

# 基于改进A\*算法的全局路径规划

武延涛<sup>1</sup>, 张震<sup>1</sup>, 刘伟良<sup>1</sup>, 卜予涵<sup>2</sup>

(1. 西安工业大学 机电工程学院, 陕西 西安 710021;

2. 西交利物浦大学 经济与管理学院, 江苏 苏州 215028)

**摘要:** 提出一种改进A\*算法, 用于解决传统A\*算法面对“U型”障碍物时路径规划效率低的问题。改进后的算法通过改变节点搜索条件, 减少无用节点的搜索; 利用摸墙算法绕过规划路径中的障碍物, 提取路径中的关键转折节点, 结合Flody算法思想对路径进行优化, 以及对优化后的路径后处理, 以提高路径规划效率和降低路径长度。仿真结果表明, 改进后的A\*算法在搜索效率上比传统A\*算法提高了63.5%, 在路径长度上, 减短了2.42%。

**关键词:** 改进A\*算法; U型; 摸墙算法; Flody算法; 路径规划

中图分类号: TP242; TP18 文献标识码: A 文章编号: 1003-7241(2025)03-0012-05

## Global Path Planning Based on Improved A\* Algorithm

WU Yan-tao<sup>1</sup>, ZHANG Zhen<sup>1</sup>, LIU Wei-liang<sup>1</sup>, BU Yu-han<sup>2</sup>

(1. School of Electromechanical Engineering, Xi'an Industrial University, Xi'an 710021 China;

2. School of Economics and Management, Xi'an Jiaotong-liverpool University, Suzhou 215028 China )

**Abstract:** This paper proposes an improved A\* algorithm to solve the problem of low path planning efficiency when the traditional A\* algorithm faces the "U-shaped" obstacles. The improved algorithm reduces the search of useless nodes by changing the node search conditions, uses the wall-following algorithm to bypass the obstacles in the planned path, extracts the key turning nodes in the path, combines the Flody algorithm idea to optimize the path, and processes the optimized path to improve the path planning efficiency and reduce the path length. Simulation results show that the improved A\* algorithm improves the search efficiency by 63.5% compared with the traditional A\* algorithm, and reduces the path length by 2.42%.

**Keywords:** improved A\* algorithm; U-shape; wall-touching algorithm; Flody algorithm; path planning

## 0 引言

近年来, 人工智能对人民的生活、经济的发展和社会的进步产生深远的影响, 而路径规划技术作为人工智能的重要分支, 成为国内外研究热点。

路径规划, 即在有障碍物的环境中规划一条使移动机器人从起点位置到达终点位置的安全路径<sup>[1-2]</sup>。依据对工作环境信息的认知程度, 路径规划可划分为局部路径规划和全局路径规划<sup>[3]</sup>, 全局路径规划<sup>[4]</sup>是在工作环境信息已知的条件下进行路径规划, 应用于全局路径规划的常用算法有A\*算法<sup>[5]</sup>、RRT算法<sup>[6]</sup>、蚁群算法<sup>[7]</sup>、粒子群算法<sup>[8]</sup>、遗传算法<sup>[9]</sup>等。其中, A\*算法因其算法简单、易实现等优点, 被国内外学者不断研究和改进。但A\*算法在面对“U型”障碍物进行路径规划时, 易陷入“U型”障碍物底部, 导致搜索效率降低。

本文针对传统A\*算法在面对“U型”障碍物规划时存

在的缺陷, 提出利用摸墙算法思想对传统A\*算法进行改进, 旨在克服陷入“U型”障碍物底部的局限性、提高搜索效率同时兼顾路径长度。

## 1 传统A\*算法

传统A\*算法<sup>[10]</sup>是一种启发式搜索算法, 该算法利用代价函数求取节点代价值作为寻路依据, 以获取最优路径<sup>[11]</sup>, 其代价函数方程为:

