

# 控制与决策

Control and Decision

## 基于常春藤高斯引导算法的无人车路径规划

沈波, 杜鹏桢

引用本文:

沈波, 杜鹏桢. 基于常春藤高斯引导算法的无人车路径规划[J]. *控制与决策*, 2026, 41(3): 822-834.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2025.0292>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 基于视觉引导多AGV系统的改进A\*路径规划算法

Improved A\* path planning algorithm for vision-guided multi-AGV system

*控制与决策*. 2021, 36(8): 1881-1890 <https://doi.org/10.13195/j.kzyjc.2019.1670>

#### 基于改进蚁群算法的水面无人艇路径规划

Path planning for unmanned surface vehicle based on improved ant colony algorithm

*控制与决策*. 2021, 36(4): 847-856 <https://doi.org/10.13195/j.kzyjc.2019.0839>

#### 求解约束优化问题的改进果蝇优化算法及其工程应用

Improved fruit fly optimization algorithm for solving constrained optimization problems and engineering applications

*控制与决策*. 2021, 36(2): 314-324 <https://doi.org/10.13195/j.kzyjc.2019.0557>

#### 基于 $\text{pm}3\sigma$ 正态概率区间分族遗传蚁群算法的机器人路径规划

Path planning of mobile robot based on  $\text{pm}3\sigma$  normal probability interval population division using genetic ant-colony algorithm

*控制与决策*. 2021, 36(12): 2861-2870 <https://doi.org/10.13195/j.kzyjc.2020.0745>

#### 基于平衡鲸鱼优化算法的无人车路径规划

Path planning of unmanned ground vehicle based on balanced whale optimization algorithm

*控制与决策*. 2021, 36(11): 2647-2655 <https://doi.org/10.13195/j.kzyjc.2020.0416>

# 基于常春藤高斯引导算法的无人车路径规划

沈波, 杜鹏桢<sup>†</sup>

(南京理工大学 计算机科学与工程学院, 南京 210094)

**摘要:** 针对常春藤算法这类固定维连续域优化方法难以直接处理具有离散约束和连贯性要求的无人车路径规划问题, 提出一种基于改进常春藤优化的高斯引导算法. 首先设计高斯引导机制, 提供一种可行的桥接式应用建模方式, 通过迭代优化引导点参数引导路径生成, 并利用路径离散特性大幅降低计算量; 然后对常春藤算法低值偏好的操作进行修正改进, 使路径搜索在全局范围更均衡, 并提出螺旋衰减策略更新生长速率, 使算法更适应路径离散环境; 最后设计一种局部陷阱逃脱机制, 利用动态时间规整算法检测局部陷阱, 重新分散种群以摆脱局部最优, 以增强全局搜索能力与算法稳定性. 在不同规模的仿真环境中进行实验, 结果表明所提出算法具有较好的求解质量和收敛效率, 同时展现出强大的稳定性与平滑性, 更符合无人车的行驶需求.

**关键词:** 无人引导车; 路径规划; 常春藤算法; 高斯引导机制; 时序相似度量; 局部最优逃脱

中图分类号: TP242 文献标志码: A

DOI: 10.13195/j.kzyjc.2025.0292

引用格式: 沈波, 杜鹏桢. 基于常春藤高斯引导算法的无人车路径规划 [J]. 控制与决策, 2026, 41(3): 822-834.

## Path planning for automated guided vehicles based on ivy Gaussian-guided algorithm

SHEN Bo, DU Peng-zhen<sup>†</sup>

(School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

**Abstract:** This paper addresses the challenge of applying fixed-dimension continuous-domain optimization methods such as the ivy algorithm to autonomous vehicle path planning with discrete constraints and continuity requirements. To this end, we propose a Gaussian-guided algorithm based on an improved ivy optimization method. A Gaussian guidance mechanism is introduced as a bridging modeling approach that iteratively optimizes guide point parameters to steer path generation while leveraging the discrete nature of the problem to substantially reduce computational cost. In addition, the original bias of the ivy algorithm toward low-value solutions is corrected to achieve a more balanced global search, a spiral decay strategy is employed to update the growth rate smoothly for better adaptation to discrete grid environments, and a dynamic time warping (DTW)-based trap escape mechanism is designed to detect stagnation and restore population diversity. Simulation experiments across environments of different scales demonstrate that the proposed method achieves high-quality solutions with improved convergence efficiency, stability, and path smoothness, making it well suited for autonomous navigation tasks.

**Keywords:** automated guided vehicle; path planning; ivy algorithm; Gaussian-guided mechanism; time series similarity measurement; local optimum escape

## 0 引言

路径规划是智能无人驾驶研究的核心问题, 其目的是在已知或部分已知的环境中, 同时考虑路径的长度、安全性、平滑性和能耗等多种因素, 为无人车找到一条满足约束条件的最优路径. 路径规划已被证明属于典型的 NP 完全问题<sup>[1]</sup>, 其复杂度随环境规模和约束条件的增加呈指数级增长. 传统路径规

划算法, 如 A\*算法<sup>[2]</sup>和快速扩展随机树算法 (rapidly-exploring random tree, RRT)<sup>[3]</sup>, 已广泛应用于无人车系统. A\*算法在低维空间效率较高, 但在高维空间中面临“维度灾难”问题, 搜索空间和计算开销剧增, 难以实用<sup>[4]</sup>. RRT 算法适用于高维空间, 但生成路径质量通常不够理想<sup>[5]</sup>. 因此, 在复杂高维环境中, 传统规划算法普遍存在效率低、路径质量不足

收稿日期: 2025-03-21; 录用日期: 2025-11-13.

基金项目: 国家自然科学基金项目 (52472314).

<sup>†</sup>通信作者. E-mail: dupengzhen@njust.edu.cn.

等问题.

近年来,智能优化算法通过模拟自然界的生物行为或物理现象,在复杂环境下展现出强大的全局搜索能力与适应能力<sup>[6]</sup>.例如,蚁群算法通过信息素传递优选路径,但在初期搜索效率低,易陷入局部最优<sup>[7-8]</sup>;遗传算法依靠选择、交叉和变异机制提升寻优能力,但路径拐点多,平滑性较差<sup>[9-10]</sup>;粒子群优化算法通过粒子间的信息共享快速找到最优路径,但容易出现早熟现象,全局搜索能力不足<sup>[11-12]</sup>.

