

基于改进双向 A* 算法的移动机器人路径规划研究

张学锋¹, 胡伟鹏¹, 石军花², 姜兴龙¹

1. 安徽工业大学 特种重载机器人安徽省重点实验室, 安徽 马鞍山 243032

2. 江苏省安全生产科学研究院, 南京 210042

摘要:目的 针对复杂环境下, 解决传统 A* 算法在复杂环境中搜索效率低、路径拐点多等问题, 提出改进的双向 A* 算法。方法 该算法采用正向和反向动态扩展目标点, 使用动态启发式函数, 并将搜索领域从 8 邻域改进为 24 邻域的 8 个方向。为优化路径, 引入多次三阶贝塞尔曲线进行路径平滑。结果 在多障碍和复杂地图情境下, 改进算法表现更高效。在路径规划中, 快速找到优化路径, 减少搜索节点和路径拐点, 实验数据指标提升超过 85%。结论 改进的双向 A* 算法具有更好的适应性和灵活性, 适用于各类复杂环境的路径规划。其性能优势使其成为处理大规模、高复杂度地图的理想选择, 为路径规划领域提供强有力的支持。

关键词: A* 算法; 路径规划; 贝塞尔曲线; 启发函数; 24 邻域

中图分类号: TP242.6 文献标识码: A doi:10.16055/j.issn.1672-058X.2026.0002.019

Research on Mobile Robot Path Planning Based on Improved Bidirectional A* Algorithm

ZHANG Xuefeng¹, HU Weipeng¹, SHI Junhua², JIANG Xinglong¹

1. Anhui Province Key Laboratory of Special Heavy Load Robot, Anhui University of Technology, Ma' anshan 243032, Anhui, China

2. Jiangsu Academy of Safety Science and Technology, Nanjing 210042, China

Abstract: Objective To solve the problems of low search efficiency and numerous path inflection points of the traditional A* algorithm in complex environments, an improved bidirectional A* algorithm is proposed. **Methods** This algorithm dynamically expands target points in both forward and reverse directions, employs dynamic heuristic functions, and improves the search area from an 8-neighborhood to 24-neighborhood in 8 directions. To optimize the path, multiple third-order Bézier curves are introduced for path smoothing. **Results** In scenarios with multiple obstacles and complex maps, the improved algorithm shows higher efficiency. During path planning, it can quickly find the optimized path, reduce the number of searched nodes and path inflection points, and the experimental data index is improved by more than 85%. **Conclusion** The improved bidirectional A* algorithm has better adaptability and flexibility, making it suitable for path planning in various complex environments. Its performance advantages make it an ideal choice for handling large-scale and highly complex maps, and this algorithm provides strong support in the field of path planning.

Keywords: A* algorithm; path planning; Bézier curve; heuristic function; 24-neighborhood

随着近年来人工智能的迅速崛起, 高度自动化和智能化水平的 AGV 应用日益普及, 智能停车场、水产养殖业^[1]、自动化分拣仓库^[2]、集装箱自动化码头^[3], 都是 AGV 常见的应用场景^[4]。在 AGV 中路径规划是

收稿日期: 2023-11-04 修回日期: 2024-02-27 文章编号: 1672-058X(2026)02-0146-10

基金项目: 安徽省教育厅重点实验室项目(TZJQR007-2023), 安徽高校自然科学基金项目(2022AH050290)。

作者简介: 张学锋(1978—), 男, 河北石家庄人, 博士, 教授, 从事虚拟现实技术与人工智能研究。

通信作者: 胡伟鹏(2001—), 男, 硕士研究生, 从事虚拟现实技术与人工智能研究。Email: 1535911291@qq.com。

引用格式: 张学锋, 胡伟鹏, 石军花, 等. 基于改进双向 A* 算法的移动机器人路径规划研究[J]. 重庆工商大学学报(自然科学版), 2026, 43(2): 146-155.

ZHANG Xuefeng, HU Weipeng, SHI Junhua, et al. Research on mobile robot path planning based on improved bidirectional A* algorithm[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2026, 43(2): 146-155.

至关重要的一环。它涉及从起始点到目标点选择最佳路径,在路径距离、消耗时间、访问到的节点数量等方面实现最优化,同时避免与已知障碍物发生碰撞。

在路径规划的领域中,众多学者进行了广泛的算法研究,不断推陈出新,提出了多样化的算法及改进方法。这些算法各具优缺点,展现各种独特的特点。几种常见的路径规划方法包括 Dijkstra 算法^[5]、A* 算法^[6]、遗传算法^[7]、DWA 算法和人工势场法^[8]以及蚁群算法^[9]等。Dijkstra 算法是通过贪婪的渐进式方式,从起始节点到所有其他节点寻找最短路径,从而解决单源最短路径问题的经典算法。与 A* 算法相比,Dijkstra 算法在最短路径搜索效率上处于劣势,主要是缺乏良好的启发性。遗传算法虽然是一种全局优化算法,适合于复杂的搜索空间,可以找到全局最优解,但是计算开销大,不适用于实时性要求高的问题。DWA 算法和人工势场法相比于 A* 算法都适用于局部最优解情况,在面对复杂环境中效果不太好。但传统的 A* 算法在实际应用场景下,仍然存在计算复杂度较高、内存消耗较大、对路径平滑性处理较弱、启发函数较为单一,这将会导致在路径规划过程中扩展很多冗余的节点^[10]。针对上述问题,研究人员对 A* 算法进行了改进,王中玉等^[11]提出了一种双向搜索的 A* 算法,将单向搜索扩大到双向,提升了搜索效率的同时,也引入了精英搜索策略,减少空白处的搜索时间,但同时也存在面对复杂环境情况下搜索效率不优的情况,而且缺少动态性,本文算法则采用了动态的启发式函数,在搜索过程中不断地调用函数,减少冗余的节点,并采用改进的邻域优化,使搜索效率显著提升。樊康生等^[12]提出了一种改进启发函数的 A* 算法,根据地图总面积再结合动态权重因子调整启发函数有效减少了搜索节点,但由于是单向搜索,因此在复杂环境下搜索速度不高。龚云鑫等^[13]提出一种类似于可视关系的邻居关系,以及凸角点的概念,在大地图中,当障碍物较少时,该算法相较于传统的 A* 算法具有更优的搜索效率。然而,在面对障碍物较多的情况下,该算法可能表现较差,所取得凸角点就比较多,搜索时间就较长。LI 等^[14]提出了多步蚁群算法来优化路径,其在规格较小的简单环境下的收敛速度较高,但在复杂环境下寻找效率较低。