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

式中:  $f(m)$ 为起始节点经过当前节点到达目标节点的总代价;  $g(m)$ 为起始节点到达当前节点的实际路径代价,  $h(m)$ 为当前节点到达目标节点的估计代价。

A\*算法找到最优解的关键在于估计代价  $h(m)$  的计算, 主要有两种计算方式: 欧几里得距离和曼哈顿距离。为了降低计算难度, 本文选用曼哈顿距离对  $h(m)$  进行求取, 如公式(2)所示。

$$h(m) = |x_e - x_m| + |y_e - y_m| \quad (2)$$

式中:  $(x_m, y_m)$ 为当前节点坐标;  $(x_e, y_e)$ 为目标节点坐标。

\*基金项目: 陕西省教育厅专项科研计划项目 (21JK0679); 陕西省自然科学基金基础研究计划项目 (2023-JC-YB-386)

收稿日期: 2024-09-02

传统A\*算法寻路思路为:从起点开始,将起点作为一个父节点并记入CLOSE列表中,利用代价函数方程求取周边8个节点中的可用节点代价值,并将节点及相应代价值记入OPEN列表,将OPEN列表中 $f(m)$ 最小的节点作为下一个父节点,重复上述过程,不断地存储和更新OPEN、CLOSE列表,直到目标节点存入CLOSE列表结束。

从寻路过程可知传统A\*算法存在的弊端:

(1) 在寻路过程中,对父节点周边的8个可用节点代价值进行计算,以确定下一个父节点,导致无用节点搜索过多,浪费搜索资源,使得运行时间变长;

(2) 当面对“U型”障碍物时,需要将U型内部的大量节点进行往返搜索才能跳出U型障碍物,导致计算量大,内存占用多,搜索效率变低;

(3) 路径规划过程中,受 $\pi/4$ 角度搜索的局限性,规划的路径会存在较多的转折点和冗余点,阻碍移动机器人运动的流畅性。

## 2 改进A\*算法

### 2.1 改进扩展节点搜索条件

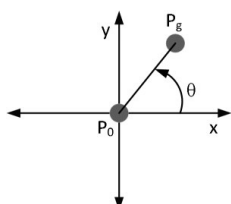
针对传统A\*算法对父节点周边的8个可用节点进行搜索,导致无用节点搜索过多问题。本文提出改进扩展节点搜索条件<sup>[12]</sup>,即:根据中心节点和目标节点的相对位置确定待扩展的节点,减少不必要节点的搜索。

(1) 将中心节点周围的8个栅格命名

以 $P_0$ 为中心节点,沿逆时针将周围的八个栅格依次命名为S1、S2、S3、S4、S5、S6、S7、S8。如图1(a)所示。



(a) 栅格命名图



(b) 偏转角度图

图1 扩展节点搜索条件图

(2) 确定扩展节点搜索条件

以x轴正方向为基准,计算目标节点坐标 $P_g(x_g, y_g)$ 相对于中心节点坐标 $P_0(x_0, y_0)$ 的偏转角度,如图1(b)所示,计算公式如下:

$$\Delta x = x_g - x_0 \quad (3)$$

$$\Delta y = y_g - y_0 \quad (4)$$

$$\theta_0 = \arcsin \frac{\Delta y}{\sqrt{(\Delta x)^2 + (\Delta y)^2}} \quad (5)$$

$$\theta = \begin{cases} 180^\circ - \theta_0 & (\Delta x < 0, \Delta y \geq 0) \\ -180^\circ - \theta_0 & (\Delta x < 0, \Delta y < 0) \\ \theta_0 & \text{else} \end{cases} \quad (6)$$

式(3)~式(6)中: $(\Delta x, \Delta y)$ 为目标节点与中心节点坐标的差值; $(x_0, y_0)$ 为中心节点坐标; $(x_g, y_g)$ 为目标节点坐标; $\theta_0$ 为未处理偏转角度; $\theta$ 为处理后真实偏转角度。根据偏转角度和表1确定需要搜索的扩展节点。

