficient, introducing the influence of the parent node, and filtering the search direction, the search area is reduced, and the search efficiency is improved. A reverse optimization strategy is used to further optimize the path, shorten the path length, and reduce the number of turning points. To evaluate the performance of the improved A* algorithm, simulation experiments are conducted in common outdoor and indoor environments with obstacles. Experimental results show that compared with the A* algorithm, the improved A* algorithm has significant advantages in efficiency, path length, and the number of turning points, and can be effectively applied in the navigation of mobile robots.

Keywords: mobile robot; path planning; A* algorithm; adaptive heuristic function; filtering search direction; path optimization

0 引言

随着机器人技术的不断成熟, 机器人产业得到了迅速的发展, 在工业、服务业、医疗业等众多领域之中被

广泛应用^[1-2], 在紧急和危机场景, 机器人甚至可能是不可替代的^[3], 因此对机器人的要求也越来越高, 从早些年机械化和单一化慢慢向人性化和智能化转变。其中, 路径规划技术是机器人走向人性化和智能化的关键技术

收稿日期:2023-11-09; 修回日期:2024-01-03。

基金项目:重庆市教育委员会科学技术研究重点项目(KJZD-K202201104)。

作者简介:宋卫猛(1997-),男,。

通讯作者:王毅(1981-),男,博士,副教授。

引用格式:宋卫猛,王毅. 基于自适应启发函数和逆向寻优策略的改进 A* 移动机器人路径规划算法[J]. 计算机测量与控制, 2025,33(1):173-180.

之一^[4-5]，可以保证机器人在不发生碰撞的前提下从起始状态到达目标状态，良好的路径规划算法可以快速规划出最优路径，提高运行效率，减少机器人的磨损。

在路径规划算法领域中，基于栅格方法的路径规划算法被广泛运用。文献 [6] 提出了著名的 Dijkstra 算法，这是一种典型的广度优先搜索算法，鲁棒性强，能计算出两点之间的最优路径解。然而，该算法采用的是无向搜索策略，随着栅格地图的扩大，效率逐渐降低，会出现冗余转折点^[7]。文献 [8] 对 Dijkstra 算法进行了改进，有效地解决包括障碍物在内的环境中的最优路径优化，但是路线的平滑性和连续性一般。文献 [9] 引入了贝塞尔曲线修剪路径，使得路径更平滑，此外这种改进算法可以在动态环境中运行并检测到移动障碍物，实时的更新路径。A* 算法是由文献 [10] 提出的一种搜索算法，它巧妙地融合了 Dijkstra 算法和 BFS (Best-First-Search) 算法的优势，被广泛认为是一种高效的启发式搜索算法。该算法的核心思想是利用启发函数来指导搜索方向，从而有效地缩小搜索范围，提高搜索效率。文献 [11] 在启发函数中融合了路径曲率的代价，使规划的路径更加平滑性；文献 [12] 在 A* 算法中引入了双向交替搜索策略和路径节点过滤功能，提高了搜索效率，有效的减少了转弯角度。文献 [13] 在 A* 的启发函数中加入父节点信息并调整权重，从而减少搜索的节点数量，提高搜索效率。文献 [14] 在 A* 算法的基础上，提出了一种结合人工势场法的新启发函数，该函数包含距离和障碍物信息。文献 [15] 使用开始目标代价函数对目标序列进行排序，并将改进的 A* 算法和动态窗口法相结合应用于多目标点规划。文献 [16] 提出了 D* 算法，该算法是由 A* 算法的发展而来，适用于未知环境或动态变化的环境。文献 [17] 通过自动分割聚类地图的方法，使用障碍物位置信息来自动计算。文献 [18] 针对位置环境的路径规划问题，将 D* 进行了改进，引入 Dubins 算法进行局部路径平滑处理。文献 [19] 在原有成本函数的基础上增加了障碍物成本项和转向角成本项作为约束条件，从而保证了导航的安全。

在基于栅格方法的路径规划算法中，A* 算法又是最常用的^[20]。然而，在室外和室内环境中存在着位置随机、形状不同的障碍物，移动机器人有时需要在这两种环境进行转换。在这些情况下，A* 算法的效率往往较低，规划的路径较差，并且转折点较多，在一定程度上影响移动机器人的导航。为了解决这些问题，提出了一种基于自适应启发函数和逆向寻优策略的改进 A* 算法。通过增加自适应权重系数、引入父节点的影响力并对搜索方向进行筛选，减少了搜索面积，提高了搜索效率。采用逆向寻优策略对路径进行进一步优化，缩短了