常春藤算法 (ivy algorithm, IVYA) 是 Ghasemi 等<sup>[13]</sup>于 2024 年提出的针对连续域变量的智能优化算法.该算法模拟了常春藤植物的生长模式,通过协调有序的种群生长以及常春藤植物的传播和进化来实现优化,兼具快速收敛能力与跳出局部最优的优势,表现出优异的求解性能<sup>[14]</sup>.

然而,路径求解面临离散化、维度不确定、连贯性约束等挑战,当前仍缺乏有效的建模方式将常春藤优化算法适配于路径规划问题.为此,本文提出一种基于改进常春藤算法的高斯引导路径规划方法.通过设计高斯引导机制,将离散约束的路径优化问题转化为引导信息的参数优化,在路径生成过程中有效引导,并结合路径质量评价反馈动态更新参数,逐步优化.在此基础上,算法充分利用路径的离散特性,显著压缩搜索空间,降低计算开销,并设计 3 项针对性改进策略:1) 解耦生长速率与初始位置并改进传播机制,以消除低值解偏好造成的路径搜索不均衡;2) 引入螺旋衰减更新,使生长速率在离散栅格中平缓可控,兼顾收敛速度与局部调优能力;3) 提出基于动态时间规整的陷阱检测与跳出机制,精准识别并打破优化停滞,提升全局探索能力.仿真实验结果表明,该方法在路径规划任务中取得更优的全局搜索性能和更高的收敛效率,为无人驾驶路径规划提供了一种有效解决方案.

## 1 基本常春藤优化算法

常春藤优化算法 (IVYA) 是一种新型的智能优化算法,灵感源自常春藤在自然界中的生长、攀爬和传播规律.算法将每个问题的可行解视为一株个体,并将其在  $d$  维搜索空间中均匀随机初始化,形成初始种群.随后,通过模拟常春藤的生长、追寻阳光进行攀爬以及传播进化等阶段,算法逐步演化出更优解,并采用“幸存者选择”机制筛选下一代种群.下文将对各阶段的数学模型与实现过程进行介绍.

### 1.1 协调有序的生长机制

常春藤在自然环境中会随着时间推移匍匐式生长,并在抵达可攀爬位置后向周围进行延展.为刻画

这一动态行为,IVYA 算法引入生长速率 (growth velocity, GV) 的概念,并假设其随时间演化具有一定的随机性.在个体初始化阶段,其生长速率由下式给出:

$$GV_i = I_i \odot (I_{\max} - I_{\min}). \quad (1)$$

其中:  $GV_i$  为第  $i$  株常春藤在离散时间系统中的生长速率;  $I_i$  为第  $i$  个体的位置向量,随机初始于  $d$  维空间;  $I_{\max}$  和  $I_{\min}$  分别为搜索空间的最大值和最小值向量;  $\odot$  为向量 Hadamard 除法,即向量逐元素除法.

在后续迭代中,个体的生长速率依据其当前状态与随机扰动动态调整,具体更新公式如下:

$$GV_i = \text{rand}^2 \odot (N^d(0, 1) \odot GV_i). \quad (2)$$

其中:  $\text{rand}$  为一个  $d$  维向量,其分量为独立均匀分布于区间  $[0, 1]$  的随机数;  $\text{rand}^2$  为对该向量逐元素平方后得到的新向量,作为期望为  $1/3$  的衰减因子;  $N^d(0, 1)$  为一个  $d$  维向量,其各分量独立服从均值为 0、标准差为 1 的标准正态分布;  $\odot$  为向量 Hadamard 乘积,即逐元素乘积.

### 1.2 漫游攀爬机制

在自然界中,常春藤在附近区域寻求可以攀爬的表面(如树干或墙壁),并优先借助已找到支撑物的其他常春藤,通过“强邻居”指引不断优化自身位置,以获得更好的“阳光”资源.为模拟常春藤通过“强邻居”指引优化自身位置,将  $\vec{I}_S = [I_1^s, \dots, I_j^s, \dots, I_{N_{\text{pop}}}^s]$  按照适应度值从最优到最差排序(即  $I_{\text{best}} = I_1^s$ ),则有

$$I_{ii} = \begin{cases} I_{j-1}^s, & I_i = I_j^s; \\ I_i, & I_i = I_{\text{best}}. \end{cases} \quad (3)$$

通过下式描述第  $i$  株常春藤利用“强邻居”  $I_{ii}$  进行生长,并在逻辑上靠近阳光:

$$I_i^{\text{new}} = I_i + |N^d(0, 1)| \odot (I_{ii} - I_i) + N^d(0, 1) \odot GV_i, \quad (4)$$

其中  $I_i^{\text{new}}$  为第  $i$  个新个体.

### 1.3 传播进化机制

当常春藤在搜索空间中漫游攀爬移动至最近最好位置时,依旧接受不到足够的阳光,会尝试向全局最优方向进一步移动.在算法中用种群最优解来模拟“最有利的攀爬点”,从而引导整个种群成员进行传播,其公式可表示为

$$I_i^{\text{new}} = I_{\text{best}} \odot (\text{rand} + N^d(0, 1) \odot GV_i). \quad (5)$$

### 1.4 幸存者选择机制

为模拟常春藤生命周期中“攀爬”和“传播进化”的交替生长,IVYA 引入对当前个体适应度与最

优适应度的比较策略: 当第*i*株常春藤的目标函数值  $f(I_i)$  小于最优  $f(I_{best})$  的  $\beta = (2 + \text{rand})/2$  倍数时, 算法将其视为进入“攀爬”阶段, 否则处于“传播进化”阶段。

此外, 为了模拟个体优胜劣汰的现象, 每次迭代结束后, 前一代算法的群体  $\vec{I}$  与当前迭代中新生成的群体  $\vec{I}^{new}$  合并, 即得到合并群体列表  $\vec{I}^{Merged} = \{\vec{I}, \vec{I}^{new}\}$ . 接着, 合并后的群体根据适应度值从最优到最差进行排序, 得到向量  $\vec{I}^{M/S} = [I_1^{M/S}, I_2^{M/S}, \dots, I_{N_{pop}}^{M/S}, I_{N_{pop}+1}^{M/S}, \dots, I_{2 \cdot N_{pop}}^{M/S}]$ . 取前  $N_{pop}$  个排名靠前的成员 (考虑算法群体大小保持常数  $N_{pop}$  不变) 作为当前群体的最终成员, 即

$$\vec{I} = [I_1^{M/S}, I_2^{M/S}, \dots, I_{N_{pop}}^{M/S}]. \quad (6)$$

## 2 应用于 ALV 全局路径规划的改进常春藤算法

### 2.1 路径规划问题建模

本文采用栅格法对地图建模, 将地图空间离散化为矩形栅格单元, 并依据车辆几何尺寸对障碍物单元适度膨胀, 以覆盖潜在碰撞区域, 确保规划路径的安全性. 如图 1 所示, 栅格的状态分为可通行状态 (白色栅格) 和占用状态 (黑色栅格).