经过改进扩展节点的搜索条件,每次搜索1个或2个节点,减少了搜索栅格的数量。

### 2.2 利用摸墙算法改进A\*算法

针对传统A\*算法面对“U型”障碍物,易陷入“U型”障碍物底部的局限性,本文提出利用摸墙算法绕过障碍物,避免A\*算法在底部往返搜索,提高搜索效率。

摸墙算法即运用左手或者右手法则绕出迷宫的初级算法,利用摸墙算法绕过路径中的障碍物流程如下所示,其中InitPath列表即为在忽视障碍物的条件下,利用改进节点搜索条件的A\*算法规划从起点到终点的路径坐标。

表1 偏转角度与搜索节点对应关系

偏转角度 $\theta$	搜索两个节点	偏转角度 $\theta$	搜索单个节点
$-180^\circ < \theta < -135^\circ$	S5、S6	$\theta = -135^\circ$	S6
$-135^\circ < \theta < -90^\circ$	S6、S7	$\theta = -90^\circ$	S7
$-90^\circ < \theta < -45^\circ$	S7、S8	$\theta = -45^\circ$	S8
$-45^\circ < \theta < 0^\circ$	S8、S1	$\theta = 0^\circ$	S1
$0^\circ < \theta < 45^\circ$	S1、S2	$\theta = 45^\circ$	S2
$45^\circ < \theta < 90^\circ$	S2、S3	$\theta = 90^\circ$	S3
$90^\circ < \theta < 135^\circ$	S3、S4	$\theta = 135^\circ$	S4
$135^\circ < \theta < 180^\circ$	S4、S5	$\theta = 180^\circ$	S5

(1) 确定障碍物位于当前节点的方向

以x轴正方向为基准,根据式(3)~式(6)计算Init-Path列表中障碍物节点 $P_{i+1}$ 相对于当前节点 $P_i$ 的偏转角度,根据表2确定障碍物所在方向。

表2 障碍物方向与角度对应关系

偏转角度 $\theta$	障碍物所在方向
$-45^\circ < \theta \leq 45^\circ$	东边
$45^\circ < \theta \leq 135^\circ$	北边
$135^\circ < \theta \leq 180^\circ$ 或者 $-180^\circ < \theta \leq -135^\circ$	西边
$-135^\circ < \theta \leq -45^\circ$	南边

(2) 摸墙算法流程图

图2中 $(x+i, y+j)$ 为当前节点,其东边节点为 $(x+i+1, y+j)$ ,其北边节点为 $(x+i, y+j+1)$ ,其西边节点为 $(x+i-1, y+j)$ ,其南边节点为 $(x+i, y+j-1)$ , $i=i+1$ 表示向东移动一格, $j=j+1$ 表示向北移动一格, $i=i-1$ 表示向西移动一格, $j=j-1$ 表示向南移动一格,阈值的选取是保证无法到达目标地点时及时退出搜索的重要指标,通过大量实验验证,当阈值为栅格地图周长的 $1/4$ 时较为合适。摸墙算法存在左手和右手摸墙之分,图2为左手摸墙算法,右手摸墙算法将图2中北边流程与南边流程互换位置即可。

为了提高路径的安全性,对利用摸墙算法绕过障碍物的路径进行关键转折节点<sup>[13]</sup>的提取,提取思路为:将路径的节点进行连线,对路径内的节点k,依次计算k-1~k与k~k+1组成线段的斜率,若斜率相等,则去除节点k;