路径长度，减少了转折点数量。

1 A* 算法

1.1 环境模型构建

在 A* 算法的应用过程中，环境建模模块的作用至关重要。它的主要功能是将实际环境转换为计算机可以解析的栅格地图，为路径规划提供了必要的基础。环境建模的技术包括栅格法、拓朴法、可视图法以及单元分解法等。和其他建模方法相比，栅格法在操作上更为简单和容易实现，被广泛应用于描述机器人环境地图。

栅格方法的具体实现是将有限个地图空间进行离散处理，将其分解为许多大小一致、状态由数字表示的栅格单元。通过这种方式，环境空间被划分为个栅格，这些栅格以均匀的方式组合，形成二维地图信息。每个栅格状态信息由 Map_{ij} 如式 (1) 所示：

$$Map_{ij} = \begin{cases} 0 & i \in [1, M], j \in [1, N] \\ 1 & \end{cases} \quad (1)$$

式中， i 表示当前栅格所在的行数，最小为 1，最大为 M 。 j 表示当前栅格所在的列数，最小为 1，最大为 N 。 $Map_{ij} = 0$ 表示可通行的自由区域， $Map_{ij} = 1$ 表示不可通行的障碍物区域。

1.2 路径规划

A* 算法利用评价函数来评估从起始节点到目标节点的最短距离，并依据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。从起始点位置到目标点位置的移动代价评价函数 $f(n)$ 如式 (2) 所示：

$$f(n) = g(n) + h(n) \quad (2)$$

式中， n 表示路径搜索过程中的当前节点位置。 $g(n)$ 表示从起始节点位置到当前节点位置的实际移动距离。 $h(n)$ 表示从当前节点位置到目标节点位置的预估距离，即启发函数。

A* 算法用欧式距离来计算 $h(n)$ ，如式 (3) ~ (4) 所示：

$$g(n) = g(f) + \sqrt{(x_n - x_f)^2 + (y_n - y_f)^2} \quad (3)$$

$$h(n) = \sqrt{(x_g - x_n)^2 + (y_g - y_n)^2} \quad (4)$$

式中， (x_n, y_n) 为当前节点 n 的位置坐标。 (x_f, y_f) 为当前节点 n 的父节点 f 的位置坐标。 (x_g, y_g) 为目标节点 g 位置坐标。

通过欧式距离计算当前节点位置到目标节点位置的代价，能够满足最短路径的要求，从而使得 A* 算法在减少搜索节点的同时，也能够像 Dijkstra 算法一样获得最短路径。

2 改进 A* 算法

2.1 自适应启发函数

在实际的路径规划过程中，当前节点位置与目标节

点位置之间的距离估计值往往和实际值不同。当当前节点位置距离目标节点位置较远时, 估计值往往小于实际值, 因此需要适当增大估计值的权重系数。而当当前节点位置距离目标节点位置较近时, 估计值和实际值相差不大, 此时应当适当降低估计值的权重。因此, 添加能够根据当前节点的位置变化自适应调整权重值^[21], 当前节点位置距离目标节点位置较远时, 权重系数大, 启发性强。当前节点位置距离目标节点位置较近时, 权重系数小, 搜索区域广。

为了进一步优化算法的性能, 将父节点的影响力加入到自适应启发函数中。这种做法可以增强启发函数的启发性, 从而减少算法需要遍历的节点数量和循环遍历的次数, 提高搜索效率^[22]。但是, 在引入父节点之后, $h(n)$ 和 $g(n)$ 之间的平衡可能会被打破, 导致搜索步长过长而陷入局部最优解。因此, 需要相应地降低引入父节点之后的权重。

自适应启发函数 $h'(n)$ 如式 (5)、式 (6) 所示:

$$h'(n) = \omega \times [h(n) + h(n-1)] \quad (5)$$

$$\omega = a \times (1 + l_n/L) \quad (6)$$

式中, ω 为自适应启发函数 $h'(n)$ 的自适应权重。 a 是调节比例系数, 为了降低引入父节点影响力之后的权重。 l_n 是当前节点位置到目标节点位置的欧式距离, L 是起始节点位置到目标节点位置的欧式距离。

2.2 筛选搜索方向

A* 算法当前节点位置到达邻近节点位置的搜索方向为 8 个。8 个搜索方向会降低算法的搜索速度, 所以将搜索方向根据当前节点位置和目标节点位置的相对方向进行优化, 减少搜索方向为 5 个^[23], 以使算法更加有效地向目标节点靠近。通过减少搜索方向的数量, 算法可以更快地找到最优路径, 并且在搜索过程中减少不必要的计算开销。