为提高传统的 A* 算法效率以及在复杂环境下避障性能较低、搜索速度较慢等问题,采用邻域优化的方法,将传统 A* 算法上的 8 邻域改为 24 邻域,并且在 24

邻域上进一步的优化成 8 邻域,在 8 邻域基础上舍弃掉冗余的节点,对评估函数加入改进的权重系数,再多次使用贝塞尔曲线对路径进行拟合,使路线更加的连续、平滑,同时引入双向搜索机制^[15]。基于以上改进,通过仿真实验验证了改进算法的有效性,为机器人路径规划^[16-18]、自动驾驶系统、人工智能游戏、无人机航线路径规划^[19]等提供了应用价值。

1 系统模型设计与双向 A* 算法改进

1.1 环境建模

在机器人路径规划中,选择合适的地图模型至关重要。常见的构建地图方法包括拓扑地图和栅格地图等。由于栅格地图可以将复杂的环境分割成规则的网络,进而简化对环境的建模和理解,因此本文对机器人移动的环境采用栅格地图对障碍物进行环境的模拟。黑色的圆圈为环境障碍物,剩余未显示的白色圆圈则可以自由移动,粉色叉号为搜索点,红色线段为路线,黑色黑点为障碍物,如图 1 所示。

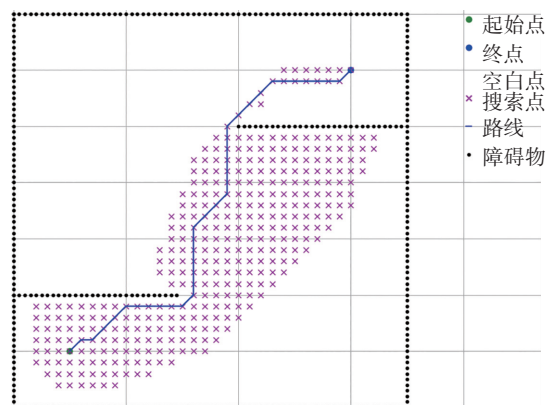


图 1 栅格地图

Fig. 1 Grid map

本文的改进算法采用栅格法进行环境建模,创建了边长为 m 的正方形栅格地图。在建立栅格地图后,对栅格单元进行了线性化处理,其中,单元格的起点位于地图左下角,栅格单元的线性坐标 n_{um} 为

$$n_{um} = x + my \quad (1)$$

式(1)中, x 和 y 表示栅格单元的行和列坐标, m 表示栅格地图的边长。

1.2 改进 A* 算法的描述

传统的 A* 算法通常采用单向搜寻策略,但搜寻节点更多,在复杂环境下相对来说效率不高。本文提出的 A* 算法为提高搜索效率,在从起始节点和目标节点同时展开搜索,采用了双向搜索模式。算法根据启发

式评估函数的值逐步向对方前进,逐渐减少移动过程中的冗余节点。此外,通过设置合理的领域、评估函数以及使用贝塞尔曲线等路径优化技术,可以进一步提高搜索效率。

本文采用双向搜索策略,在此基础上,建立了前向查找和后向查找的 OPEN_A 表, CLOSE_A 表, Current_A 表, 评价函数,并提供 OPEN_B 表, CLOSE_B 表, Current_B 表,评价函数。nstart 和 ngoal 都被用作两次查找的目前节点。具体地,正向搜索在寻找目标节点 ngoal 时,以 nstart 为起始点,获取周围一个最优节点,并存入 Current_A 列表中,再以最优节点进行扩展。此时转向反向搜索,以 ngoal 为起始点, nstart 为目标节点,又获取周围邻域的一个最优节点,并存入 Current_B 列表中,然后根据这个最优节点进行扩展,如此交替反复,直至两个方向的 Current_A 和 Current_B 的纵横坐标同时终止。为了使路径进一步优化,本文还采用了贝塞尔曲线(Bessel Curve)来优化路径。

改进的 A* 算法具体流程如下:

(1) 知道初始点 nstart 被设定成向前的当前节点 Current_A,然后将其添加到 OPEN_A 的名单中。

(2) 通过遍历搜索 Current_A 点附近的所有可达结点,利用前向评估函数求出所有结点的费用,并将其添加至 OPEN_A 表。

(3) 知道起点 ngoal 被设定成向前的 Current_B,然后把它添加到 OPEN_B 的名单。

(4) 遍历搜索 Current_B 点附近的可达结点,利用前向评估函数求出其所有结点的费用,然后将其添加至 OPEN_B 协议中。

(5) 在将初始点 Current_B 作为新添加节点的父结点之后,将其从 OPEN_B 表中移除并添加到 CLOSE_B 表格中。

(6) 遍历搜寻 Current_B 点周围可达到的节点,通过正向评价函数计算该节点的总代价,并且把它们加入 OPEN_B 列表中。将起始点 Current_B 设置为新加入节点的父节点后,再将它从 OPEN_B 表中删除,加入 CLOSE_B 表中。

(7) 每次选取正反向中 OPEN_A、OPEN_B 表中最小代价的节点作为父节点,再从 OPEN 表中删除,加入 CLOSE 表中,然后扩展出最优节点,进行下一次的循环,循环往复的执行,若正向和反向扩展出的最优解点为相同节点,则说明相遇,即中止搜索。

