

DOI: 10.13718/j.cnki.xdzk.2026.06.017

彭星宇, 何辉波, 李华英, 等. 基于改进 A\* 与 DFS 算法的割草机全覆盖路径规划 [J]. 西南大学学报(自然科学版), 2026, 48(6): 214-223.

## 基于改进 A\* 与 DFS 算法的 割草机全覆盖路径规划

彭星宇, 何辉波, 李华英, 黄云, 刘海洋

西南大学 工程技术学院, 重庆 400715

**摘要:** 针对智能割草机全覆盖路径规划拐点多、路径长的问题, 提出一种结合改进深度优先搜索 (Depth-First Search, DFS) 和 A\* 算法的路径规划方法。通过 MATLAB 软件对割草机作业环境进行栅格建模, 采用膨胀算法对障碍物区域进行处理, 为路径规划提供了安全边界。改进后的启发式成本函数结合了实际路径成本、启发式距离和覆盖修正项, 在仿真试验中有效提升了算法对复杂地形环境的适应性和路径覆盖效率。实物试验结果表明: 改进算法在保证路径最优性的同时, 实现了高效的全覆盖规划, 减少了冗余路径, 路径长度由 424 m 减少至 403 m, 拐点数量从 86 个减少至 48 个, 重复路径长度从 45 m 减少至 28 m。

**关键词:** 割草机; 功能设计; 遍历路径规划; 路径规划算法;  
深度优先搜索算法; A\* 算法

中图分类号: S231.9; TP242.6

文献标识码: A

开放科学(资源服务)标识码(OSID):



文章编号: 1673-9868(2026)06-0214-10

## Full Coverage Path Planning for Lawn Mowers Based on Improved A\* and DFS Algorithms

PENG Xingyu, HE Huibo, LI Huaying,  
HUANG Yun, LIU Haiyang

College of Engineering and Technology, Southwest University, Chongqing 400715, China

**Abstract:** This paper proposes a path planning method that combines an improved depth-first search (DFS) algorithm with the A\* algorithm to address the issues of excessive turning points and long path lengths in the full coverage path planning of intelligent lawn mowers. The operating environment of the lawn mower was modeled using a grid-based approach in MATLAB, and an inflation algorithm was applied to obstacle areas to establish safety boundaries for path planning. The improved heuristic cost function, which integrates actual path cost, heuristic distance, and a coverage correction term, effectively enhances the

收稿日期: 2025-04-21

基金项目: 教育部产学研合作协同育人项目(4322400254); 重庆市技术创新与应用发展专项面上项目(cstc2020jcsx-msxmX0192)。

作者简介: 彭星宇, 硕士研究生, 主要从事农业机械化及其自动化研究。

通信作者: 何辉波, 博士, 副教授, 硕士研究生导师。

adaptability of the algorithm and the efficiency of path coverage under complex terrains in the simulation test. Experimental results demonstrate that the improved algorithm achieves efficient full coverage planning while ensuring path optimality, significantly reducing redundant paths. Specifically, the total path length is reduced from 424 m to 403 m, the number of turning points is decreased from 86 to 48, and the length of redundant paths is reduced from 45 m to 28 m.

**Key words:** lawn mower; functional design; traverse path planning; path planning algorithm; depth-first search algorithm; A \* algorithm

随着现代化果园的建设和城市绿化的推进,除草工作变得尤为重要。传统的人工除草存在劳动效率低、生产成本低、工作量大等问题<sup>[1]</sup>,智能割草机可以解放人们的双手,大大提升劳动效率<sup>[2]</sup>。智能割草机是一个集外界环境感知、路径规划、障碍物检测与躲避以及多种传感器技术相结合的综合系统<sup>[3]</sup>。在割草机的应用中,研究的重点在于如何规划割草机的行走路径<sup>[4]</sup>,主要集中在全覆盖路径规划、路径跟踪控制和避障路径规划等方面<sup>[5]</sup>。路径规划的目的是最大限度地减少割草机的行走路径和行驶时间,同时使它们能够自主避开障碍物<sup>[6-7]</sup>,减少设备磨损,并降低制造成本<sup>[8]</sup>。常用的路径规划算法包括快速探索随机树(RRT)<sup>[9]</sup>、概率路线图法(PRM)<sup>[10]</sup>、Dijkstra 算法、A \* 算法<sup>[11]</sup>等。

近年来,国内外众多学者对割草机及其路径规划开展了大量研究。谢金燕等<sup>[12]</sup>提出了一种改进遗传算法,为每台割草机分配并优化作业路径。宋黎明等<sup>[13]</sup>以 STM32 为主控制器,结合无线传感技术,提高了草坪修剪机器人的修剪效率。谢逢博等<sup>[14]</sup>在对果园割草机器人进行路径规划时,控制方法确定为横向偏差修正算法,有效降低了机器人行进的误差。Höffmann 等<sup>[15]</sup>提出了一种智能割草机自动路径规划和高精度定位的方法。张旭东等<sup>[16]</sup>为割草机增加了避障、报警和自适应割草等功能,提高了割草效率。Martelli 等<sup>[17]</sup>提出了农业四轮差速驱动车辆开发路径规划和路径跟踪算法,引入 LD 的动态值提高了算法效率并减少了计算时间。周敬东等<sup>[18]</sup>提出了一种基于改进蚁群算法的移动机器人路径规划方法,通过改变初始化信息素浓度分配、改变启发式函数等方法对蚁群算法进行优化,提高了算法的鲁棒性和寻优能力。上述研究在割草机路径规划的效率和精度上取得了较好的进展,但在复杂地形的适应性、动态环境的实时路径规划以及算法的鲁棒性和计算效率方面仍有改进空间。