当前节点和目标节点的连线和水平的夹角设为 θ , 夹角 θ 、保留方向和舍弃方向之间的对应关系如表 1 所示。

表 1 夹角 θ 、保留方向和舍弃方向的对应

θ	保留方向	舍弃方向
[22.5, 67.5)	左上, 上, 右上, 右, 右下	下, 左下, 左
[67.5, 112.5)	左, 左上, 上, 右上, 右	右下, 下, 左下
[112.5, 157.5)	左下, 左, 左上, 上, 右上	右, 右下, 下
[157.5, 202.5)	下, 左下, 左, 左上, 上	右上, 右, 右下
[202.5, 247.5)	右下, 下, 左下, 左, 左上	上, 右上, 右
[247.5, 292.5)	右, 右下, 下, 左下, 左	左上, 上, 右上
[292.5, 337.5)	右上, 右, 右下, 下, 左下	左, 左上, 上
[337.5, 360][0, 22.5)	上, 右上, 右, 右下, 下	左下, 左, 左上

2.3 逆向寻优的路径优化

当改进 A* 算法生成一条路径后, 可能存在着部分

中间节点位置较差和大量转折点, 不利于移动机器人进行移动。为了解决该问题, 使用逆向寻优的路径优化^[24]对路径进行处理, 生成最终路径。逆向寻优的路径优化是一种在已生成路径的基础上进一步优化的方法。它通过反向追溯终点到起点, 并对路径中的相邻节点进行评估和调整, 以缩短路径和减少路径中的转折点。

具体实现流程如图 1 所示, 图中黑色矩形为障碍物, 黑色实线为优化后的路径, S 节点为起始节点, G 节点为目标节点, S 节点、按顺序的数字节点和 G 节点为优化前的路径。首先, 连接 S 节点和 G 节点并检查连线上是否存在障碍物。如果没有障碍物, 那么 S 节点和 G 节点之间的连线为最终路径。如果有障碍物, 需要找到 G 节点的父节点 10 号节点, 连接 S 节点和 10 号节点并检查连线上是否存在障碍物。在这个例子中, S 节点和 G 节点、10 号节点、9 号节点、8 号节点、7 号节点和 6 号节点的连线上均存在障碍物, 直到连接 S 节点和 5 号节点后检查连线上不存在障碍物, 保留这段连线, 5 号节点作为新的起始节点和 G 进行连接并检查, 重复上述步骤, 最终完成对整条路径的优化。

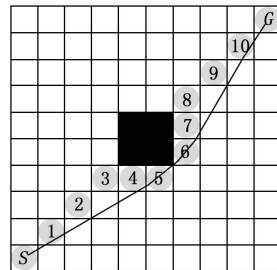


图 1 逆向寻优的路径优化示意图

3 仿真实验

为了验证改进 A* 算法的有效性和优化效果, 设置了 5 种常见的障碍物环境: 室外或室内单一房间内多障碍物环境、2 种室内共用走廊的多房间障碍物环境、2 种综合障碍物环境。仿真实验的硬件平台为 MacOS 操作系统、Apple M1 Pro 处理器、16 GB 内存, 软件平台为 Python 3.8.13。

3.1 室外或室内单一房间的多障碍物环境仿真实验

室外或室内单一房间的多障碍物环境由一些圆形障碍物和矩形障碍物组成。各路径规划算法生成路径对比如图 2、图 3 所示, 栅格地图由个栅格地图组成, 图中黑色区域部分为障碍物, 灰色区域部分为各算法已搜索区域, 灰色实线部分为各算法最终生成的路径。图 2 中的起始节点位置为 (1, 1), 位于栅格地图左下角, 目标节点位置为 (98, 98), 位于栅格地图右上角。图 3 中的起始节点位置为 (98, 1), 位于栅格地图右下角,

目标节点位置为 (1, 98)，位于栅格地图左上角。

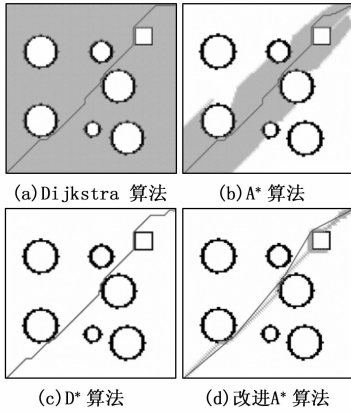


图 2 室外或室内单一房间的多障碍物环境 a 下各路径规划算法生成路径对比图

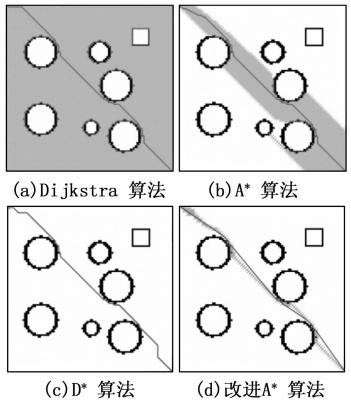


图 3 室外或室内单一房间的多障碍物环境 b 下各路径规划算法生成路径对比图

在室外或室内单一房间的多障碍物环境下的仿真实验数据如表 2 所示，可以看出改进 A* 算法规划时间比 Dijkstra 算法规划时间短 55.00%，比 A* 算法规划时间短 64.00%，比 D* 算法规划时间短 93.08%。规划转折点数量比 Dijkstra 算法规划转折点数量少 25.00%，比 A* 算法规划转折点数量少 55.56%，比 D* 算法规划转折点数量少 33.33%。规划路径长度比 Dijkstra 算法、A* 算法、D* 算法规划路径长度短 4.21%。