改进的 A* 算法具体流程如图 2 所示。

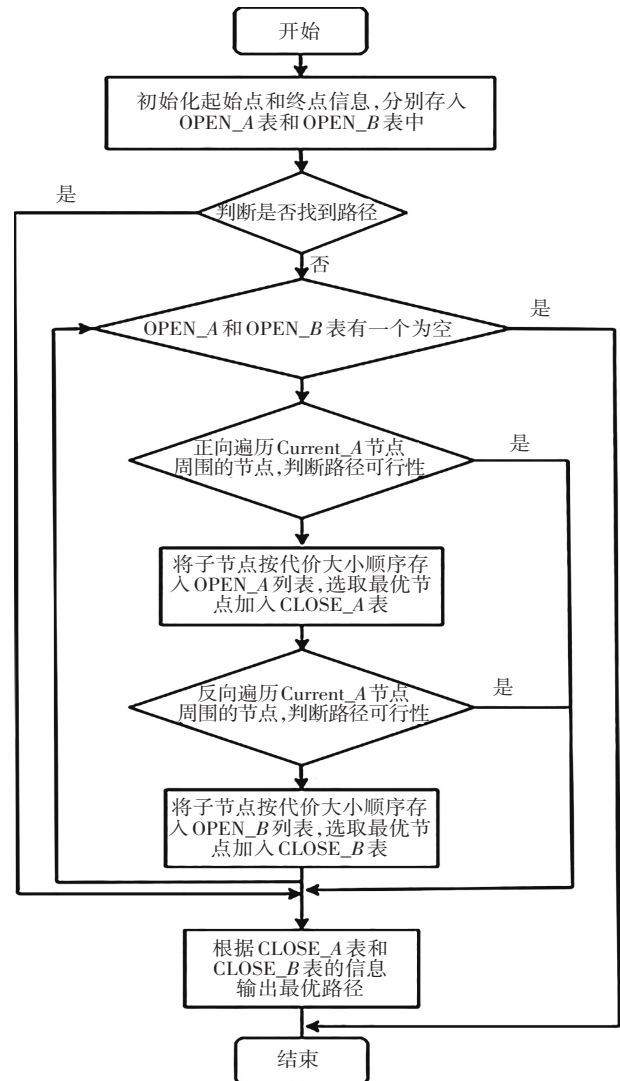


图 2 双向搜索的 A* 算法路径规划流程图

Fig. 2 Flowchart of A* algorithm path planning for bidirectional search

1.3 领域扩展

传统的 A* 算法在路径规划中通常采用 8 邻域的搜索方式,其中 8 邻域指的是节点与其周围的八个相邻节点相连。这种连接方式在离散网格地图的情况下得到广泛应用,特别适用于栅格地图等简单地图结构。8 邻域的定义相对简单,易于实现,适用于基本的路径规划需求,并具有较高的计算效率。

然而,传统的 8 邻域也存在一些缺点。首先,由于包含对角线方向的连接,路径可能在角落处出现剧变,导致路径带有不必要的拐点。其次,在复杂地形下,8 邻域未能充分考虑地形的变化,缺乏对地形细节的灵活性。对于需要更精确路径规划的场景,8 邻域可能无法提供足够的灵活性和精度。

为了解决这些问题,本文将传统的 8 邻域改进为 24 邻域的 8 方向搜索方式,具体形式见图 3。通过将

24 邻域的 8 个方向编号,根据目标点与当前点的相对位置确定目标点所在的象限,再根据连线是否经过障碍物剪枝,将原本的 8 搜索方向剪枝为 6 方向,大大提高了搜索效率。这种改进不仅保留了搜索的基本方向,同时通过剪枝减少了冗余的节点,使算法更灵活、高效。

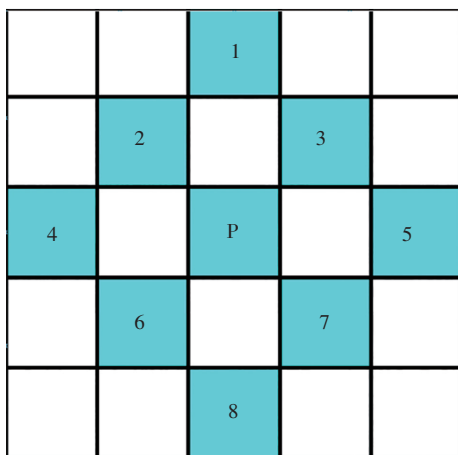


图 3 改进双向 A* 算法的 8 搜索方向图

Fig. 3 Improved bidirectional A* algorithm for 8-search direction graphs

在具体的实现中,通过表格的形式对保留和舍去的节点信息进行整理,以便更清晰地了解搜索方向的调整(表 1)。这样的改进在面对复杂环境和对路径规划精度要求较高的情况下,能够更好地适应地形变化,提高路径规划算法的性能。

表 1 24 邻域改 6 向搜索方向

Table 1 Changing the search directions from 24-neighborhood to 6-direction

目标点所处位置	保留的节点	舍去的节点
第一象限 & 无障碍	P1、P2、P3、P6、P7、P8	P4、P8
第一象限 & 有障碍	P2、P3、P5、P6、P7、P8	P5、P8
第二象限 & 无障碍	P2、P3、P4、P6、P7、P8	P5、P8
第二象限 & 有障碍	P2、P3、P4、P6、P7、P8	P4、P8
第三象限 & 无障碍	P1、P2、P3、P6、P7、P8	P1、P5
第三象限 & 有障碍	P2、P3、P4、P6、P7、P8	P1、P4
第四象限 & 无障碍	P2、P3、P4、P5、P6、P7	P1、P4
第四象限 & 有障碍	P2、P3、P4、P5、P6、P7	P1、P5

1.4 改进的评价函数

传统的评价函数在路径规划等问题中具有简单直观、计算效率高以及广泛应用的优点,尤其在 A* 算法等经典算法中的启发式函数中得到了广泛应用。然而,这些评价函数也存在一些不足之处。它们可能难以适应复杂和动态的环境,无法有效地处理多目标冲突,对问题特定性要求较高,并且在面对实时变动的因素时表现有限。在应对这些挑战的同时,研究人员正在探索更复杂、灵活的评价函数设计,涉及机器学习方

法、多目标考虑、以及对动态环境的适应性,以提高算法的性能和适用性。传统 A* 算法总代价一般表达式为

$$f(n) = s(n) + t(n) \quad (2)$$

式(2)中, $s(n)$ 是用来表示从初始节点 n_{start} 到节点 n 的实际代价,而 $t(n)$ 则是用来估计从节点 n 到目标节点的最佳路径的代价。在这里, $s(n)$ 的已知信息使得 $t(n)$ 更具启发性,而 $s(n)$ 则更能体现搜索在广度上的优先偏向。