本文提出一种基于改进深度优先搜索(Depth-First Search, DFS)算法和 A \* 算法的全覆盖路径规划方法,结合 2 种算法的优势,改进传统路径规划技术在复杂环境中的适应性和效率。通过引入覆盖优先级导向机制和动态深度控制策略,优化 DFS 算法的局部搜索效率;采用 A \* 算法通过改进启发式成本函数,动态调整路径选择,以期提升保障能力和覆盖效率。此外,本文还结合膨胀算法对障碍物区域进行处理,为路径规划提供安全边界,降低冗余路径,提高割草机作业效率。

## 1 环境栅格地图的构建与处理

割草机在进行路径规划之前,需要对其作业环境建立地图模型,本文使用 MATLAB R2022b 软件中的栅格法对割草机的工作环境进行建模,将割草机的工作环境全局信息分解成一系列具有二值信息的等大小正方形网格单元<sup>[19]</sup>。环境地图由 30×30 个单元栅格构成,其特征是具有分布不均匀的障碍物,赋予栅格不同的值来表示栅格不同的状态:0 表示黑色栅格,代表环境地图中的障碍物区域;1 表示白色栅格,代表环境地图中的自由区域。

由于环境中存在不规则的障碍物,所建立的栅格地图中的一些障碍物不能占满一个完整的单元栅格,且此时建立的栅格地图未考虑割草机的车身尺寸,不仅容易使规划后的路径与障碍物发生碰撞,也可能陷入盲区或死角,因此需要对不规则的障碍物进行膨胀处理。在栅格地图路径规划中,膨胀算法可以帮助处理栅格图中障碍物的边界,使障碍物的范围适当扩大,从而为路径规划留出一定的安全边界,避免割草机靠近障碍物边缘并产生碰撞风险。

### 1.1 膨胀操作的基础公式

膨胀操作的定义是使用一个结构元素  $B$  扩展图像中的目标区域  $A$ 。在二值图像中, 该操作将所有与结构元素重叠的像素设为目标像素, 从而使目标区域扩大, 其公式如下:

$$A \oplus B = \{Z \mid (B_Z \cap A) \neq \emptyset\} \quad (1)$$

式中:  $A$  为原始栅格图中的障碍物区域;  $B$  为结构元素, 是一个形状规则的  $3 \times 3$  方形小矩阵;  $B_Z$  表示将结构元素  $B$  平移到位置  $Z$ 。  $A \oplus B$  的结果是所有能使  $B$  与  $A$  至少有一个像素重叠的位置的集合。这一膨胀操作的功能实现可以参考 ROS Wiki 中的 `costmap_2d` 功能包<sup>[20]</sup>。

### 1.2 路径规划中的安全区域膨胀

在路径规划中, 为确保安全性, 通常需要在障碍物区域周围留出一个安全边界, 即对障碍物区域执行膨胀。假设  $A$  表示栅格图中的障碍物区域, 定义安全距离为  $d$ , 结构元素可以选择具有半径为  $d$  的圆形结构元素  $B_d$ , 其公式如下:

$$A_{\text{expanded}} = A \oplus B_d \quad (2)$$

式中:  $B_d$  是半径为  $d$  的圆形结构元素。膨胀后的障碍物区域  $A_{\text{expanded}}$  包含了原障碍物区域及其周围的安全距离区域。

### 1.3 多步膨胀公式

为了实现更大范围的膨胀, 可以多次应用膨胀操作, 以获得所需的扩展效果。对于重复执行膨胀操作  $n$  次, 其膨胀结果可以表示为:

$$A_n = A \oplus B_1 \oplus B_2 \oplus \dots \oplus B_n \quad (3)$$

栅格图上膨胀算法的实现步骤如下:

**步骤 1** 选择结构元素。根据障碍物区域的形状特征和所需的安全距离, 选择大小和形状适当的结构元素  $B$ 。

**步骤 2** 应用膨胀操作。对栅格地图中的障碍物区域  $A$  进行膨胀操作, 将  $B$  在栅格图上进行逐像素移动, 检查其是否与障碍物区域有重叠。

**步骤 3** 生成膨胀后的地图。将膨胀结果合并到栅格图中, 得到包含安全区域的膨胀障碍物区域  $A_{\text{expanded}}$ 。

所建栅格地图与经过膨胀处理后的栅格地图进行对比, 如图 1 所示。

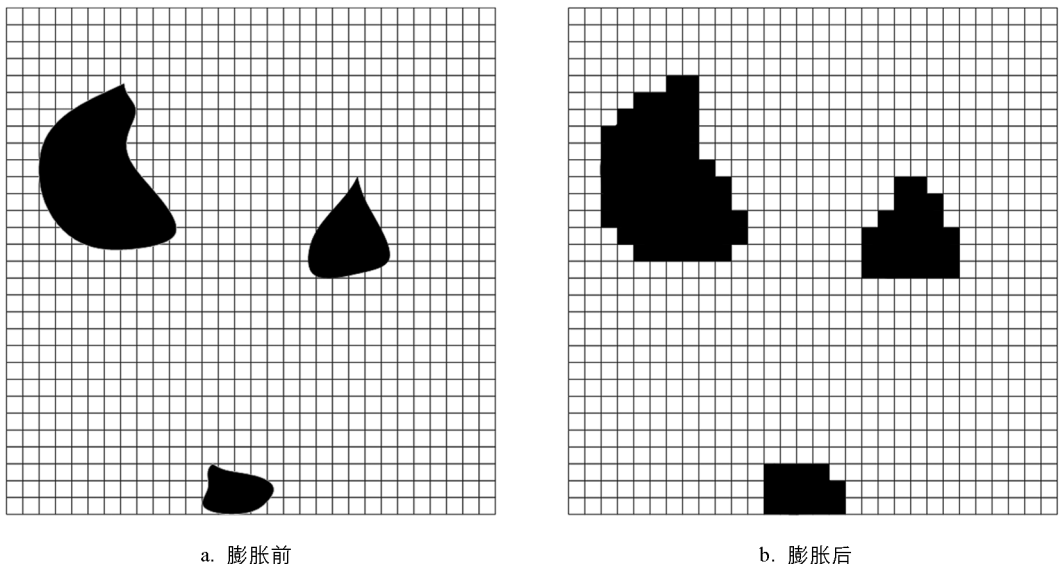


图 1 膨胀处理前后栅格地图对比

在实际应用中,膨胀算法结合路径规划有助于在智能割草机导航时避开障碍物,并创建更安全的路径。通过设置合适的膨胀半径和结构元素,能实现灵活的障碍物扩展效果,从而提高路径规划的可靠性。

## 2 基于改进 DFS 和 A \* 算法的覆盖路径规划

覆盖路径规划(Coverage Path Planning, CPP)是移动机器人执行清扫、巡逻等任务的核心技术,其目标是高效覆盖目标区域,避免冗余路径和遗漏区域。经典的 DFS 和 A \* 算法因其良好的搜索性能和路径规划能力被广泛应用,但各有局限。DFS 算法虽具有低复杂度和快速覆盖的优势,但在复杂或障碍物密集环境中易产生冗余路径,效率较低;A \* 算法通过综合实际成本与启发式成本,擅长全局路径优化,但对区域覆盖的关注不足。

本文结合两者优点,提出改进算法。对 DFS 算法引入覆盖优先级导向机制,使其优先访问未覆盖区域,减少重复覆盖;对 A \* 算法改进其代价函数,引入障碍物距离和未覆盖区域优先级,提高避障能力和覆盖效率。此外,通过分阶段规划方法,在初始阶段用改进 DFS 实现基础覆盖,在动态调整阶段结合改进 A \* 优化路径,兼顾全局最优性与局部最优性。

### 2.1 改进路径规划算法设计

为了提高路径规划的效率和适应性,本文对传统 A \* 算法和 DFS 算法进行了改进,分别提出了改进的启发式成本函数和局部搜索方法,以下将详细阐述改进方法及其作用。

#### 2.1.1 改进的 A \* 启发式成本函数

改进后的启发式成本函数基本形式如下:

$$f'(n) = g(n) + \alpha \cdot h(n) + \beta \cdot c(n) \quad (4)$$

式中:  $g(n)$  为从起点到节点  $n$  的实际成本,与传统 A \* 算法相同;  $h(n)$  为节点  $n$  到目标节点的启发式距离,可以使用曼哈顿距离;  $c(n)$  为覆盖成本修正项,用于衡量当前节点  $n$  的覆盖需求优先级;  $\alpha$  和  $\beta$  为权重系数,分别控制距离启发项和覆盖修正项对总成本的影响。

##### 2.1.1.1 曼哈顿距离

启发式距离保持与原始 A \* 算法一致,使用曼哈顿距离表示当前位置与目标位置之间的距离。

##### 2.1.1.2 覆盖成本修正项 $c(n)$

覆盖成本修正项提升未覆盖区域优先级,并结合障碍物分布动态调整路径选择。定义覆盖成本修正项  $c(n)$  为:

$$c(n) = \frac{U_n}{S_n} + \gamma \cdot d_{\text{obstacle}}(n) \quad (5)$$

式中:  $U_n$  为节点  $n$  周围未覆盖的邻居节点数;  $S_n$  为节点  $n$  的总邻居数(最大为 8);  $\gamma$  为障碍物权重系数;  $d_{\text{obstacle}}(n)$  为节点  $n$  到最近障碍物的距离,定义为曼哈顿距离,即:

$$d_{\text{obstacle}}(n) = \min_{O \in O} (|x_n - x_0| + |y_n - y_0|) \quad (6)$$

式中:  $O$  为障碍物节点的集合;  $(x_n, y_n)$  为节点  $n$  的坐标;  $(x_0, y_0)$  为障碍物位置的坐标。

##### 2.1.1.3 改进后的启发式成本函数

将上述改进部分结合,得到改进后的启发式成本函数为:

$$f'(n) = g(n) + \alpha \cdot h(n) + \beta \cdot \left[ \frac{U_n}{S_n} + \gamma \cdot d_{\text{obstacle}}(n) \right] \quad (7)$$

改进后的启发式成本函数  $f'(n)$  通过结合未覆盖区域优先级和障碍物距离控制,使得算法在进行路径规划时,既能够快速找到目标位置,又能有效覆盖未访问区域,同时还能动态避开障碍物,提升了整体路径规划的安全性和效率。

### 2.1.1.4 算法验证

为了验证上述改进 A\* 算法的有效性, 在试验环境为 Windows 10 系统、处理器为 Intel i5-7200U、运行内存为 16 GB 的硬件平台上采用 MATLAB R2022b 仿真软件进行试验。将改进的 A\* 算法与传统 A\* 算法在复杂环境下进行了对比, 仿真结果如图 2 所示。