表 2 室外或室内单一房间的多障碍物环境实验数据

障碍物环境	算法	时间/s	转折点数量/个	路径长度/m
室外或室内单一房间的多障碍物环境 a	Dijkstra	0.10	13	146.55
	A*	0.15	24	146.55
	D*	0.65	17	146.55
	改进 A*	0.04	10	140.12
室外或室内单一房间的多障碍物环境 b	Dijkstra	0.10	19	145.38
	A*	0.10	30	145.38
	D*	0.65	19	145.38
	改进 A*	0.05	12	139.52

3.2 室内共用走廊的多房间障碍物环境 1 仿真实验

室内共用走廊的多房间障碍物环境 1 由一些墙体组成，共用走廊为一字形，走廊两侧有 6 间房间。各路径规划算法生成路径对比如图 4~6 所示，栅格地图由个栅格地图组成，图中黑色区域部分为障碍物，灰色区域部分为各算法已搜索区域，灰色实线部分为各算法最终生成的路径。图 4~6 中的起始节点位置为 (1, 1)，位于栅格地图左下角的房间内。图 4 目标节点位置为 (98, 98)，位于栅格地图右上角的房间内。图 5 中目标节点位置为 (1, 98)，位于栅格地图左上角的房间内。图 6 目标节点位置为 (98, 1)，位于栅格地图右下角的房间内。

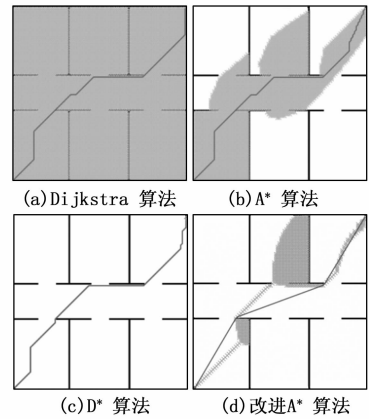


图 4 室内共用走廊的多房间障碍物环境 1a 下各路径规划算法生成路径对比图

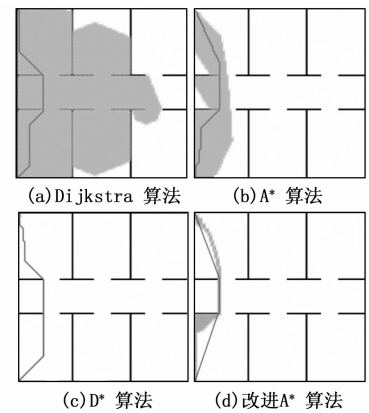


图 5 室内共用走廊的多房间障碍物环境 1b 下各路径规划算法生成路径对比图

在室内共用走廊的多房间障碍物环境 1 下的仿真实验数据如表 3 所示，可以看出改进 A* 算法规划时间比 Dijkstra 算法规划时间短 32.35%，比 A* 算法规划时间短 37.84%，比 D* 算法规划时间短 86.39%。规划转折点数量比 Dijkstra 算法规划转折点数量少 35.29%，比

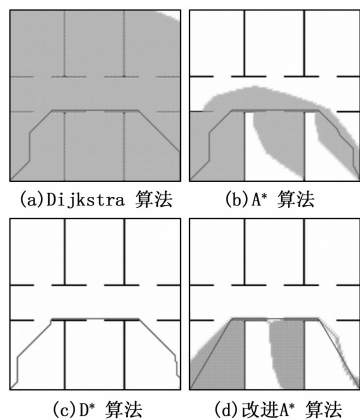


图 6 室内共用走廊的多房间障碍物环境 1c 下各路径规划算法生成路径对比图

A* 算法规划转折点数量少 84.93%, 比 D* 算法规划转折点数量少 62.07%。规划路径长度比 Dijkstra 算法、A* 算法、D* 算法规划路径长度短 5.71%。