如斜率不相等,则此节点k为关键转折节点,保留节点k。图3为利用左手摸墙算法和右手摸墙算法躲避障碍物规划的路径,其中围绕障碍物的深灰色栅格即为提取的关键转折节点。

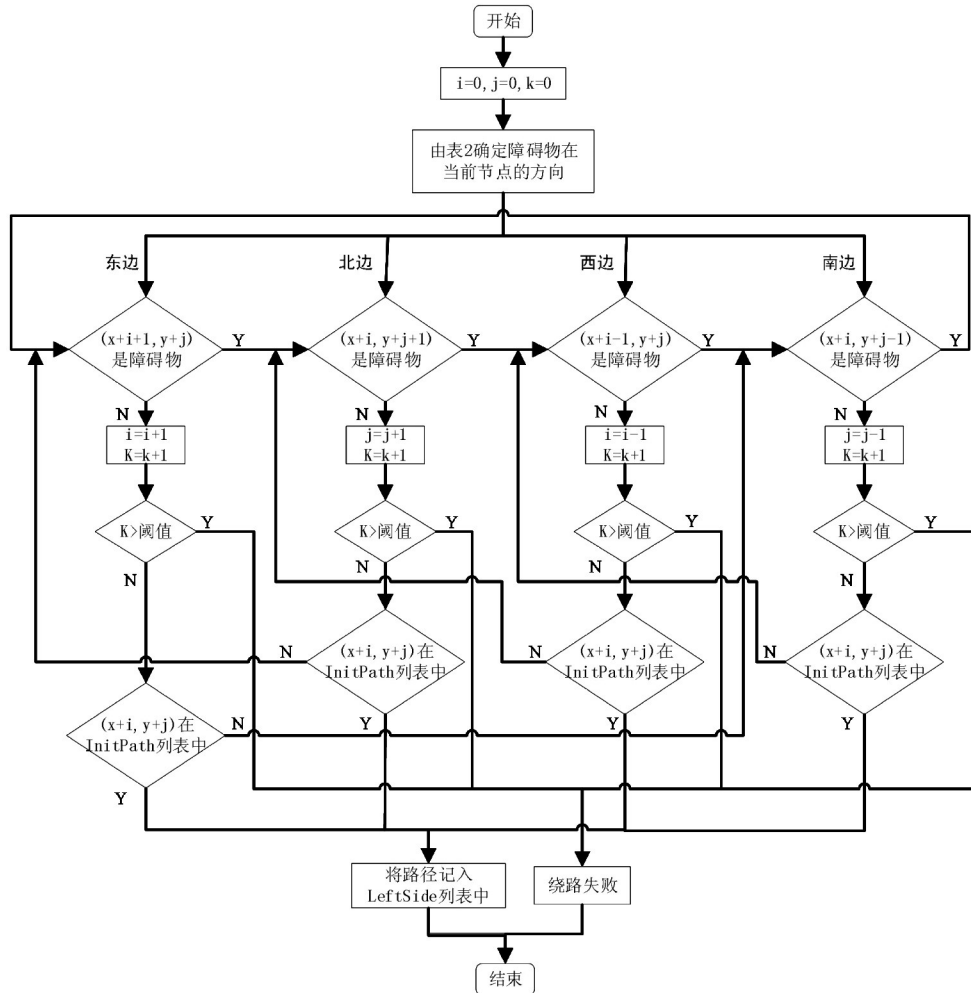


图2 摸墙算法流程图

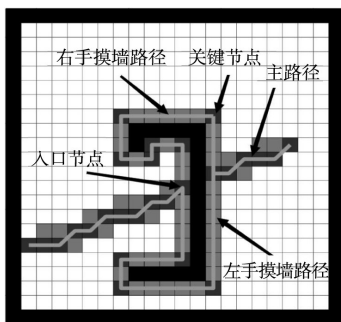


图3 摸墙算法路径规划图

### 2.3 路径冗余节点优化

针对传统A\*算法规划的路径存在较多转折点和冗余点的问题,本文结合Flody算法思想,去除路径中不必要的节点,缩短路径长度,减少转折点,提高移动机器人的运行效率。