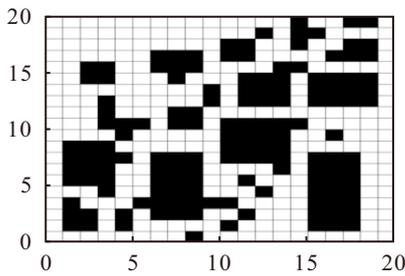


图1 栅格环境建模

**定义 1** 将离散化后的环境抽象为节点集合  $V = \{(i, j) | i = 1, 2, \dots, M, j = 1, 2, \dots, N\}$ ,  $(i, j)$  表示第  $i$  行、第  $j$  列栅格节点索引,  $M$  和  $N$  分别为栅格地图的行数与列数.

**定义 2** 函数  $\text{gridmap}(i, j)$  表示环境栅格的占用状态, 若  $\text{gridmap}(i, j) = 0$  则节点  $(i, j)$  可通行, 若  $\text{gridmap}(i, j) = 1$  则该节点被障碍物占据.

**定义 3** 路径解定义为有序节点序列  $\mathcal{P} = (p_1, p_2, \dots, p_n)$ , 其中  $p_i \in V, n \in \mathbb{N}, n \geq 2$ , 路径节点数  $n$  为大小不确定的正整数.

**定义 4** 路径节点间的连通关系由八邻域集合  $N_8(p) = \{(i + \Delta i, j + \Delta j) | \Delta i, \Delta j \in \{-1, 0, 1\}, (\Delta i, \Delta j) \neq (0, 0)\}$  定义, 即所在节点  $p$  后可向周围 8 个栅格移动. 其中: 直线栅格移动的距离定义为 1, 对角栅格移动的距离定义为 1.41, 真实的移动距离

可通过缩放比例信息进行还原.

基于上述定义, 无人车路径规划问题可建模为如下数学规划:

$$\begin{aligned} \min_{\mathcal{P}} J(\mathcal{P}) &= \sum_{k=1}^K \lambda_k C_k(\mathcal{P}). \\ \text{s.t. } p_1 &= \text{start}, p_n = \text{target}, \text{ (起点终点约束);} \\ p_{i+1} &\in N_8(p_i), \\ i &= 1, 2, \dots, n-1, \text{ (连贯性约束);} \\ \text{gridmap}(p_i) &= 0, \\ i &= 1, 2, \dots, n, \text{ (无碰撞约束).} \end{aligned} \quad (7)$$

其中:  $p_i = (x_i, y_i)$  为第  $i$  个路径节点在栅格中的坐标;  $J(\mathcal{P})$  为综合代价, 按权重对若干代价项加权求和. 典型代价项包括路径长度、平滑/拐点惩罚、安全裕度、能耗/时间等, 本文对比实验以路径长度  $L(\mathcal{P})$  与拐点数量  $T(\mathcal{P})$  为代表性项.

该优化问题属于 NP 困难问题. 在八邻域连通下, 每个栅格节点移动的分支因子为  $b = 8$ , 若从起点到终点的路径长度为  $n$ , 则所有可能路径序列的数量可表示为  $\mathcal{O}(b^n) = \mathcal{O}(8^n)$ ,  $n$  为路径长度且  $n \leq |V| = MN$ . 路径解的搜索空间随着栅格地图的规模呈指数级增长, 具有典型的组合爆炸特征, 同时其目标函数还包含平滑性、安全性等非线性代价项. 因此, 无人车路径规划问题通常被视为 NP 完全问题, 求解其最优解具有较高的计算复杂度.

### 2.2 高斯引导机制

在栅格离散化的无人车路径规划中, 主流元启发式方法多基于组合优化算法进行建模扩展, 典型代表包括遗传算法 (GA) 和蚁群算法 (ACO).

GA 算法采用基于序列组合的建模方式, 将初始路径序列作为编码, 通过选择、交叉和变异等遗传算子优化路径序列 (见图 2). 其中, 交叉算子通过重组两条父代路径的相交节点的后续序列产生新解, 变异则从随机路径节点后重新生成至终点的局部路径. 尽管该方法能够在已有路径序列的组合空间中实现较好的探索, 但缺乏对路径空间邻域의 针对性搜索, 难以在较优路径周围进行持续地路径调优.

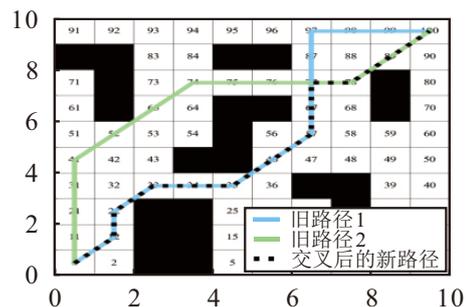


图2 遗传算法序列重组机制

ACO算法的核心思想是通过在路径节点上累积与挥发信息素来引导路径生成算法搜索(见图3)。在路径规划任务中,信息素能够有效表示历史路径节点的优劣,然而无法对其邻域节点提供有效刻画。这限制了算法在优质路径周围的探索能力,难以在已知解附近挖掘潜在更优路径,使得该算法需要较大种群规模以提升搜索空间覆盖率,才能有较好的效果。