表 3 室内共用走廊的多房间障碍物环境 1 实验数据

障碍物环境	算法	时间/s	转折点数量/个	路径长度/m
室内共用走廊的多房间障碍物环境 1a	Dijkstra	0.13	7	155.34
	A*	0.20	24	155.34
	D*	0.70	11	155.34
	改进 A*	0.13	4	144.73
室内共用走廊的多房间障碍物环境 1b	Dijkstra	0.08	5	108.60
	A*	0.05	33	108.60
	D*	0.35	7	108.60
	改进 A*	0.02	3	102.14
室内共用走廊的多房间障碍物环境 1c	Dijkstra	0.13	5	148.88
	A*	0.12	16	148.88
	D*	0.64	11	148.88
	改进 A*	0.08	4	142.38

3.3 室内共用走廊的多房间障碍物环境 2 仿真实验

室内共用走廊的多房间障碍物环境 2 由一些墙体组成, 共用走廊为工字形, 走廊两侧有 8 间房间。各路径规划算法生成路径对比如图 7~9 所示, 栅格地图由个栅格地图组成, 图中黑色区域部分为障碍物, 灰色区域部分为各算法已搜索区域, 灰色实线部分为各算法最终生成的路径。图 7~9 中的起始节点位置为 (1, 1), 位于栅格地图左下角的房间内。图 7 目标节点位置为 (98, 98), 位于栅格地图右上角的房间内。图 8 中目标节点位置为 (1, 98), 位于栅格地图左上角的房间内。图 9 目标节点位置为 (98, 1), 位于栅格地图右下角的房间内。

在室内共用走廊的多房间障碍物环境 2 下的仿真实验数据如表 4 所示, 可以看出改进 A* 算法规划时间比

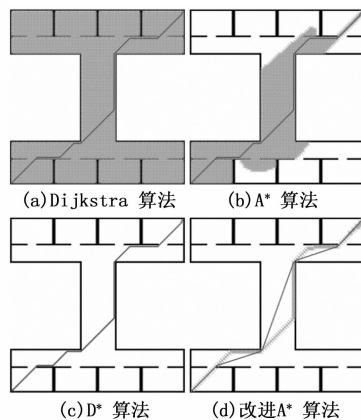


图 7 室内共用走廊的多房间障碍物环境 2a 下各路径规划算法生成路径对比图

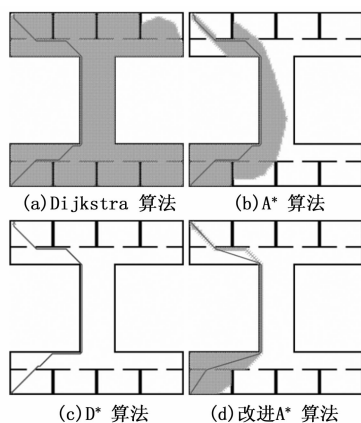


图 8 室内共用走廊的多房间障碍物环境 2b 下各路径规划算法生成路径对比图

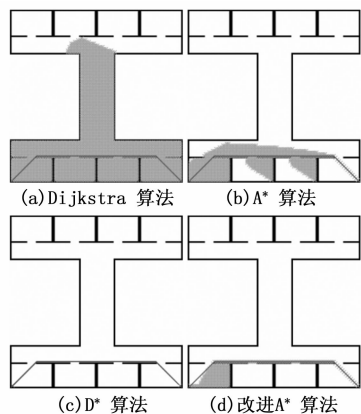


图 9 室内共用走廊的多房间障碍物环境 2c 下各路径规划算法生成路径对比图

Dijkstra 算法规划时间短 23.08%, 比 A* 算法规划时间短 37.50%, 比 D* 算法规划时间短 89.13%。规划转折点数量比 Dijkstra 算法规划转折点数量少 15.00%, 比 A* 算法规划转折点数量少 22.73%, 比 D* 算法规划转折点数量少 10.53%。规划路径长度比 Dijkstra 算法、

A* 算法、D* 算法规划路径长度短 2.95%。

表 4 室内共用走廊的多房间障碍物环境 2 实验数据

障碍物环境	算法	时间/s	转折点数量/个	路径长度/m
室内共用走廊的多房间障碍物环境 2a	Dijkstra	0.05	9	156.51
	A*	0.07	10	156.51
	D*	0.38	8	156.51
	改进 A*	0.04	7	147.84
室内共用走廊的多房间障碍物环境 2b	Dijkstra	0.05	9	149.64
	A*	0.06	10	149.64
	D*	0.37	9	149.64
	改进 A*	0.04	8	146.09
室内共用走廊的多房间障碍物环境 2c	Dijkstra	0.03	2	108.60
	A*	0.03	2	108.60
	D*	0.17	2	108.60
	改进 A*	0.02	2	108.60