Floyd算法<sup>[14]</sup>是一种寻找两点之间最短路径的算法。本文将结合Flody算法思想对路径进行优化,将第一个节点作为起始节点,依次将路径的下一个节点i作为目标节点,利用A\*算法对两点之间进行路径规划,判断路径中是否存在障碍物,若存在,则将i-1保留并视为新起始节点;若不存在,则接着路径的下一个节点;直至路径中的最后一个节点。

### 2.4 路径后处理

优化后的路径相邻节点之间若直接相连,会存在未搜索的区域,给移动机器人的安全带来威胁,需要进行后处理。

对路径相邻节点后处理思路为:利用A\*算法对相邻节点之间进行路径规划,对规划的路径节点进行连线,从第二个节点i开始,依次计算i-1~i与i~i+1组成线段的斜率,若斜率相同,判断此节点是否为倒数第二个节点,若是,删除所有中间节点,若不是,则接着计算。若斜率

不相同,则保留节点*i*-1,删除所有中间节点。图4为相邻节点处理前后的对比。

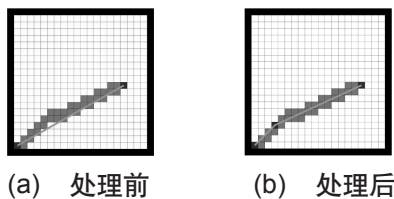


图4 相邻节点处理前后对比

由图4可知,对相邻节点处理后,规划路径均在搜索过的区域,保证了路径安全性。

### 2.5 改进A\*算法总体流程

利用摸墙算法改进A\*算法,其总体流程如下所示。

Step1: 获取环境信息并建立相应的栅格地图,执行Step2;

Step2: 在忽略障碍物的条件下,利用改进节点搜索条件的A\*算法规划一条从起始节点到目标节点的路径,并存入InitPath列表中,执行Step3;

Step3: 对InitPath列表中的节点依次判断是否为障碍物,若是则执行Step4,若不是,则将此节点存入Open列表中,并判断下一个节点。直至最后一个节点为止,则执行Step7;

Step4: 利用图2的摸墙算法流程图躲避障碍物,获得

左手摸墙算法路径LeftSide和右手摸墙算法路径RightSide。接着对左右路径进行关键转折点提取处理,执行Step5。若左右路径均绕路失败,则寻路失败,寻路结束;

Step5: 将Open加入左右路径中,并对路径进行冗余节点优化处理,执行Step6;

Step6: 计算优化后路径相邻节点距离之和,然后对左右路径距离进行比较,选择距离较近的路径替换Open列表,将InitPath列表中与Open列表最后一个点相同的节点作为下一个判断的节点,跳转执行Step3;

Step7: 对Open路径中相邻节点做后处理,并存入OverPath列表中,即获得最终路径。

### 3 仿真实验及分析

为测试本文改进A\*算法的各项性能,将利用仿真软件Matlab2016b进行模拟。计算机硬件配置:处理器为Intel(R) Core(TM)i5-8250U CPU@1.60 GHz运行内存为8 GB。对传统A\*算法与改进A\*算法进行仿真对比实验。仿真环境所采用的地图尺寸均为40像素×40像素(density independent pixels,dip)的二维栅格地图,进行3组实验,每组实验中障碍物设置难易程度不同,其中深灰色栅格代表路径节点,浅灰色栅格代表路径规划中遍历的栅格,黑色栅格代表障碍物存在的栅格,白色栅格代