A* 算法通常使用曼哈顿距离、欧几里德距离和切比雪夫距离作为启发函数。曼哈顿距离计算简单,只需对坐标差值取绝对值并相加,但是它仅仅考虑 4 个方向,没有斜向移动,不够准确,特别是在实际问题中需要考虑对角线方向的距离,切比雪夫距离对角线距离考虑了直线路径和对角线路径,适用于需要考虑多个方向性的问题,这在实际应用场景下更为合适,而欧几里德可能过度估计了两点之间的距离,在复杂环境中计算欧几里德距离可能会变得计算复杂度高。本文采用的就是切比雪夫距离表示启发函数,其公式为

$$dx = |x_n - x_{goal}| \quad (3)$$

$$dy = |y_n - y_{goal}| \quad (4)$$

$$\min xy = \min(dx, dy) \quad (5)$$

$$h(n) = dx + dy + (\sqrt{2} - 2) \cdot \min xy \quad (6)$$

其中: x_n 为当前节点的横坐标, y_n 为当前节点的纵坐标, x_{goal} 为目标节点的横坐标, y_{goal} 为目标节点的纵坐标, dx 表示 x 方向的坐标差, dy 表示 y 方向的坐标差, $\min xy$ 表示 dx 和 dy 比较后的最小值。

在改进前的 A* 算法由于在起点与终点之间有许多障碍物,因此路径实际代价大于 $h(n)$,所以导致搜索空间变大,效率降低。若 $h(n)$ 代价大于实际路径,则会导致所规划的路径难以成为最短路径。为了进一步的提高搜索效率,本文对启发函数 $h(n)$ 进行动态的权重优化,即进入 $w(n)$ 权重函数。

(1) 正向搜索估价函数如下:

$$f(n1) = g(n1) + w(n1) * h(n1) \quad (7)$$

其中, $w(n1)$ 表示节点 $n1$ 的权重,即评估函数的权重项,由于搜索过程中不能只考虑搜索速度而不考虑规划的路径,所以采用动态加权的方式,即在放弃搜索最优路径的情况下,使用动态加权来缩短搜索的时间。在搜索开始时,快速到达目的地所在区域更重要。以 $h(n1)$ 为判断依据,将它声明为 d ,当 $d > d1$ 时, $w = w1$,此时算法搜索速度更快;当 $d \leq d1$ 时, $w = w2$,也就是接近终点的时候,优先考虑最优路径。其中 $f(n1)$ 表示正向搜索的评估函数,而 $h(n1)$ 则是以反向搜索的当前节点为目标点的启发式函数,采用切比雪夫距离。

(2) 反向搜索估价函数如下:

$$f(n2) = g(n2) + w(n2) * h(n2) \quad (8)$$

其中, $w(n2)$ 表示节点 $n2$ 的权重, 即评估函数的权重项, 其 $w(n2)$ 与正向搜索中的取值相同。 $f(n2)$ 表示反向搜索的评估函数, 而 $h(n2)$ 则是以正向的当前节点为目标点的启发式函数, 采用切比雪夫距离。

1.5 评价函数的参数确定

评价函数中 w 与 h 的值要根据自己设定地图的大小、复杂程度进行多次调节, 因此以 140×140 大小的地图为例, 通过多次实验发现当 $d1$ 等于 11.9, $w1$ 等于 5 时候搜索节点最佳, 如表 2 下划线所示。

表 2 $w1$ 和 $d1$ 参数的确定

Table 2 Determination of parameters $w1$ and $d1$

$w1$	$d1$					
	11.8	11.9	12	12.1	12.2	12.3
0	747	747	744	744	744	746
1	406	410	402	402	402	402
2	274	259	284	284	284	285
3	158	158	211	211	211	335
4	156	139	141	141	141	139
5	148	<u>122</u>	188	188	188	137
6	195	194	224	224	224	195
7	178	180	184	184	184	181
8	128	179	171	171	171	145
9	124	124	172	172	172	146
10	125	141	138	138	138	145
11	222	128	134	134	134	126
12	226	128	134	134	134	126

随后采用控制变量法, 将这两个参数固定后, 再找出 $w2$ 的最佳值, 如表 3 所示, 当 $w2$ 等于 2 左右时候最佳, 然后在 2 左右以 0.1 的间隔去试验, 最终得出当 $w1 = 5, w2 = 1.8, d1 = 11.9$ 时为最优解。

表 3 $w2$ 参数的确定

Table 3 Determination of the parameter $w2$

$w2$	搜索节点
0	362
1	199
2	127
3	247
4	153
5	240
6	187
7	551
8	300
9	227
10	181
11	179
12	201

1.6 贝塞尔曲线的优化

贝塞尔曲线是通过一组控制点实现了对折线的平滑处理, 本文着重介绍了利用贝塞尔曲线进行路径平滑的方法。贝塞尔曲线的高阶通用公式如下:

$$B(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i P_i \quad (9)$$

其中, t 是参数, 取值范围在 $[0, 1]$ 之间。 $\binom{n}{i}$ 是组合数, 表示从 n 个元素中选择 i 个元素的组合方式。公式含义为曲线上的点 $B(t)$ 是由控制点 P_0, P_1, \dots, P_n 按照二项式系数加权组合而成的。权重由 $(1-t)^{n-i}$ 控制。

高阶贝塞尔曲线在处理复杂路径时可能出现稳定性较差的情况, 导致路径穿越障碍物, 并且曲线效果不理想。为解决这问题, 采用多组三阶贝塞尔曲线对路径进行优化。

三阶贝塞尔曲线公式为

$$B(t) = (1-t)^3 \cdot P_0 + 3 \cdot (1-t)^2 \cdot t \cdot P_1 + 3 \cdot (1-t) \cdot t^2 \cdot P_2 + t^3 \cdot P_3 \quad (10)$$