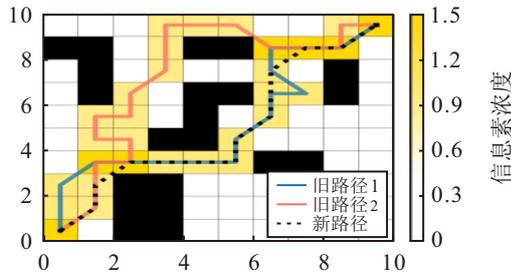


图3 蚁群算法信息素机制

上述两类方法表明,尽管这些建模机制能够高效探索已有路径的组合空间,但由于缺乏对路径空间邻域的刻画,在优质路径附近的调优能力有限。事实上,许多连续域智能优化算法在收敛速度和搜索能力方面表现优异,例如常春藤算法(ivy algorithm, IVYA)兼具快速收敛和较好的全局探索能力。然而,如第2.1节所述,路径解本质上是可变长度的离散节点序列 $\mathcal{P} = (p_1, p_2, \dots, p_n)$ ,必须满足式(7)所给出的起终点、连贯性和无碰撞约束,难以直接用固定维度的连续优化算法求解。

为此,本文提出一种基于高斯分布的权重引导机制(简称“高斯引导机制”),作为连续域优化算法迁移至路径规划问题的桥梁建模框架。该建模机制通过连续域优化算法迭代优化一组参数(引导点位置 $I_i^{\text{guide}}$ 、引导强度 $\sigma$ ),形成覆盖全栅格的引导权重分布,从而引导路径生成算法搜索。如图4所示,星形标记表示引导点位置,所在列对应的权重分布由曲

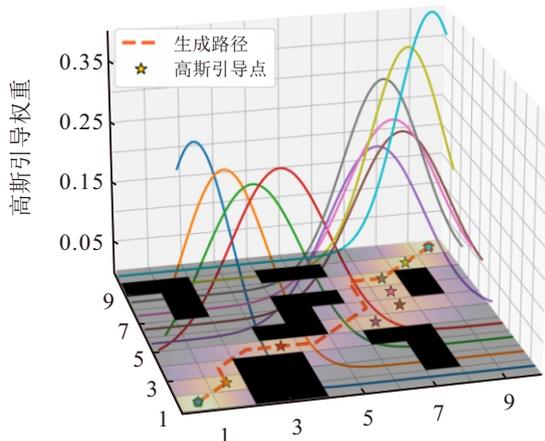


图4 高斯引导机制

线描绘,在该权重的引导下结合路径生成算法生成图中虚线所示的路径解。

具体而言,在每一轮迭代中,首先根据当前引导点参数生成栅格的权重信息。每列栅格由均值为 $I_i^{\text{guide}}$ 、标准差为 $\sigma$ 的高斯分布进行加权,其权重定义如下:

$$w_{\text{guide}}(i, j) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{l(i - I_i^{\text{guide}}[j]r)^2}{2\sigma^2}\right). \quad (8)$$

其中: $w_{\text{guide}}(i, j)$ 为第 $j$ 列第 $i$ 行的高斯引导权重; $I_i^{\text{guide}}[j]$ 为第 $j$ 列对应的引导点位置; $\sigma$ 为权重分布的宽度,用于控制引导强度。

随后根据高斯引导信息生成路径,并进行路径质量评价以评估当前引导点配置的优劣,将评价结果反馈给优化算法,更新下一轮的引导点位置和引导强度。通过这一闭环迭代过程,引导权重在智能优化算法的驱动下持续优化更新,使生成的路径不断改进。该机制可视为一种动态优化的启发式函数,为路径生成持续提供有效引导,并对路径空间邻域的引导信息进行充分表达,从而拥有较好的路径调优效果。

值得强调的是,高斯引导机制专为离散栅格路径规划设计,不同于连续域优化问题需逐维收敛至极高数值精度(如10的负数百次方级别),该方法只需在有限的离散行索引中确定该列引导点位置 $I_i^{\text{guide}}$ ,从而显著缩小搜索空间并大幅降低计算开销。

### 2.3 路径生成与局部优化算法

在路径规划过程中,每条路径解均为从起点到终点的无碰撞路径。本文提出的路径生成算法结合距离启发信息、平滑度启发信息以及高斯引导权重信息,从起点出发以轮盘赌的策略逐步搜索邻域栅格集构建可行路径。

#### 2.3.1 启发函数

在路径搜索过程中,每个候选栅格的选择基于以下权重计算:

1) 距离启发函数。

距离启发函数用于衡量候选栅格与目标点的接近程度,优先选择距离目标较近的栅格,具体公式如下:

$$w_{\text{distance}}(i_n, j_n) = \frac{d_{\text{max}}(\mathcal{S}) - d_{\text{min}}(\mathcal{S}) + 1}{d(i_n, j_n) - d_{\text{min}}(\mathcal{S}) + 1}. \quad (9)$$

其中: $(i_n, j_n) \in \mathcal{S}$ ,  $\mathcal{S}$ 为当前栅格current领域内且未访问的自由栅格集合,即可用于下一步扩展的栅格集合; $d(i_n, j_n)$ 表示 $(i_n, j_n)$ 节点到目标节点的欧氏距离; $d_{\text{max}}(\mathcal{S})$ 和 $d_{\text{min}}(\mathcal{S})$ 分别为 $\mathcal{S}$ 中所有节点到目标

节点的最大与最小距离.

### 2) 平滑启发函数.

平滑启发函数用于衡量路径的平滑程度, 鼓励路径在方向上保持连续性, 减少转弯, 使路径更加自然. 本文定义平滑启发函数如下:

$$w_{\text{smooth}}(i, j) = 1 + \gamma \cdot \frac{d_{\text{prev}} \cdot d_{\text{curr}}}{\|d_{\text{prev}}\| \|d_{\text{curr}}\|}. \quad (10)$$

其中:  $d_{\text{prev}} = (x_{\text{curr}} - x_{\text{prev}}, y_{\text{curr}} - y_{\text{prev}})$  为上一步移动方向,  $d_{\text{curr}} = (x_{\text{next}} - x_{\text{curr}}, y_{\text{next}} - y_{\text{curr}})$  为选择当前栅格所前进的方向;  $\gamma$  为影响程度参数, 可以控制平滑度的影响程度.

### 3) 综合启发式.

综合考虑距离、平滑度启发函数, 结合高斯引导机制, 提高路径规划的合理性, 综合启发信息的公式如下:

$$w(i, j) = w_{\text{distance}}(i, j) \cdot w_{\text{smooth}}(i, j) \cdot w_{\text{guide}}(i, j). \quad (11)$$

## 2.3.2 路径生成算法的相关定义与具体步骤

**定义 5** start 为路径起点, current 为当前栅格节点, target 为领终点.

**定义 6** 对于当前节点 current, 其可行域定义为  $\text{accessible}(\text{current}) = \{q \in N_s(\text{current}) | \text{gridmap}(q) = 0\}$ , 即由当前节点八邻域中所有可通行的栅格节点构成. 集合 visited 单独记录已访问的节点, 初始时  $\text{visited} = \emptyset$ , 在路径搜索过程中用于避免重复访问.