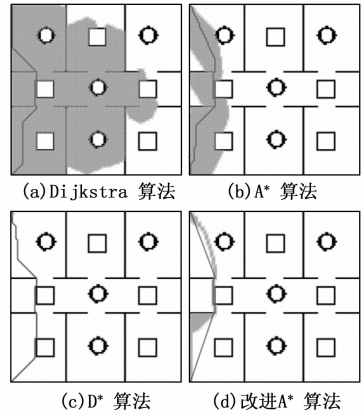


图 11 综合障碍物环境 1b 下各路径规划算法生成路径对比图

3.4 综合障碍物环境 1 仿真实验

综合障碍物环境 1 由室外或室内单一房间的多障碍物环境和室内共用走廊的多房间障碍物环境 1 组成。各路径规划算法生成路径对比如图 10~12 所示，栅格地图由个栅格地图组成，图中黑色区域部分为障碍物，灰色区域部分为各算法已搜索区域，灰色实线部分为各算法最终生成的路径。图 10~12 中的起始节点位置为 (1, 1)，位于栅格地图左下角的房间内。图 10 目标节点位置为 (98, 98)，位于栅格地图右上角的房间内。图 11 中目标节点位置为 (1, 98)，位于栅格地图左上角的房间内。图 12 目标节点位置为 (98, 1)，位于栅格地图右下角的房间内。

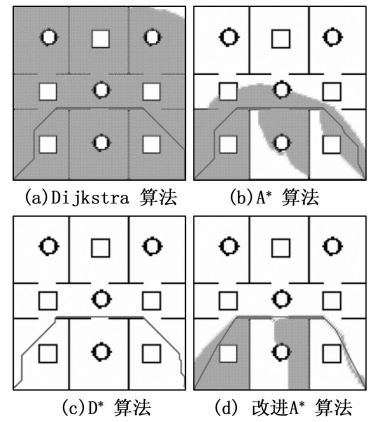


图 12 综合障碍物环境 1c 下各路径规划算法生成路径对比图

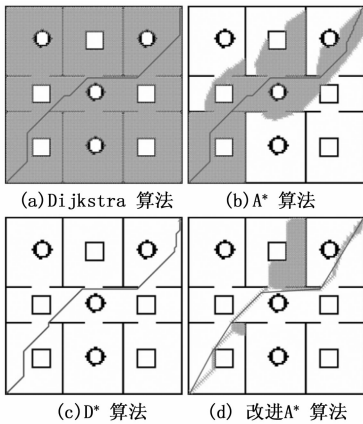


图 10 综合障碍物环境 1a 下各路径规划算法生成路径对比图

在综合障碍物环境 1 下的仿真实验数据如表 5 所示，可以看出改进 A* 算法规划时间比 Dijkstra 算法规划时间短 31.03%，比 A* 算法规划时间短 31.03%，比 D* 算法规划时间短 90.10%。规划转折点数量比 Dijkstra 算法规划转折点数量少 9.52%，比 A* 算法规划转折点数量少 75.32%，比 D* 算法规划转折点数量少

45.71%。规划路径长度比 Dijkstra 算法、A* 算法、D* 算法规划路径长度短 4.74%。

表 5 综合障碍物环境 1 实验数据

障碍物环境	算法	时间/s	转折点数量/个	路径长度/m
综合障碍物环境 1a	Dijkstra	0.11	7	155.34
	A*	0.15	24	155.34
	D*	0.80	11	155.34
	改进 A*	0.10	6	148.29
综合障碍物环境 1b	Dijkstra	0.07	9	109.43
	A*	0.05	37	109.43
	改进 A*	0.03	7	102.96
综合障碍物环境 1c	Dijkstra	0.11	5	148.88
	A*	0.09	16	148.88
	D*	0.81	11	148.88
	改进 A*	0.07	6	142.79

3.5 综合障碍物环境 2 仿真实验

综合障碍物环境 2 由室外或室内单一房间的多障碍物环境和室内共用走廊的多房间障碍物环境 2 组成。各路径规划算法生成路径对比如图 13~15 所示，栅格地

图由个栅格地图组成, 图中黑色区域部分为障碍物, 灰色区域部分为各算法已搜索区域, 灰色实线部分为各算法最终生成的路径。图 13~15 中的起始节点位置为 (1, 1), 位于栅格地图左下角的房间内。图 13 目标节点位置为 (98, 98), 位于栅格地图右上角的房间内。图 14 中目标节点位置为 (1, 98), 位于栅格地图左上角的房间内。图 15 目标节点位置为 (98, 1), 位于栅格地图右下角的房间内。