拟合效果如图 4 所示。

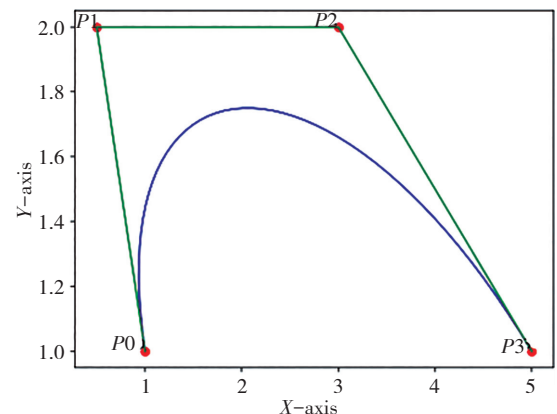


图 4 三阶贝塞尔曲线

Fig. 4 Third-order Bézier curve

2 仿真实验与结果分析

2.1 仿真实验环境

在实验证实了改进双向 A* 算法的领域扩展数量、评价函数参数以及贝塞尔曲线的优化后, 进行了改进双向 A* 算法和传统 A* 算法的比较。分别比较了它们在遍历节点数量、运行时间和路径长度方面的表现, 并对这两种算法在不同环境下的性能进行了分析。最终, 评估了改进双向 A* 算法的效果。首先在小尺寸环境中可能会出现过拟合情形, 而在大尺寸环境中泛化能力较差。其次, 在不同环境中可以验证算法的稳定性, 可以更加全面地评估不同算法的性能, 最后实际场景中环境大小各不相同, 都有可能是机器人需要应对的场景。因此仿真分别在 $70 \text{ m} * 70 \text{ m}$ 、 $140 \text{ m} * 140 \text{ m}$ 、 $210 \text{ m} * 210 \text{ m}$ 的环境地图进行, 然后再分别进行传统 A* 算法和改进双向

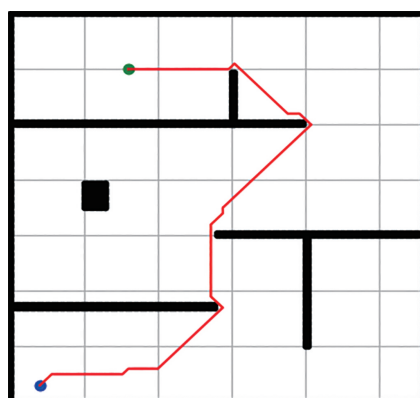
A* 算法进行仿真。为了准确的评估算法的执行效率,本实验所用的电脑显卡为 13th Gen Intel (R) Core (TM) i7-13650HX,软件为 PyCharm2023。

在地图中绿色圆圈表示起始点,蓝色圆圈表示目标点,黑色圆圈表示障碍物,粉色×表示搜索节点。每一个栅格大小对应实际地图大小为 2 m,机器人的半径为 1 m。其中,140 m * 140 m 地图中起始点和目标点坐标为(10,6)和(75,120),210 m * 210 m 地图中起始点和目标点坐标为(10,6)和(50,190),红色线段表示起始点到目标点的路线。

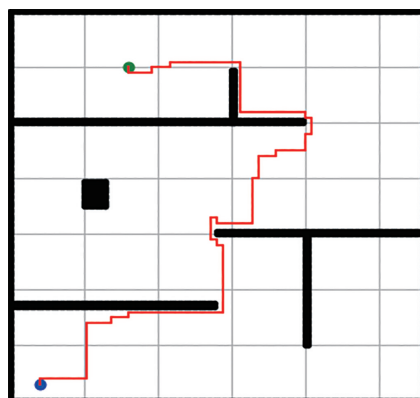
由图 5—图 7 可见:改进双向 A* 算法在不同较为复杂的环境下所遍历的节点明显减少很多,而且搜索的效率提高显著,为计算机减少了大量的计算,而且对比传统 A* 算法的路径存在拐点较多的情况,在改进的双向 A* 算法中利用多组三阶贝塞尔曲线也得到了优化,可以看出改进后的路线的拐点明显减少。尽管传统的 A* 算法在最短路径上可能会略优于改进后的双向,但在实际场景下改进后算法的全面的搜索效率更好,例如在火灾现场救援行动,化学厂爆炸现场,以及煤气泄漏场景下等危急关头下,搜索效率就显得格外重要。本文算法与 4 种算法在两种不同地图尺寸下的部分仿真如图 6 所示。

2.2 140×140 尺寸下的路径规划测试

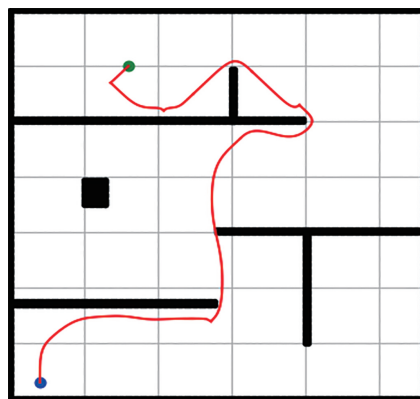
通过对比图 5 和表 4 的实验结果分析,可以得知在地图尺寸为 140×140 且障碍物较为复杂的情况下,改进的双向 A 算法在相对路径长度上虽然没有明显优势,但相较于传统的 A 算法,规划时间减少了 92.4%,搜索节点显著减少约为 95.3%。与 Dijkstra 算法相比,由于 Dijkstra 没有启发函数进行优化,引入了大量的扩展节点,搜索效率比传统的 A* 算法更慢。文献[20]中的 A* 算法采用的是单向搜索方式,没有采用动态函数,这导致了路径虽然是最佳值,但在搜索节点和搜索速度上面不如本算法,从表 4 可以看出加入动态函数后的必要性。



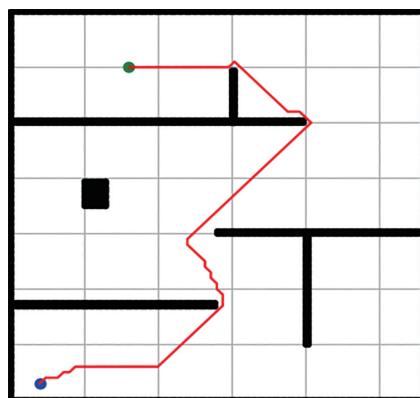
(b) 传统的A* 算法



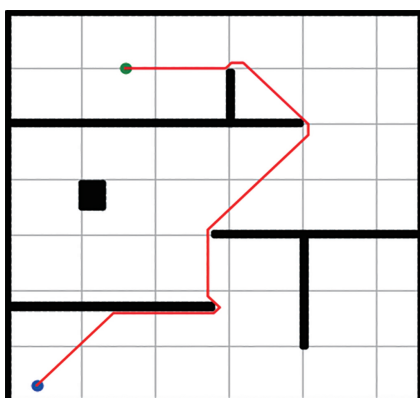
(c) 文献[9]算法



(d) 本文算法



(e) 文献[1]算法



(a) Dijkstra 算法

图 5 140×140 尺寸的路径对比

Fig. 5 Path comparison on a 140×140 map

表 4 五个算法的对比表(140×140)

Table 4 Comparison table of five algorithms(140×140)