路径生成算法具体步骤如下所示.

**step 1:** 将当前栅格 current 设置为起点 start, 并且加入 path 列表与 visited 集合.

**step 2:** 根据当前栅格  $\text{accessible}(\text{current})$  和 visited 集合, 计算当前栅格 current 八邻域内未被访问的自由栅格集  $\mathcal{S}$ . 根据以下情形分别执行相应处理:

1) 若  $\mathcal{S}$  中包含 target, 则将 target 加入 path, 找到路径, 算法结束.

2) 若  $\mathcal{S}$  非空且 target 不在  $\mathcal{S}$  中, 则进入 step 3.

3) 若  $\mathcal{S}$  为空且 path 路径长度为 1, 则算法结束, 无可通行路径.

4) 若  $\mathcal{S}$  为空且路径长度不为 1, 则移除 path 最后一个路径点, 并将 current 更新为新路径的末端, 重新执行 step 2.

**step 3:** 结合距离、平滑度启发函数与高斯权重引导机制, 计算  $\mathcal{S}$  集合中每个待选栅格的权重, 如式 (11) 所示.

**step 4:** 以下式计算  $\mathcal{S}$  中待选栅格的概率权重  $P$ ,

通过轮盘赌的方式选出下一步栅格 next:

$$P(i_{\text{next}}, j_{\text{next}}) = \frac{w(i_{\text{next}}, j_{\text{next}})}{\sum_{(i_k, j_k) \in \mathcal{S}} w(i_k, j_k)}. \quad (12)$$

**step 5:** 将 next 加入 path 列表, 加入 visited 集合, 并且将当前栅格 current 设置为 next. 继续 step 2.

图 5 展示了在无障碍环境中受列高斯权重 (标准差为 2) 引导生成的路径. 栅格灰度表示高斯权重大小, 部分权重虽值较小但始终非零. 结果表明, 高斯引导机制能够通过固定数量的引导点构建栅格权重图, 间接完成优化过程的映射, 有效地引导生成路径.

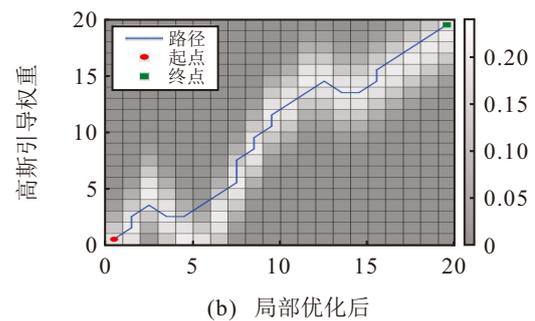
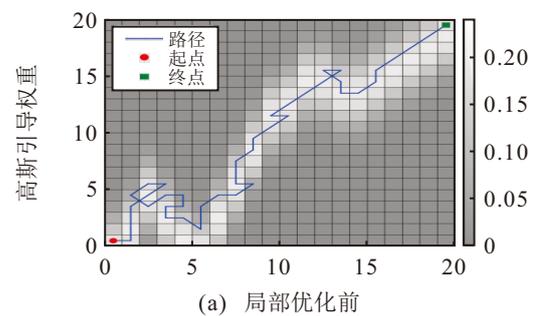


图5 高斯权重引导下生成的路径

## 2.3.3 路径的局部优化

由于路径生成算法具有一定的随机性, 生成的路径中不可避免地会出现“抖动”和“回环”现象, 如图 5(a) 所示. 为了解决这一问题, 采用一种简单的路径局部优化方法: 从路径的起点开始, 逐节点检查其八邻域范围内的后续路径节点, 若当前节点的八邻域内存在多个后续节点, 则直接将当前节点与最靠后的节点相连, 从而跳过中间的冗余节点. 经局部优化的路径如图 5(b) 所示.

## 2.4 改进生长速率机制

### 2.4.1 改进生长速率初始化

在原常春藤算法中, 生长速率 GV 的初始化机制与常春藤个体的初始位置密切相关, 如式 (1) 所示. 若常春藤个体初始位置位于低值区域 (靠近 0), 则 GV 值较小; 若位于高值区域, 则 GV 值较大. 这种设计导致初始位置在低值区域的个体生长速率较低,

探索能力弱,向高值解优化的速度慢.因此,算法在低值区域收敛速度比高值区域更快,出现明显的优化不均衡现象.在多峰测试函数中,这一局限性也使得算法更容易陷入低值区域的局部最优解.

为避免常春藤算法中低值搜索偏好,重新设计生长速率GV初始化机制为

$$GV_i^{\text{init}} = \frac{\text{MAXROW}}{10} \cdot \text{rand}. \quad (13)$$

其中:MAXROW为栅格地图的最大行数,rand为C维分量为[0,1]范围随机数的向量.这种设计使得生长速率GV的初始值能够适应不同规模的栅格环境,同时引入更多随机性以增强种群的灵活性.

#### 2.4.2 生长速率螺旋衰减更新策略

在本文提出的高斯引导机制中,将常春藤算法原本用于连续域的生长速率GV更新机制来控制引导强度,即高斯分布的宽度 $\sigma$ .然而,原始GV更新衰减速度过快,虽在高精度参数优化中表现良好,但在栅格图引导信息优化中会使高斯分布宽度过小,邻域权重趋近于零,导致中后期引导作用弱,优化收敛变慢,甚至过早停滞.经过大量实验测试,本文算法采用螺旋衰减更新策略来更新生长速率,有

$$GV_i = GV_i^{\text{init}} \cdot (1 + \alpha \cdot \sin(\omega T)) \cdot e^{-\beta \cdot T}. \quad (14)$$

其中: $T$ 为当前个体存活的迭代次数, $\alpha$ 为调制参数, $\beta$ 为衰减参数, $\omega$ 为频率参数.

该螺旋衰减策略通过周期性调制保证了初期较大的邻域搜索能力,并通过参数可控的衰减方式,使后期搜索过程逐步收敛,避免周围引导信息的缺失而导致局部调优能力弱,从而兼顾收敛速度与解的质量.