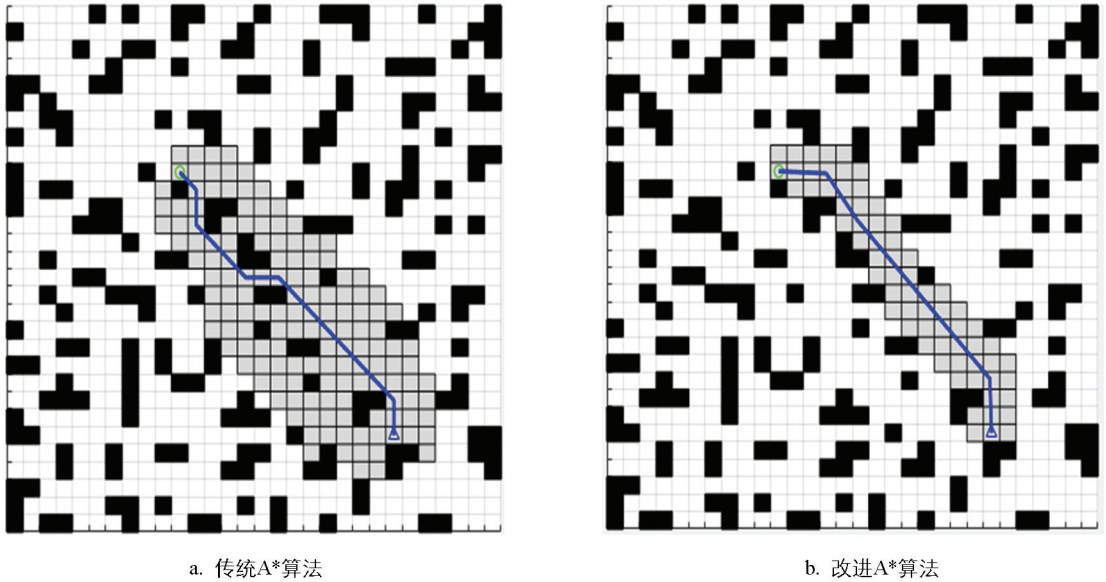


图 2 复杂环境下仿真结果对比

由仿真结果可知, 本文提出的改进 A\* 算法相较于传统 A\* 算法在复杂环境下的搜索过程更具目的性和导向性, 生成的路径不仅更短, 同时路径平滑度显著提升。在本次试验中, 该算法表现出更高的效率和适应性。

为进一步量化性能差异, 本文对改进 A\* 算法与传统 A\* 算法在路径长度、拐点数量、遍历节点数量方面进行统计, 结果如图 3 所示。统计数据显示, 改进 A\* 算法在遍历节点数量上减少了 53.74%, 拐点数量减少了 40%, 路径长度降低了 12.55%, 充分证明了改进算法在提升路径规划效率和质量方面的有效性和优势。

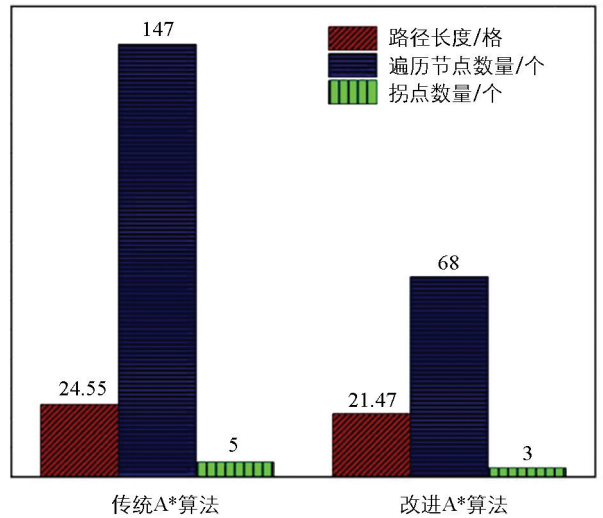


图 3 传统 A\* 算法和改进 A\* 算法的性能对比

### 2.1.2 改进的 DFS 局部搜索方法

DFS 算法用来寻找从出发点至目标点的可行路径, 通过依次访问与上一节点相邻的可行节点, 访问完图中所有与出发点有路径相通的节点<sup>[21]</sup>。在传统 DFS 算法基础上, 设定一个改进的 DFS 算法, 定义路径优先级评分  $P_{score}(n)$  与动态深度控制函数  $d_{limit}(n)$ , 以控制 DFS 算法的搜索方向和深度。

#### 2.1.2.1 路径优先级评分

每个节点的路径优先级评分  $P_{score}(n)$  定义如下:

$$P_{score}(n) = \omega_1 \cdot \left( \frac{U_n}{S_n} \right) + \omega_2 \cdot \left[ \frac{1}{d_{center}(n) + 1} \right] + \omega_3 \cdot \left[ \frac{1}{1 + \sum_{j \in N(n)} P(j)} \right] \quad (8)$$

式中:  $U_n$  为节点  $n$  周围未覆盖的邻居节点数;  $S_n$  为节点  $n$  的总邻居数;  $d_{\text{center}}(n)$  为节点  $n$  到局部搜索中心(起始节点)的曼哈顿距离;  $N(n)$  为节点  $n$  的邻居集合;  $P(j)$  为指示函数, 如果节点  $j$  是障碍物或已覆盖, 则  $P(j)=1$ , 否则为 0;  $\omega_1$ 、 $\omega_2$ 、 $\omega_3$  分别为控制每个评分项的权重系数;  $\frac{U_n}{S_n}$  为未覆盖邻居比率, 鼓励割草机优先访问未覆盖邻居较多的节点;  $\frac{1}{d_{\text{center}}(n)+1}$  为路径优先级评分的节点到中心节点的距离, 鼓励在局部区域内保持一定的覆盖密度;  $\frac{1}{1+\sum_{j \in N(n)} P(j)}$  为障碍物和已覆盖节点的密度。