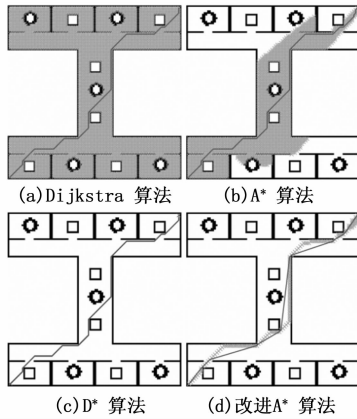


图 13 综合障碍物环境 2a 下各路径规划算法生成路径对比图

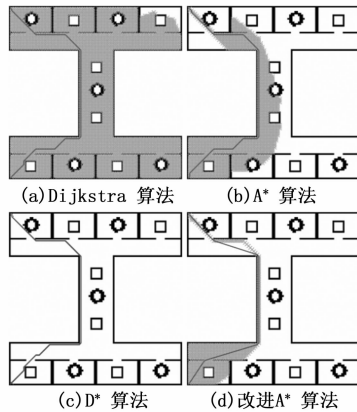


图 14 综合障碍物环境 2b 下各路径规划算法生成路径对比图

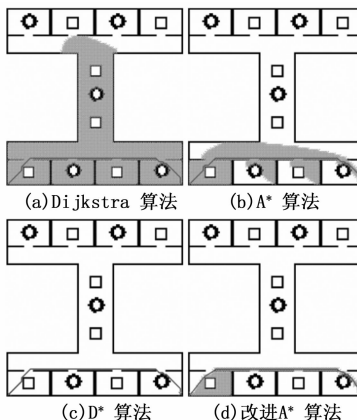


图 15 综合障碍物环境 2c 下各路径规划算法生成路径对比图

在综合障碍物环境 2 下的仿真实验数据如表 6 所示, 可以看出改进 A* 算法规划时间比 Dijkstra 算法规划时间短 9.10%, 比 A* 算法规划时间短 23.08%, 比 D* 算法规划时间短 90.29%。规划转折点数量比 Dijkstra 算法规划转折点数量少 14.29%, 比 A* 算法规划转折点数量少 30.23%, 比 D* 算法规划转折点数量少 11.76%。规划路径长度比 Dijkstra 算法、A* 算法、D* 算法规划路径长度短 2.83%。

表 6 综合障碍物环境 2 实验数据

障碍物环境	算法	时间/s	转折点数量/个	路径长度/m
综合障碍物环境 2a	Dijkstra	0.04	15	158.85
	A*	0.06	18	158.85
	D*	0.43	15	158.85
	改进 A*	0.04	12	153.23
综合障碍物环境 2b	Dijkstra	0.04	11	151.23
	A*	0.04	11	151.23
	D*	0.42	10	151.23
	改进 A*	0.04	8	146.76
综合障碍物环境 2c	Dijkstra	0.03	9	111.53
	A*	0.03	14	111.53
	D*	0.18	9	111.53
	改进 A*	0.02	10	109.70

4 结束语

针对在室内、室外障碍物环境下 A* 算法无法同时保持高效率、短路径和少转折点的问题, 提出了一种基于自适应启发函数和逆向寻优策略的改进 A* 算法, 在室外、室内常见障碍物环境下进行了仿真验证, 仿真实验结果表明, 改进 A* 算法规划所需时间更短, 规划路径长度更短, 规划路径转折点更少, 证明了该算法的优势, 能够有效地应用于移动机器人的导航中。

参考文献:

- [1] RUBIO F, VALERO F, LLOPIS-ALBERT C. A review of mobile robots: concepts, methods, theoretical framework, and applications [J]. International Journal of Advanced Robotic Systems, 2019, 16 (2): 1729881419839596.
- [2] CHAE H W, CHOI J H, SONG J B. Robust and autonomous stereo visual-inertial navigation for non-holonomic mobile robots [J]. IEEE Transactions on Vehicular Technology, 2020, 69 (9): 9613 - 9623.
- [3] ALI M A H, SHANONO I H. Path planning methods for mobile robots: A systematic and bibliometric review [J]. ELEKTRIKA-Journal of Electrical Engineering, 2020, 19 (3): 14 - 34.
- [4] KAZEMI M, GUPTA K K, MEHRANDEZH M. Randomized kinodynamic planning for robust visual servoing [J]. IEEE Transactions on Robotics, 2013, 29 (5): 1197

- 1211.

[5] GORETKIN G, PEREZ A, PLATT R, et al. Optimal sampling-based planning for linear-quadratic kinodynamic systems [C] //2013 IEEE International Conference on Robotics and Automation. IEEE, 2013: 2429 - 2436.

[6] JOHNSON D B. A note on Dijkstra's shortest path algorithm [J]. Journal of the ACM (JACM), 1973, 20 (3): 385 - 388.

[7] ZHU D D, SUN J Q. A new algorithm based on Dijkstra for vehicle path planning considering intersection attribute [J]. IEEE Access, 2021, 9: 19761 - 19775.