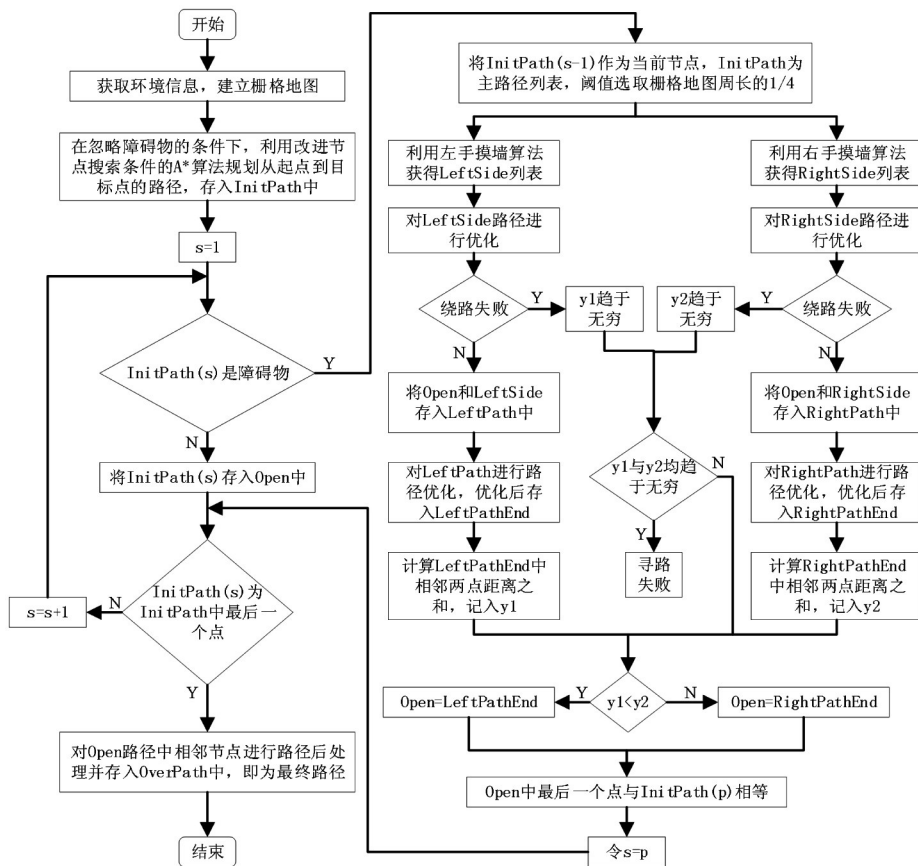


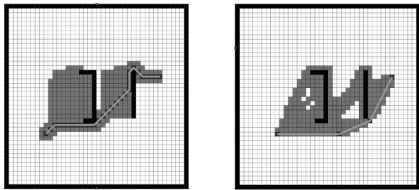
图5 改进算法总体流程图

表可通行的栅格。仿真结果如图6、图7和图8所示。图中起点与终点坐标如表3所示。

表3 不同模式下起点与终点坐标

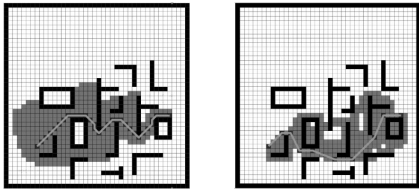
模式	起点坐标	终点坐标
简单模式	(9,12)	(35,25)
中等模式	(7,9)	(37,16)
困难模式	(9,9)	(35,25)

根据3组仿真实验图直观看出,传统A\*算法规划的路径与障碍物多次接触,且尖锐角和路径冗余节点较多,而改进A\*算法规划的路径,因对路径进行关键转折节点提取,使得路径安全性提高,与障碍物无接触。结合Flody算法思想对路径进行优化,使得冗余节点数量降低,路径更加平缓。



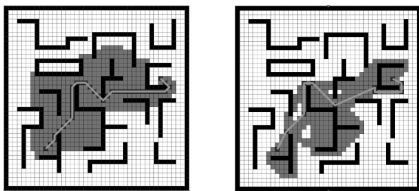
(a) 传统A\*算法 (b) 改进A\*算法

图6 实验1(简单模式)



(a) 传统A\*算法 (b) 改进A\*算法

图7 实验2(中等模式)



(a) 传统A\*算法 (b) 改进A\*算法

图8 实验3(困难模式)