### 2.1.2.2 动态深度控制函数

动态深度控制函数  $d_{\text{limit}}(n)$  根据当前节点  $n$  周围的未覆盖密度来动态调整 DFS 的最大搜索深度, 以避免进入过深的递归, 定义如下:

$$d_{\text{limit}}(n) = d_{\text{base}} + \alpha \cdot \left( \frac{U_n}{S_n} \right) \quad (9)$$

式中:  $d_{\text{limit}}(n)$  为节点  $n$  的动态深度控制;  $d_{\text{base}}$  为基本搜索深度, 可以根据任务设定一个初始值;  $\alpha$  为控制未覆盖密度对深度的影响。

当前节点  $n$  的深度  $d(n) > d_{\text{limit}}(n)$  时, 表示当前区域的未覆盖邻居节点密度低, 应限制搜索深度, 防止搜索陷入不必要的深层。当节点  $n$  周围的未覆盖密度较高时,  $d_{\text{limit}}(n)$  增加, 允许 DFS 深入搜索该区域; 当未覆盖密度较低时,  $d_{\text{limit}}(n)$  保持在较小值, 减少不必要的深度搜索。

### 2.1.2.3 输出局部路径的触发条件

输出局部路径的触发条件包括:

① 当前节点  $n$  的  $P_{\text{score}}(n)$  达到最大值(即邻域中最优)。

② 在某次 DFS 扩展过程中, 遍历到的所有邻居节点评分均低于某一设定阈值  $\theta$ , 局部搜索收益下降。其中,  $\theta$  为评分阈值, 可根据任务密集度经验设定(如 0.3~0.5), 本文仿真中取值为 0.4。

③ 搜索深度已达到  $d_{\text{limit}}(n)$ , 无法继续向前推进, 即当前节点  $n$  的深度  $d(n) > d_{\text{limit}}(n)$ 。

当计算路径优先级评分时, 若为以上①和②的情况, 则触发局部路径输出, 否则触发动态深度控制; 当动态深度控制中达到情况③时, 也触发局部路径输出。

### 2.1.2.4 搜索路径优化方法

针对 DFS 算法在局部搜索中的应用, 本文对其进行优化, 以提升局部覆盖的效率和路径规划的合理性。通常情况下, DFS 容易导致重复访问或不必要的深度回溯, 因此可以引入以下改进方法来优化其搜索路径。

① 路径优先级评分: 在每个节点上为不同方向赋予优先级评分, 以引导割草机优先访问未覆盖、低成本或重要的节点。

② 动态深度控制: 引入动态深度控制, 根据当前区域的未覆盖密度来调整 DFS 的搜索深度, 以避免进入不必要的深度递归。

③ 局部启发式选择: 在每次深度搜索中, 使用启发式信息(如未覆盖邻居数、距离中心区域的距离等)选择最佳路径。

改进前后 DFS 算法的路径规划如图 4 所示, 可以看出, 改进前 DFS 在规划过程中的拐点数量为 4 个, 路径平滑度较低, 而改进后的 DFS 通过优先访问未覆盖区域、合理控制搜索深度, 有效减少了路径中的拐点数量, 拐点仅为 2 个, 提升了局部覆盖的效率和路径的整体平滑性。改进后的算法流程图如图 5 所示。