### 2.5 高斯引导机制下的攀爬和传播方式

对于服从均值为 $\mu$ 、标准差为 $\sigma$ 高斯分布的随机变量 $X$ ,即 $X \sim \mathcal{N}(\mu, \sigma^2)$ ,若存在随机变量 $Y = C_1 X + C_2$ ,其中 $C_1$ 、 $C_2$ 为常数,则 $Y$ 服从均值为 $C_1 \mu + C_2$ 、标准差为 $|C_1| \sigma$ 的高斯分布,即 $Y \sim \mathcal{N}(C_1 \mu + C_2, C_1^2 \sigma^2)$ .

基于这一高斯分布的变换特性,将常春藤算法的“攀爬”和“传播进化”机制,与高斯权重引导机制巧妙结合进行等效设计.为了在固定维度上表征上一代路径解,本文将路径每列栅格中经过的最大行栅格集合作为采样锚点集,数学化表示如下:

$$A_i = \{(r_{\max}(c), c) | c \in \{1, 2, \dots, C\}\}. \quad (15)$$

其中: $i$ 为个体在种群中的索引, $r_{\max}(c)$ 为路径在第 $c$ 列中经过的最大行栅格索引, $C$ 为栅格地图的列数.

基于上一代个体的锚点集 $A$ ,通过常春藤算法

的“攀爬”与“传播进化”机制优化生成新个体,即引导点位置与引导参数,从而构建整图的高斯权重,最终由轮盘赌形式的路径生成算法将权重信息坍缩生成符合约束的路径解.

#### 2.5.1 攀爬机制

在常春藤算法中,攀爬机制是重要的邻域搜索步骤.针对路径规划的离散特性,本文通过攀爬机制更新引导点位置和引导强度信息,进而更新路径解.攀爬机制的引导点位置和宽度更新采用如下形式:

$$\begin{cases} I_i^{\text{guide}} = A_i + |N^d(0, 1)| \odot (A_{ii} - A_i), \\ \sigma_i = GV_i. \end{cases} \quad (16)$$

其中: $I_i^{\text{guide}}$ 与 $\sigma_i$ 分别为第 $i$ 个新个体的引导点向量与高斯分布宽度, $A_i$ 为原种群个体 $i$ 的路径采样锚点集, $A_{ii}$ 为个体 $i$ “强邻居”的采样锚点集(同式(3)).

#### 2.5.2 改进传播进化机制

传播进化机制通过学习最优个体并在其邻域加强探索生成新解,以保持优质解特征并增强全局探索能力.但是从原来的式(5)可以看出,原传播进化机制存在与第2.3节所述“低值偏好”相似的问题:一方面,处于较高值区域的个体在更新时具有更大的搜索范围,而较低值区相对受限;另一方面,算法倾向往最优解低值一侧进行移动,从而进一步加剧了对“低值区域”的偏好,影响了全局搜索性能.本文为克服“低值偏好”带来的不均衡性,改进传播进化机制的高斯引导点和宽度更新如下:

$$\begin{cases} I_i^{\text{guide}} = A_{\text{best}} + \lambda \cdot \text{rand}(-1, 1), \\ \sigma_i = GV_i^{\text{init}}. \end{cases} \quad (17)$$

其中: $A_{\text{best}}$ 为适应度最优个体的路径采样锚点集, $\text{rand}(-1, 1)$ 为分量是 $[-1, 1]$ 的独立随机数组成的 $d$ 维向量, $\lambda$ 为用于控制在最优位置周围的扰动步长, $GV_i^{\text{init}}$ 为由式(13)计算的初始生长速率.

### 2.6 局部陷阱逃脱策略

智能优化算法在迭代过程中常因个体趋同而陷入局部最优,导致探索能力下降甚至完全停滞.要实现“探索”与“开发”阶段的动态切换,关键在于准确判定搜索是否停滞.然而,仅依赖评价函数下降的速度难以全面反映这一状态,尤其在离散搜索空间中,更应关注种群的路径解是否趋于稳定,以更准确地识别停滞,从而恢复多样性并增强全局探索能力.

针对这一问题,本文提出了基于动态时间规整(dynamic time warping, DTW)的局部陷阱检测方法,以实现局部最优陷阱的准确识别和及时规避.

对于两个长度分别为 $m$ 和 $n$ 的序列 $X = (x_1, x_2, \dots, x_m)$ 、 $Y = (y_1, y_2, \dots, y_n)$ .定义距离矩阵

$D(i, j)$  表示序列  $X$  中第  $i$  个点与序列  $Y$  中第  $j$  个点之间的距离, 通常采用欧氏距离计算. DTW 通过动态规划构建累积距离矩阵  $C$ , 其中每个元素  $C(i, j)$  表示从起点  $(1, 1)$  到点  $(i, j)$  的最小累积距离, 以此找到最优匹配方式. 其递推关系如下:

$$C(i, j) = D(i, j) + \min \begin{cases} C(i-1, j), \\ C(i, j-1), \\ C(i-1, j-1), \end{cases} \quad (18)$$

其中初始条件为  $C(1, 1) = 0$ .

最终, DTW 距离定义为最优对齐路径的累积距离, 即  $DTW(X, Y) = C(m, n)$ . 本文中, 若两条路径的 DTW 距离小于设定阈值 (栅格列数的  $1/4$ ), 则视为路径近似相等. DTW 作为一种时间序列相似性度量方法, 能够有效度量不同长度、非线性变形的路径相似性.

本文利用 DTW 算法进行局部陷阱检测. 对适应度进行稳定排序后的种群, 最优个体与最差个体若满足以下条件则认为陷入局部陷阱: 在连续 3 次迭代中, 二者的适应度值相等, 且路径近似相同. 一旦检测到局部陷阱, 种群将仅保留当前最优个体, 其余个体将在全局范围内通过随机路径生成算法重新初始化, 以增强种群的多样性并提升探索能力. 随后, 利用常春藤优化算法对种群进行优化, 使算法在跳出局部陷阱后能够持续高效搜索并优化路径解, 从而避免算法停滞后的无效迭代.