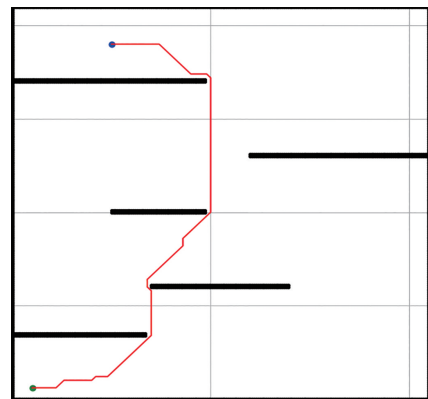
算法类型	相对路径长度/m	规划时间/ms	搜索节点个数	拐点个数
传统的A*算法	227.28	14.09	2 055	14
Dijkstra 算法	227.28	42.69	4 347	10
文献[11]算法	294.21	4.12	410	34
本文算法	285.34	1.97	206	4
文献[20]算法	233.91	8.59	1 373	22

从搜索节点和规划时间的角度来看,本文算法取得了显著提升,弥补了传统 A* 算法和 Dijkstra 演算法的不足,同时对权重优化功能进行了进一步改进,使得该算法比传统 A* 算法更好地规划了时间和搜寻节点。值得注意的是,双向 A 算法通常使用 8 邻域的邻域扩展,导致在面对复杂环境时搜索范围较小且速度较慢。而本文算法将原先的 8 邻域改为 24 邻域的 8 搜索方向,扩大了搜索范围,并通过根据目标点位置进行剪枝,舍去不必要的扩展节点,以提高搜索效率,最终得到 6 个搜索方向。从表 3 可以看出,无论是相对路径长度、规划时间还是搜索节点数量,本文算法都取得了显著提升。

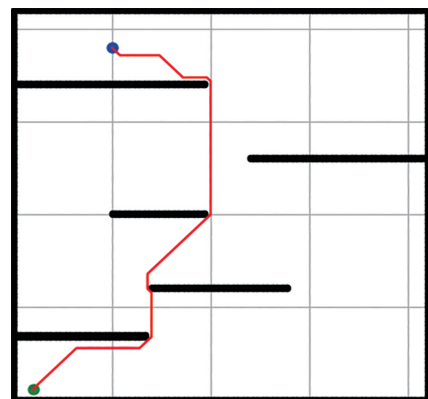
此外,从拐点数量的角度来看,本文算法具有最少的拐点数量,相较于其他算法提升了至少 50%。这更加证明了本文算法在复杂环境下的优越性。最后,对实验结果进行降重和润色,进一步突显了改进算法在路径规划任务中的出色性能。

2.3 210×210 尺寸下的路径规划测试

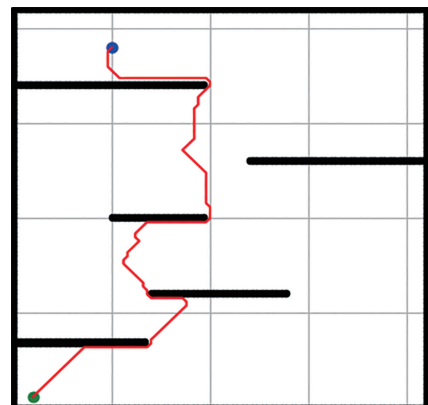
通过图 6 和表 5 的实验结果分析,在 210×210 和 140×140 地图尺寸的对比中,本文算法在面对障碍物多且地图尺寸较大的环境中展现了更小的搜索节点和规划时间。相较于传统的双向 A* 算法,改进效果更为显著,这归因于本文采用了更加灵活的邻域扩展策略。然而,值得注意的是,在相对路径长度方面,本文算法不如传统 A* 算法,这是因为双向 A* 算法虽然从两个方向扩展,但每次总体代价的计算是基于两个当前位置的计算,因此在全局最优路径上的效果相对较差。尽管存在相对路径长度的不足,但在实际应用中,特别是在障碍物多且大尺寸环境下,本文算法仍然是最优选择。这是因为本文算法在运行速度和搜索效率上表现出色,采用了更加高效的邻域扩展方法。



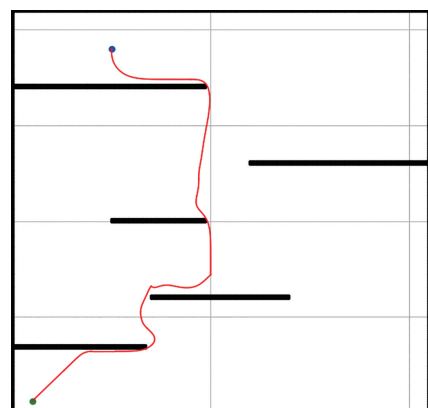
(a) Dijkstra 算法



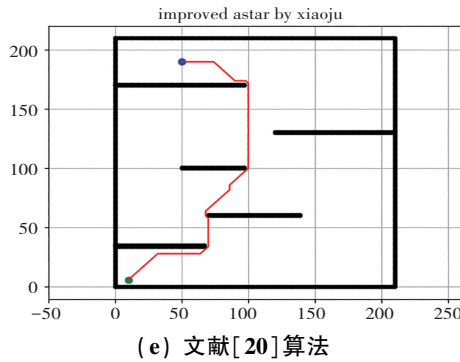
(b) 传统A*算法



(c) 文献[9]算法



(d) 本文算法



(e) 文献[20]算法

图 6 210×210 尺寸的路径对比图

Fig. 6 Path comparison on a 210×210 map

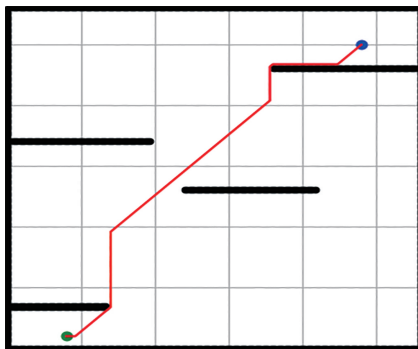
表 5 五种算法的对比表(210×210)

Table 5 Comparison table of five algorithms(210×210)