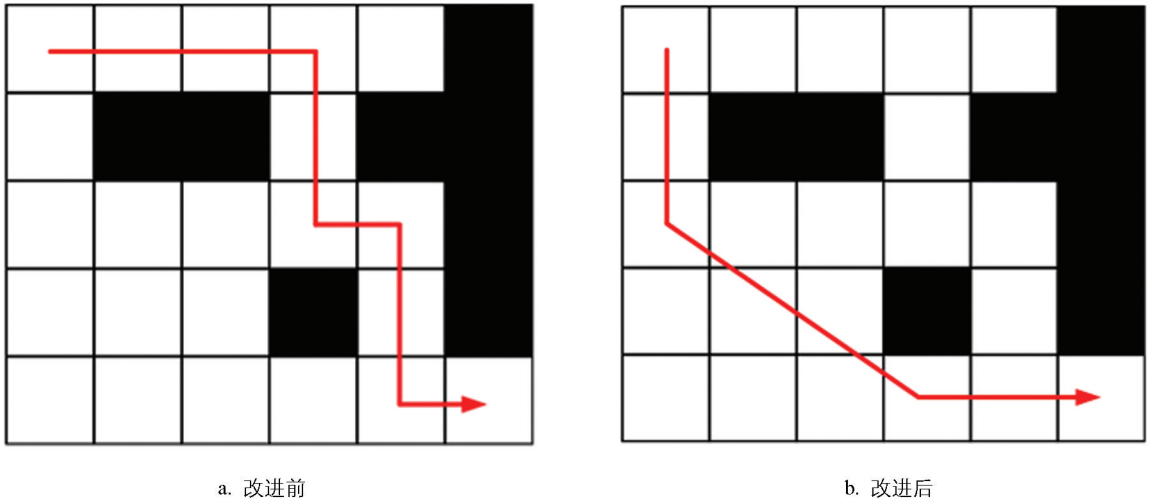


图 4 改进前后 DFS 算法的路径规划图

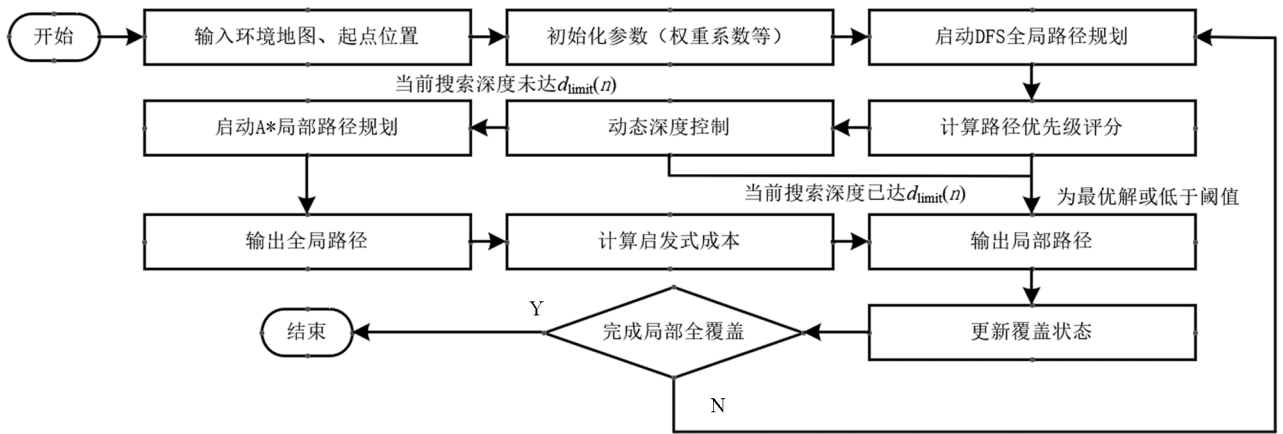


图 5 改进后的算法流程图

## 2.2 算法仿真

为了验证本文所提出的改进 DFS 算法和 A\* 算法的可靠性, 对改进前和改进后的算法分别进行了仿真试验。试验在不同障碍物数量的栅格地图中进行, 以测试算法在不同复杂环境下的性能表现。在试验环境为 Windows 10 系统、处理器为 Intel i5-7200U、运行内存为 16 GB 的硬件平台上采用 MATLAB R2022b 仿真软件进行试验。

本文设计了 3 组对照试验, 分别在 3 张地图上设置 1、3、5 个障碍物, 以此来模拟不同复杂度的环境, 其中 5 个障碍物的对照试验图如图 6 所示。在地图中, 粉色五星表示路径的起始栅格, 红色直线表示割草机覆盖的路径, 绿色栅格表示重复遍历的区域, 灰色栅格表示障碍物。

仿真结果显示, 在障碍物数量相同且起点一致的地图中, 改进后的算法相较于改进前在转向次数、重复遍历的栅格数量以及路径长度等关键指标上均展现出显著优化效果。改进后的算法路径更加高效, 避免了不必要的重复和多余的转向操作。

改进前后算法在不同障碍物数量下的性能对比如表 1 所示, 随着障碍物数量的增加, 地图复杂度提升, 割草机的转向次数、重复遍历栅格数量和路径长度均呈上升趋势。然而, 无论在何种复杂程度下, 改进后的算法能够有效减少转向次数和重复遍历区域, 同时缩短路径长度, 尤其在高复杂度地图中优势更加明显, 表明了算法改进的有效性。

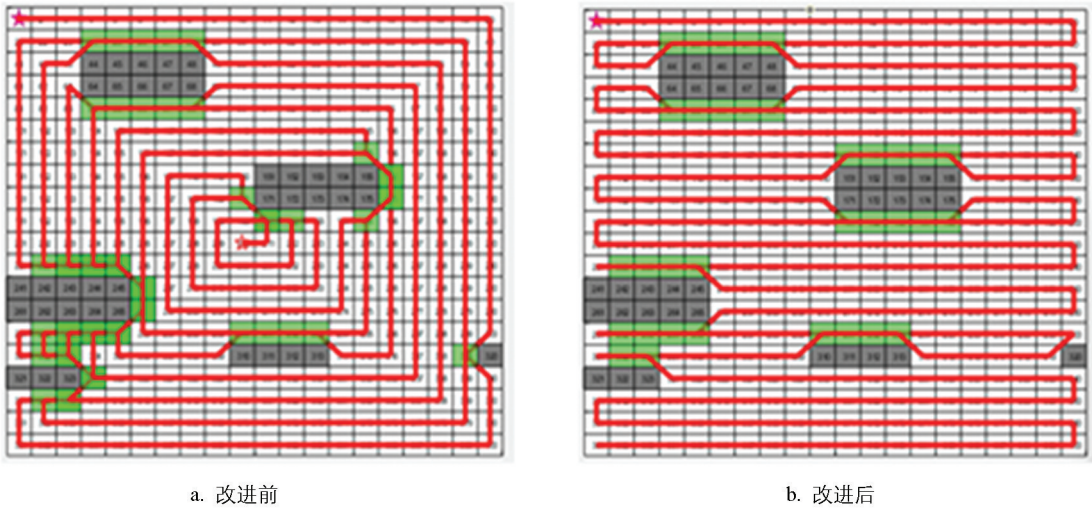


图 6 算法改进前后的路径覆盖结果(5个障碍物)