[8] ALSHAMMREI S, BOUBAKER S, KOLSI L. Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance [J]. Comput. Mater. Contin, 2022, 72: 5939 - 5954.

[9] DURAKLI Z, NABIYEV V. A new approach based on Bezier curves to solve path planning problems for mobile robots [J]. Journal of Computational Science, 2022, 58: 101540.

[10] 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.

[11] MIN H, XIONG X, WANG P, et al. Autonomous driving path planning algorithm based on improved A* algorithm in unstructured environment [J]. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2021, 235 (2-3): 513 - 526.

[12] LI C, HUANG X, DING J, et al. Global path planning based on a bidirectional alternating search A* algorithm for mobile robots [J]. Computers & Industrial Engineering, 2022, 168: 108123.

[13] LIN M, YUAN K, SHI C, et al. Path planning of mobile robot based on improved A algorithm [C] // Control & Decision Conference. IEEE, 2017. DOI:10.1109/CCDC.2017.7979125.

[14] ZHANG J, WU J, SHEN X, et al. Autonomous land vehicle path planning algorithm based on improved heuristic function of A-Star [J]. International Journal of Advanced Robotic Systems, 2021, 18 (5): 172.

[15] WANG H, YIN P, ZHENG W, et al. Mobile robot path planning based on improved A* algorithm and dynamic window method [J]. Robot, 2020, 42 (3): 346 - 353.

[16] STENTZ A. Optimal and efficient path planning for partially-known environments [C] //Proceedings of the 1994 IEEE international conference on robotics and automation. IEEE, 1994: 3310 - 3317.

[17] HEO S, CHEN J, LIAO Y, et al. Auto-splitting D* lite path planning for large disaster area [J]. Intelligent Service Robotics, 2022, 15 (3): 289 - 306.

[18] YU J, YANG M, ZHAO Z, et al. Path planning of unmanned surface vessel in an unknown environment based on improved D*Lite algorithm [J]. Ocean Engineering, 2022, 266: 112873.

[19] SUN B, ZHANG W, LI S, et al. Energy optimised D* AUV path planning with obstacle avoidance and ocean current environment [J]. The Journal of Navigation, 2022, 75 (3): 685 - 703.

[20] 王中玉, 曾国辉, 黄勃, 等. 改进 A* 算法的机器人全局最优路径规划 [J]. 计算机应用, 2019, 39 (9): 2517.

[21] 李杰. 基于改进 A* Star 算法的移动机器人动态路径规划 [J]. 黑龙江科学, 2023, 14 (16): 47 - 52, 56.

[22] 沈克宇, 游志宇, 刘永鑫, 等. 基于改进 A* 算法的移动机器人路径规划 [J]. 计算机应用研究, 2023, 40 (1): 75 - 79.

[23] 迟旭, 李花, 费继友. 基于改进 A* 算法与动态窗口法融合的机器人随机避障方法研究 [J]. 仪器仪表学报, 2021, (3): 132 - 140.

[24] SHAN E, DAI B, SONG J, et al. A dynamic RRT path planning algorithm based on B-spline [C] //2009 Second International Symposium on Computational Intelligence and Design. IEEE, 2009, 2: 25 - 29.

(上接第 172 页)

[20] PETER BAKIA, GABOR SZEKELYA, Gabor Kosab. Design and characterization of a novel, robust, tri-axial force sensor [J]. Sensors and Actuators A: Physical, 2013, 192: 101 - 110.

[21] 游江辉. 快速称重传感器弹性体结构参数优化及试验研究 [D]. 长沙: 中南林业科技大学, 2019.

[22] 刘九卿. 称重传感器弹性元件金属材料的选择与分析 [J]. 衡器, 2001 (5): 10 - 16.

[23] 高秀者. 称重传感器弹性体材料 40CrNiMo 钢滞弹性的研究 [D]. 镇江: 江苏大学, 2009.

[24] 朱向群. 称重传感器弹性体材料关键技术研究及应用

[D]. 镇江: 江苏大学, 2010.

[25] 王廷喜. 弹簧钢的应力松弛行为研究 [D]. 成都: 西南交通大学, 2012.

[26] 代煜, 张建勋. 基于小波变换和维纳滤波的半导体器件 1/f 噪声滤波 [J]. 物理学报, 2011, 60 (11): 185 - 190.

[27] 孙会娇, 代煜, 张建勋, 等. 直流电源激励下的电路高分辨力应变信号处理 [J]. 仪器仪表学报, 2019, 40 (8): 184 - 190.

[28] 刘九卿. 解析称重传感器误差计算用最佳拟合直线 [J]. 衡器, 2014, 43 (2): 32 - 36.