## 2.7 改进算法的总体步骤

step 1: 设置种群大小、最大迭代次数等参数.

step 2: 基于距离和平滑启发信息生成随机初始路径, 并计算适应度排序.

step 3: 对初始路径通过式 (15) 生成采样锚点集, 作为初始引导点位置. 根据式 (13) 初始化生长速率  $GV$ , 作为表示引导强度的参数.

step 4: 若迭代次数未达到最大值, 则继续迭代; 否则算法终止, 输出最优路径.

step 5: 根据上一代个体的路径适应度评价, 优化生成新个体的引导参数 (引导点位置和引导强度). 如果旧个体适应度小于最优  $f(I_{best})$  的  $\beta = (2 + \text{rand})/2$  倍数, 则通过攀爬式 (16) 更新引导点和宽度; 否则, 通过传播进化式 (17) 更新引导点和宽度.

step 6: 在距离、平滑启发信息及高斯引导点的引导作用下, 利用路径生成算法生成新的路径解. 计算新个体的采样锚点集, 作为下轮优化的引导点位置基准, 同时初始化新个体的生长速率  $GV$ , 将新个体加入种群.

step 7: 根据生长速率螺旋衰减策略 (式 (14))

更新原种群个体的生长速率  $GV$ .

step 8: 对种群进行适应度排序, 并通过幸存者选择机制 (式 (6)) 筛选个体.

step 9: 利用 DTW 进行局部陷阱检测, 若在连续 3 次迭代中, 最优个体与最差个体的适应度相等, 且路径近似相同 (DTW 距离小于阈值), 则判定陷入局部陷阱, 进入 step 10; 若未检测到局部陷阱, 则进入 step 4.

step 10: 仅保留当前最优个体, 其余个体在全局范围内通过随机路径生成算法重新初始化, 并更新相关参数, 跳至 step 3 继续迭代.

## 2.8 时间复杂度分析

本算法的计算复杂度主要分为初始化阶段和迭代阶段. 将种群规模记为  $N_{pop}$ , 最大迭代次数为  $Iter_{max}$ , 单条路径生成的平均时间复杂度记为  $O(Q)$ , 引导点数量 (参数优化的维度) 记为  $D$ .

在初始化阶段, 需要为整个种群生成初始路径, 其复杂度为  $O(N_{pop} \cdot Q)$ . 随后需要计算适应度并进行排序, 前者复杂度为  $O(N_{pop})$ , 后者复杂度为  $O(N_{pop} \log N_{pop})$ . 因此, 初始化阶段的总体复杂度为  $O(N_{pop}Q + N_{pop} \log N_{pop})$ . 迭代阶段中, 每轮迭代的主要计算量来自 3 个部分: 引导参数的更新计算复杂度为  $O(N_{pop} \cdot D)$ ; 本轮迭代路径生成复杂度为  $O(N_{pop} \cdot Q)$ ; 路径适应度计算和幸存者选择的种群排序的复杂度分别为  $O(N_{pop})$  和  $O(N_{pop} \log N_{pop})$ . 局部陷阱检测仅需比较当前最优个体与最差个体的路径相似度, 开销为常数级  $O(1)$ , 对整体复杂度影响可忽略. 将这些步骤合并后, 每轮迭代的时间复杂度可简化表示为  $O(N_{pop}D + N_{pop}Q + N_{pop} \log N_{pop})$ .

结合最大迭代次数  $Iter_{max}$ , 算法总体时间复杂度为  $O(Iter_{max} \cdot [N_{pop}D + N_{pop}Q + N_{pop} \log N_{pop}]r)$ .

## 3 仿真实验分析

仿真实验运行环境为: macOS 14.0, PyCharm 2023, 处理器 Apple M2 Pro, 主频率 3.5 GHz, 内存 16 GB.

实验中, MF-ACO、GFA 和 PGWO 算法的参数配置分别依据其原文献<sup>[15-17]</sup>. 本文改进算法 (I-IVYA)、基本蚁群算法 (ACO)、遗传算法 (GA)、粒子群算法 (PSO)、差分进化算法 (DE) 的参数配置如表 1 所示.

### 3.1 低值偏好验证分析

本文针对低值偏好问题进行修正, 主要包括两方面: 解耦  $GV$  初始值与个体位置, 以提高不同位置的扩散均衡性 (第 2.4.1 节所述); 优化传播进化机制,

表1 算法实验参数

算法	参数
ACO	$N_p = 100, \text{Iter}_{\max} = 100, \alpha = 1, \beta = 3, \rho = 0.6, q = 100$
GA	$N_p = 100, \text{Iter}_{\max} = 100, P_{\text{selection}} = 0.9, P_{\text{mutation}} = 0.15$
I-IVYA	$N_p = 20, \text{Iter}_{\max} = 100, \lambda = 2, \alpha = 0.5, \beta = 0.3, w = 3, \gamma = 0.5$
PSO	$N_p = 20, \text{Iter}_{\max} = 100, w = 0.8, c_1 = 1.5, c_2 = 1.5, v_{\max} = 3.0$
DE	$N_p = 20, \text{Iter}_{\max} = 100, F = 0.6, \text{CR} = 0.9$

使算法在最优值周围探索更全面(第2.5.2节所述)。为了验证改进算法对低值解偏好的修正,将本文提出的原算法与改进算法在两个测试函数上进行实验评估,测试函数分别为标准 Sphere 函数  $f_1$  和偏移 Sphere 函数  $f_2$ 。实验的参数设置如下:种群规模  $N = 30$ ,搜索维度  $D = 5$ ,最大迭代次数  $T = 100$ 。

测试函数 1(标准 Sphere 函数):

$$f_1(x) = \sum_{i=1}^D x_i^2.$$

测试函数 2(偏移 Sphere 函数):

$$f_2(x) = \sum_{i=1}^D (x_i - 50)^2.$$

将原常春藤优化算法(IVYA)和改进常春藤算法(I-IVYA)分别对测试函数  $f_1$ 、 $f_2$  进行实验。

图6展示了改进前的收敛情况,红色线与蓝色线分别为原常春藤优化算法  $f_1$  和  $f_2$  测试函数的收敛情况。 $f_2$  函数解仅对  $f_1$  函数解进行水平偏移,但原常春藤算法对  $f_1$  的优化收敛速度明显大于对  $f_2$ ,原算法对低值解具有明显的偏好性。绿色线与黑色线为算法改进后对  $f_1$ 、 $f_2$  测试函数的优化收敛情况,表明其对低值和高值算法都有较好的收敛效果,克服了偏好问题。