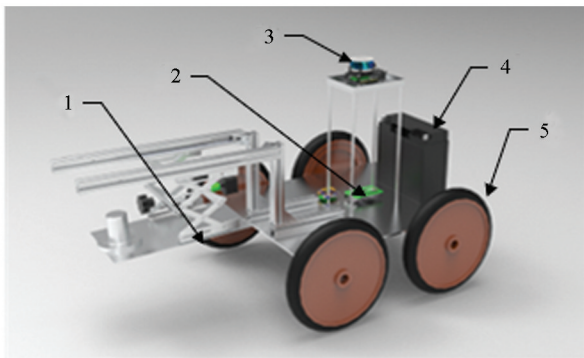
表 1 改进前后算法性能对比

序号	障碍物数量/个	算法	路径长度/m	拐点数量/个	重复遍历栅格数量/个
1	1	改进前算法	399	42	5
		改进后算法	394	38	0
2	3	改进前算法	412	80	27
		改进后算法	406	67	23
3	5	改进前算法	436	104	63
		改进后算法	400	61	34

### 3 智能割草机总体方案设计及试验验证

#### 3.1 总体方案

本文设计的智能割草机主要由割草模块、控制模块、传感器模块、供电模块以及驱动模块组成,如图 7 所示。该智能割草机移动平台采用四轮驱动模式,四轮驱动有更好的牵引力和通过性,并且减少轮胎打滑的可能性,能够使割草机在较为复杂的环境下进行割草作业,例如果园以及地形崎岖的草地。



a. 割草机整机建模



b. 割草机整机实物

1. 割草模块; 2. 控制模块; 3. 传感器模块; 4. 供电模块; 5. 驱动模块。

图 7 割草机整机图

### 3.2 试验验证

为验证本文提出的改进路径规划算法在实际应用中的可靠性与有效性, 将其应用于四驱智能割草机。割草机装配了激光雷达、惯性测量单元、编码器等传感器, 通过电脑上的虚拟机进行远程连接, 发送指令到 ROS 主控负责路径规划, STM32 控制板接收路径信息并控制电机运动完成任务。试验评价指标包括路径长度、路径拐点数量以及重复路径长度。

试验在草坪上进行, 如图 8 所示, 割草机通过激光雷达构建二维地图, 规划路径并完成覆盖作业。ROS 可视化界面显示生成的全覆盖导航路径, 系统自动记录路径长度、拐点数量和重复路径长度等指标, 用于对比分析不同算法的性能。

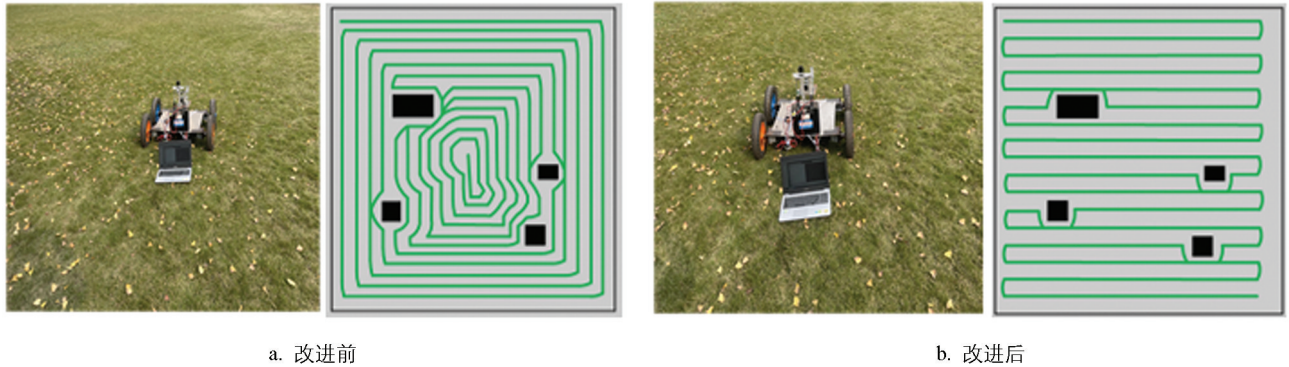


图 8 全覆盖路径规划结果

试验结果如表 2 所示, 改进算法在路径长度、拐点数量和重复路径长度 3 个方面均表现出显著优势。具体而言, 改进前算法的路径长度为 424 m, 而改进后算法将其缩短至 403 m, 减少了 4.95%, 表明改进算法能够生成更加紧凑的覆盖路径。此外, 拐点数量由 86 个减少至 48 个, 降幅达 44.19%, 显著提升了路径的平滑性, 有助于减少割草机运行中的转向时间和能耗。同时, 重复路径长度由 45 m 减少至 28 m, 降低了 37.78%, 充分体现了改进算法在避免重复覆盖方面的效率。

表 2 试验评价结果与对比分析

算法	路径长度/m	拐点数量/个	重复路径长度/m
改进前算法	424	86	45
改进后算法	403	48	28

## 4 结论

为验证智能割草机在复杂环境下的全覆盖路径规划性能, 本文对改进 DFS 算法和 A\* 算法进行了仿真试验与实物验证研究, 得出以下结论:

1) 通过仿真试验对改进前后算法在不同障碍物数量下(1、3、5 个)的性能进行对比研究, 结果表明改进算法在路径长度、拐点数量和重复遍历栅格数量方面均表现出显著优势。在障碍物数量为 5 个的场景中, 改进算法将路径长度从 436 m 减少至 400 m, 拐点数量从 104 个减少至 61 个, 重复遍历栅格数量从 63 个减少至 34 个, 验证了改进算法在复杂环境下的优越性。