表4 传统A\*算法与改进A\*算法实验结果对比

模式	算法	路径长度/dip	运算时间/s	路径节点/个	遍历节点/个
简单	传统A*算法	36.627 4	0.516	31	262
	改进A*算法	32.796 1	0.25	4	242
中等	传统A*算法	40.698 5	1.242	33	331
	改进A*算法	39.556 2	0.439	10	237
困难	传统A*算法	44.627 4	1.86	39	399
	改进A*算法	47.285	0.476	11	297

表4为传统A\*算法与改进A\*算法在不同模式下仿真实验数据。从表中可知,改进后的A\*算法在规划时长上相较于传统A\*算法平均减少了63.5%,在路径长度上平均减少了2.42%,在路径节点和遍历节点上,分别减少了

76.2%和20.5%。仿真实验结果表明,在不同的环境模式下,改进后的A\*算法相比于传统A\*算法,在搜索效率、冗余点数量 and 安全性等方面均有所提高,同时也兼顾了路径长度。

## 4 结束语

通过摸墙算法克服了传统A\*算法面对“U型”障碍物易陷入底部的局限性,提高了搜索效率,对扩展节点的搜索范围进行限定,减少搜索量,结合Flody算法思想,减少路径冗余节点和路径长度。通过仿真实验验证了改进后的A\*算法在规划时长、安全性、冗余点等方面均优于传统A\*算法,同时兼顾了路径长度。本文的改进A\*算法只能应用于全局静态路径规划中,没有考虑动态障碍物的避障,未来将继续改进。

## 参考文献:

- [1] 陆新华,张桂林.室内服务机器人导航方法研究[J].机器人,2003(1):80-87.
- [2] 王殿君.基于改进A\*算法的室内移动机器人路径规划[J].清华大学学报(自然科学版),2012,52(8):1085-1089.
- [3] 梁军,韩冬冬,盘朝奉,等.基于移动机器人的智能车库关键技术综述[J].机械工程学报,2022,58(3):1-20.
- [4] 杨韵,王成彦,巫凯旋,等.移动机器人全局路径规划算法综述[J].信息记录材料,2022,23(3):29-32.
- [5] 周恒旭,熊治国,肖晓波.面向非结构障碍物避障的A\*算法改进[J].自动化技术与应用,2025,44(2):9-12,84.
- [6] 陈文杰,平雪良.基于ROS的桌面机器人轨迹规划仿真研究[J].自动化技术与应用,2023,42(12):10-14.
- [7] 郑武略,郑扬亮,张鑫,等.复杂环境输电线路巡检机器人协同路径规划[J].自动化技术与应用,2025,44(2):13-16,25.
- [8] 王洋.改进量子粒子群算法的物流配送路径优化研究[J].自动化技术与应用,2021,40(7):14-17.
- [9] 金强,石永康,吕玉龙,等.基于改进遗传算法的植保无人机姿态控制[J].农机化研究,2024,46(5):1-6.
- [10] 孙淑光,孙涛.基于融合A\*算法的无人机路径规划研究[J].电子测量技术,2022,45(9):82-91.
- [11] TANG G,TANG C,CLARAMUNT C,et al.Geometric A-star algorithm:An improved a-star algorithm for AGV path planning in a port environment[J].IEEE Access,2021(9):59196-59210.
- [12] 申瑞,刘佳,王梦园,等.改进A\*算法的移动机器人路径规划设计[J].天津职业技术师范大学学报,2022,32(1):14-19.
- [13] 陈麒,阮艺冰.基于节点优化与清洗的改进A\*路径规划算法[J].电脑与信息技术,2022,30(2):5-8.
- [14] 吴海峰.最短路径算法—Dijkstra及Floyd算法[J].中国新通信,2019,21(2):32-33.

作者简介:武延涛(1997-),男,硕士,研究方向:人工智能与机器人。

通信作者:张震(1984-),男,博士,讲师,研究方向:机械振动,机械动力学与运动学。