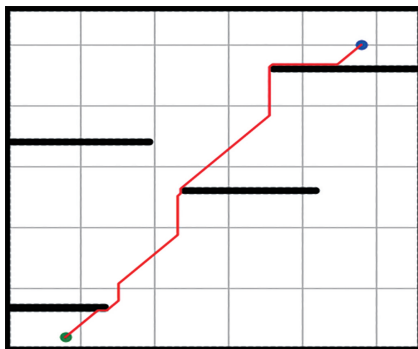
算法类型	相对路径长度/m	规划时间/ms	搜索节点	拐点个数
传统的A*算法	281.14	45.57	4 260	10
Dijkstra 算法	281.14	162.12	9 623	13
文献[11]算法	339.87	3.35	290	20
本文算法	331.08	2.53	65	4
文献[20]算法	318.79	10.83	1 480	13

2.4 280×280 尺寸下的路径规划测试

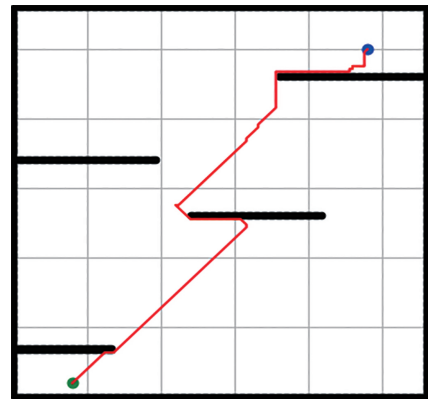
通过图 7 和表 6 的实验结果分析,在 280×280 地图尺寸中,本文算法的算法效率依旧较高,在搜索节点上比文献[20]高出约 90%,比文献[11]高出约 77%,在规划时间上也提升显著,虽然相对路径长度比传统 A* 算法和 Dijkstra 算法低,但往往在实际场景中,其搜索效率上排在首位。



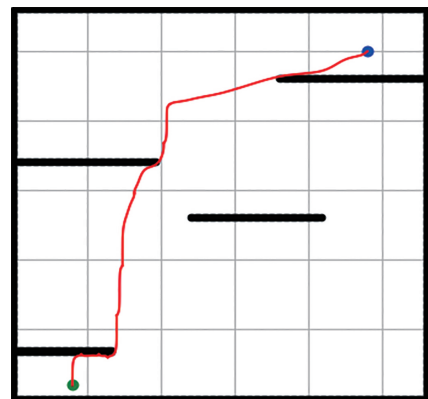
(a) Dijkstra 算法



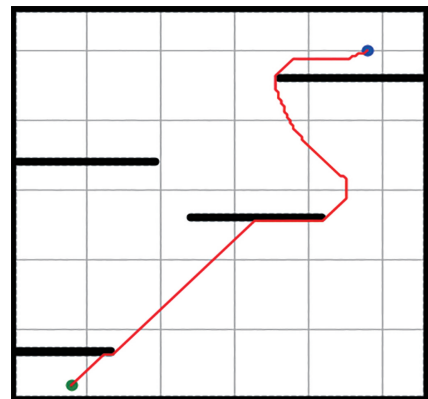
(b) 传统A*算法



(c) 文献[9]算法



(d) 本文算法



(e) 文献[1]算法

图 7 280×280 尺寸的路径对比图

Fig. 7 Path comparison on a 280×280 map

表 6 五种算法的对比表(280×280)

Table 6 Comparison table of five algorithms(280×280)

算法类型	相对路径长度/m	规划时间/ms	搜索节点	拐点个数
传统的A*算法	352.13	70.94	5 422	10
Dijkstra 算法	352.13	856.72	18 487	7
文献[11]算法	422.36	6.61	302	14
本文算法	415.08	4.80	86	4
文献[20]算法	420.01	17.19	1 650	20

在一些紧急情况下,如火灾、煤气泄漏、化工厂爆炸等危险环境,时间至关重要。在这些情况下,最优路径的相对长度可能不如搜索效率重要。因此,在争分夺秒的情境中,本文算法的高效搜索能力成为至关重要的因素。根据表 4—表 6 的数据,可以看出本文算法在地图尺寸越大且环境越复杂的情况下,搜索效率表现越为突出。通过贝塞尔曲线的优化,算法的安全性得到了进一步保障,具备了良好的鲁棒性和稳定性。这使得本算法在处理大规模且复杂环境的路径规划任务中展现出卓越的性能。最终,通过精炼和简化,突显了算法在实际应用中的实用性和卓越性。

3 结 论

移动机器人路径规划一直是人工智能领域备受关注的研究方向。在危险且复杂的环境中,寻找最优路径是至关重要的。传统的 A* 算法通常用于规划最短路径,然而,在实际环境中,由于存在复杂的障碍物和拐点,为了解决 A* 算法可能产生大量不必要节点而导致路径不够安全的问题,提出了基于 24 邻域的 8 方向搜索方式的改进双向 A* 算法,因为有复杂的障碍物和拐点。该算法引入了双向目标节点,优化了启发函数权重,并对原先的 8 搜索方向进行了剪枝,同时使用贝塞尔曲线进行路径的优化处理。实验结果显示:改进后的 A* 算法不仅在搜索节点个数上超越了传统的 A* 算法,从而能够更快地找到最优路径,而且在路径的平顺性和规划时间上也有优异的表现。考虑到化工厂等复杂环境中的路径规划问题,下一步的研究方向将集中在如何快速而准确地找出最短路径的方法上,这一改进对于提升路径规划的效率和安全性具有重要意义。

参考文献(References):