2) 在实际草坪场景中对改进算法进行试验验证, 改进算法显著优化了路径长度、拐点数量和重复路径长度等指标。实物试验中, 路径长度由 424 m 减少至 403 m, 拐点数量从 86 个减少至 48 个, 重复路径长度从 45 m 减少至 28 m。改进算法生成的路径在 ROS 可视化界面中更加平滑高效, 实物运行过程表现出良好的覆盖率和作业效率。

3) 实物试验与仿真试验在路径规划指标的变化趋势上具有较高的一致性, 证明了改进算法具有一定的可靠性。同时, 验证了本文提出的路径规划算法能够有效减少转向次数和重复路径长度, 从而提升割草机的覆盖效率, 为复杂环境下的路径规划提供了可靠的技术支持。

## 参考文献:

- [1] LI Y, GUO Z Q, SHUANG F, et al. Key Technologies of Machine Vision for Weeding Robots: A Review and Benchmark [J]. *Computers and Electronics in Agriculture*, 2022, 196: 106880.
- [2] 陈镜宇, 郭志军, 尹亚昆. 基于混合算法的智能割草机全遍历路径规划及其系统设计 [J]. *计算机科学*, 2021, 48(S1): 633-637.
- [3] 张得明, 高龙琴, 陈金豹, 等. 一种基于 ARM 的割草机器人边界图像处理方法设计研究 [J]. *机械制造与自动化*, 2017, 46(4): 166-169.
- [4] ZHANG M K, LI X G, WANG L, et al. A Path Planning System for Orchard Mower Based on Improved A\* Algorithm [J]. *Agronomy*, 2024, 14(2): 391.
- [5] 谢金燕, 刘丽星, 杨欣, 等. 改进粒子群优化算法的果园割草机作业路径规划 [J]. *中国农业大学学报*, 2023, 28(11): 182-191.
- [6] HICHRI B, GALLALA A, GIOVANNINI F, et al. Mobile Robots Path Planning and Mobile Multirobots Control: A Review [J]. *Robotica*, 2022, 40(12): 4257-4270.
- [7] AGARWAL D, BHARTI P S. Implementing Modified Swarm Intelligence Algorithm Based on Slime Moulds for Path Planning and Obstacle Avoidance Problem in Mobile Robots [J]. *Applied Soft Computing*, 2021, 107: 107372.
- [8] XIE B B, JIN Y C, FAHEEM M, et al. Research Progress of Autonomous Navigation Technology for Multi-Agricultural Scenes [J]. *Computers and Electronics in Agriculture*, 2023, 211: 107963.
- [9] ABDALLAOUI S, AGLZIM E H, CHAIBET A, et al. Thorough Review Analysis of Safe Control of Autonomous Vehicles: Path Planning and Navigation Techniques [J]. *Energies*, 2022, 15(4): 1358.
- [10] CHEN G, LUO N, LIU D, et al. Path Planning for Manipulators Based on an Improved Probabilistic Roadmap Method [J]. *Robotics and Computer-Integrated Manufacturing*, 2021, 72: 102196.
- [11] 王云峰, 曹小华, 郭兴. 基于改进 A\* 算法和系统短期状态预测的仓储 AGV 路径规划方法 [J]. *计算机集成制造系统*, 2023, 29(11): 3897-3908.
- [12] 谢金燕, 刘丽星, 杨欣, 等. 苹果园内无人割草机多机协同作业路径优化算法 [J]. *华南农业大学学报*, 2024, 45(4): 578-587.
- [13] 宋黎明, 马福贵. 基于 STM32 的智能草坪修剪机器人设计 [J]. *机电工程技术*, 2020, 49(10): 141-143.
- [14] 谢逢博, 李君, 李振伟, 等. 果园割草机器人路径规划与轨迹跟踪控制研究 [J]. *中国热带农业*, 2021(1): 17-24, 16.
- [15] HÖFFMANN M, CLEMENS J, STRONZEK-PFEIFER D, et al. Coverage Path Planning and Precise Localization for Autonomous Lawn Mowers [C] //2022 Sixth IEEE International Conference on Robotic Computing (IRC), December 5-7, 2022, Italy. IEEE, 2023: 238-242.
- [16] 张旭东, 罗安江, 周安仲, 等. 多功能智能割草机设计 [J]. *拖拉机与农用运输车*, 2023, 50(4): 58-62.
- [17] MARTELLI S, MOCERA F, SOMÀ A. Autonomous Driving Strategy for a Specialized Four-Wheel Differential-Drive Agricultural Rover [J]. *AgriEngineering*, 2024, 6(3): 1937-1958.
- [18] 周敬东, 高伟周, 杨文广, 等. 基于改进蚁群算法的移动机器人路径规划 [J]. *科学技术与工程*, 2022, 22(28): 12484-12490.
- [19] 王新彦, 盛冠杰, 张凯, 等. 基于改进 A\* 算法和 DFS 算法的割草机器人遍历路径规划 [J]. *中国农机化学报*, 2023, 44(2): 142-147.
- [20] YUAN R P, ZHANG F H, QU J D, et al. A Novel Obstacle Avoidance Method Based on Multi-Information Inflation Map [J]. *Industrial Robot: The International Journal of Robotics Research and Application*, 2019, 47(2): 253-265.
- [21] 宫金良, 王伟, 张彦斐, 等. 基于农田环境的农业机器人协同作业策略 [J]. *农业工程学报*, 2021, 37(2): 11-19.