- [1] 刘建杰, 劳睿腾, 阮振荣, 等. 基于 AGV 的集约化智能喂装备[J]. 科技创新与应用, 2023, 13(25): 50-53.
LIU Jian-jie, LAO Rui-teng, RUAN Zhen-rong, et al. Design of intensive intelligent feeding equipment based on AGV[J]. Technology Innovation and Application, 2023, 13(25): 50-53.
- [2] 余娜娜, 李铁克, 王柏琳, 等. 自动化分拣仓库中多 AGV 调度与路径规划算法[J]. 计算机集成制造系统, 2020, 26(1): 171-180.
- [3] 韦焱, 刘晓东. 集装箱自动化码头 AGV 带时间约束的路径规划研究[J]. 新型工业化, 2016, 6(2): 41-45.
WEI Yi, LIU Xiao-dong. Research of routing schedule of AGV in automated container terminal and subject to time constraints[J]. The Journal of New Industrialization, 2016, 6(2): 41-45.
- [4] 王迈新, 闫莉, 李雨菲. 基于改进 A* 算法的车间 AGV 路径规划[J]. 自动化与仪表, 2023, 38(11): 45-49, 59.
WANG Mai-xin, YAN Li, LI Yu-fei. Path planning of AGV in workshop based on improved A* algorithm[J]. Automation & Instrumentation, 2023, 38(11): 45-49, 59.
- [5] 刘云翔, 杜杰, 张晴. 基于路径优化的 A* 算法与 Dijkstra 算法的性能比较[J]. 现代电子技术, 2017, 40(13): 181-183, 186.
LIU Yun-xiang, DU Jie, ZHANG Qing. Performance comparison between A* algorithm and Dijkstra algorithm based on path optimization[J]. Modern Electronics Technique, 2017, 40(13): 181-183, 186.
- [6] FRANSEN K, VAN EEKELEN J. Efficient path planning for automated guided vehicles using A* (Astar) algorithm incorporating turning costs in search heuristic[J]. International Journal of Production Research, 2023, 61(3): 707-725.
- [7] 陶浪, 马昌喜, 朱昌锋, 等. 基于遗传算法的定制公交线路多目标优化[J]. 兰州交通大学学报, 2018, 37(2): 31-37.
TAO Lang, MA Chang-xi, ZHU Chang-feng, et al. Multiobjective optimization of customized bus routes based on a genetic algorithm[J]. Journal of Lanzhou Jiaotong University, 2018, 37(2): 31-37.
- [8] 王洪斌, 郝策, 张平, 等. 基于 A* 算法和人工势场法的移动机器人路径规划[J]. 中国机械工程, 2019, 30(20): 2489-2496.
WANG Hong-bin, HAO Ce, ZHANG Ping, et al. Path planning of mobile robots based on A* algorithm and artificial potential field algorithm[J]. China Mechanical Engineering, 2019, 30(20): 2489-2496.

- [9] 罗梓瑄, 刘学文. 基于蚁群算法的物流配送路径优化研究[J]. 重庆工商大学学报(自然科学版), 2020, 37(4): 89-94.
LUO Zi-xuan, LIU Xue-wen. Research on logistics distribution path optimization based on ant swarm algorithm[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2020, 37(4): 89-94.
- [10] 刘生伟, 马钺, 孟树峰, 等. 改进 A* 算法的 AGV 路径规划[J]. 计算机应用, 2019, 39(S2): 41-44.
LIU Sheng-wei, MA Yue, MENG Shu-feng et al. Improved A* algorithm for AGV path planning [J]. Computer Applications, 2019, 39(S2): 41-44.
- [11] 王中玉, 曾国辉, 黄勃. 基于改进双向 A* 的移动机器人路径规划算法[J]. 传感器与微系统, 2020, 39(11): 141-143, 147.
WANG Zhong-yu, ZENG Guo-hui, HUANG Bo. Mobile robot path planning algorithm based on improved bidirectional astar[J]. Transducer and Microsystem Technologies, 2020, 39(11): 141-143, 147.
- [12] 樊康生, 杨光永, 黄训爱, 等. 一种改进启发函数的 A* 算法[J]. 扬州大学学报(自然科学版), 2023, 26(5): 52-57.
FAN Kang-sheng, YANG Guang-yong, HUANG Xun-ai, et al. An improved heuristic function A* algorithm[J]. Journal of Yangzhou University (Natural Science Edition), 2023, 26(5): 52-57.
- [13] 龚云鑫, 刘桂华, 张文凯, 等. 利用凸角点改进 A* 算法的路径规划方法[J]. 计算机工程与应用, 2023, 59(12): 309-315.
GONG Yun-xin, LIU Gui-hua, ZHANG Wen-kai, et al. Path planning method of improving A* algorithm using convex corner[J]. Computer Engineering and Applications, 2023, 59(12): 309-315.
- [14] LI D, WANG L, CAI J, et al. Research on terminal distance index-based multi-step ant colony optimization for mobile robot path planning[J]. IEEE Transactions on Automation Science and Engineering, 2023, 20(4): 2321-2337.
- [15] 郝振国, 王玉玫. 双向 A* 算法在军事路径规划中的应用[J]. 计算机工程与应用, 2011, 47(29): 246-248.
HAO Zhen-guo, WANG Yu-mei. Application of bidirectional A* method in military route planning[J]. Computer Engineering and Applications, 2011, 47(29): 246-248.
- [16] 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, 42(3): 132-140.
CHI Xu, LI Hua, FEI Ji-you. Research on robot random obstacle avoidance method based on fusion of improved A* algorithm and dynamic window method[J]. Chinese Journal of Scientific Instrument, 2021, 42(3): 132-140.
- [17] 张润梅, 任瑞, 袁彬, 等. 装配式建筑机器人施工路径优化方法[J]. 计算机工程与设计, 2021, 42(12): 3516-3524.
ZHANG Run-mei, REN Rui, YUAN Bin, et al. Construction path optimization method of prefabricated building robot[J]. Computer Engineering and Design, 2021, 42(12): 3516-3524.
- [18] 韩学行, 顿向明, 林子洋. 基于 A* 改进算法的机器人移动路径优化仿真[J]. 计算机仿真, 2021, 38(2): 313-317.
HAN Xue-xing, DUN Xiang-ming, LIN Zi-yang. Simulation of robot mobile path optimization based on A* improved algorithms[J]. Computer Simulation, 2021, 38(2): 313-317.
- [19] 刘凯, 秦锋, 徐浩, 等. 基于多旋翼无人机的输气管道巡检路径规划研究[J]. 重庆工商大学学报(自然科学版), 2021, 38(6): 34-41.
LIU Kai, QIN Feng, XU Hao, et al. Research on path planning of gas pipeline inspection based on multi-rotor UAV[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2021, 38(6): 34-41.
- [20] KRISHNAMOHAN T. Computing the runs that should be scored every over when chasing a target in limited-overs cricket using the A* algorithm[J]. Applied Artificial Intelligence, 2021, 35(15): 2087-2101.
- [21] 王保剑, 胡大裳, 蒋玉明. 改进 A* 算法在路径规划中的应用[J]. 计算机工程与应用, 2021, 57(12): 243-247.
WANG Bao-jian, HU Da-sha, JIANG Yu-ming. Application of improved A* algorithm in path planning[J]. Computer Engineering and Applications, 2021, 57(12): 243-247.

责任编辑:陈芳