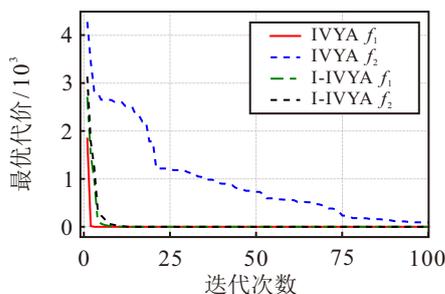
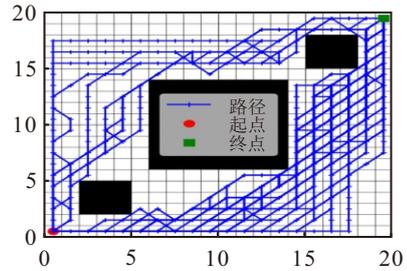


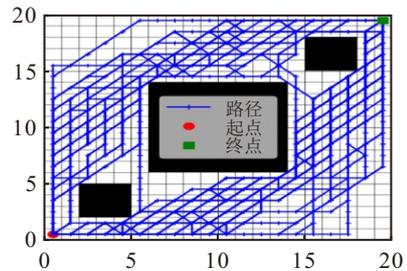
图6 改进前后的收敛情况

在路径规划任务中,常春藤算法的搜索偏好性导致路径探索的不均衡性,主要集中于地图的下半部分。图7(a)展示了原始算法在寻优过程中的路径

覆盖情况,显然,其优化搜索更倾向于低值区域,导致探索范围受限,路径分布不均衡。这种不均衡性将会影响最终规划结果的全局最优性,使得某些潜在的优质路径未能被充分挖掘。本文对常春藤算法进行了搜索偏好修正,图7(b)为改进后算法的路径覆盖情况,改进算法在搜索均衡性上的优化,有助于降低局部最优陷阱的影响,提高路径的可行性和规划效率。



(a) 原算法的优化覆盖情况



(b) 修正偏好的优化覆盖情况

图7 偏好修正前后的优化路径覆盖情况

### 3.2 消融实验分析

本节对改进算法进行消融实验,以评估各个改进策略对算法性能的影响。实验在图1所示环境下进行,每种方法独立运行50次,并记录最短路径长度的平均路径长度、平均迭代次数及路径最优率作为评估指标,以分析各策略的优化效果。

表2展示了低值偏好修正、生长速率GV螺旋衰减、局部陷阱逃离3种单策略改进及混合策略对常春藤算法路径优化性能的影响。从实验数据可以看出,原始常春藤算法在50次运行中最短路径长度与各改进算法保持一致,但在平均路径长度、平均迭代次数和路径最优率方面存在明显差异,反映出其优化全局寻优、稳定性和收敛性能较弱。低值偏好修正策略能够有效提升路径最优率(从42%提高至80%),同时减少平均路径长度,表明该策略可以避免过度集中于低值区域,使优化搜索更加均衡。引入生长速率螺旋衰减策略来控制引导强度(高斯宽度),求解的平均路径长度降低至31.5987,路径最优率提升至50%,且平均迭代次数有所减少。该结果体现了螺旋衰减策略通过可控的衰减速度,使得算法在中

后期保持适当的分布宽度,避免路径周围引导权重趋于0,保留了较强的引导信息,从而提升解的质量与搜索效率.局部陷阱检测逃离策略在检测到搜索停滞重新启动更大范围的探索,因此平均收敛次数增加至12.6,收敛速度有所减缓.然而,该策略显著提升了求解质量,使平均路径长度降低至31.4101,路径最优率提升至62%,能够有效跳出局部最优并增强全局搜索能力.

表2 消融实验的实验对比数据

算法	原常春藤算法	低值偏好修正	GV螺旋衰减策略	局部陷阱逃离策略	混合改进
最短路径长度	30.9705	30.9705	30.9705	30.9705	30.9705
平均路径长度	31.7193	31.1662	31.5987	31.4101	31.0225
平均迭代次数	11.9	9.8	11.2	12.6	10.3
路径最优率/%	42	80	50	62	98

结合3种改进方法的混合策略算法在所有指标上均表现最佳,平均路径长度进一步降低至31.0225,平均迭代次数减少至10.3,路径最优率大幅提升至98%.这一结果表明,融合低值偏好修正、生长速率螺旋衰减和局部陷阱检测的混合策略具有协同增益效果,能够有效增强算法的搜索能力,提高解的稳定性和收敛效率,实现路径质量、计算效率和稳定性的全面提升,进一步验证了本文的改进策略在优化路径规划算法中的有效性和优越性.

### 3.3 对比实验分析

无人车路径规划问题是一个多维复杂优化问题,其优劣不仅取决于路径长度,还与转弯次数、路径安全度等多因素密切相关.本文对比实验选取路径长度 $L(\mathcal{P})$ 和路径拐点数量 $T(\mathcal{P})$ 作为代表性优化指标,并通过加权的方式将多目标优化转化为单目标优化,其适应度函数定义为

$$f(\mathcal{P}) = \theta \cdot L(\mathcal{P}) + (1 - \theta) \cdot T(\mathcal{P}), \quad (19)$$

其中 $\theta \in [0, 1]$ 为权重参数,本文设置 $\theta = 0.75$ .

在此优化目标下,本节共设计4组仿真对比实验.对比实验1主要用于评估建模方式和未改进的基础算法,验证本文提出的高斯引导机制建模方式及其IVYA作为基础优化算法的有效性.对比实验2~对比实验4则选取近年来性能较优的改进算法作为对照,以进一步验证本文改进算法在高水平基线下的优越性.为保证对比的公平性,避免因参数调优带来的性能偏差,对比实验2~对比实验4的地图环境及算法参数均严格按照对照算法文献中的设置进行配置.其中,对比实验2~对比实验4中还加入

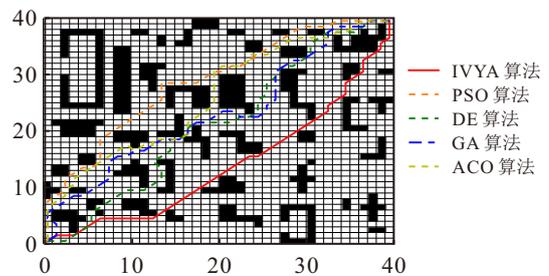
了A\*算法的对比,A\*算法属于纯搜索算法的优化,无法对路径拐点数量等非线性目标进行优化,但其在路径长度上可以达到理论最优,实验中作为最优路径长度的参考.

#### 3.3.1 对比实验1

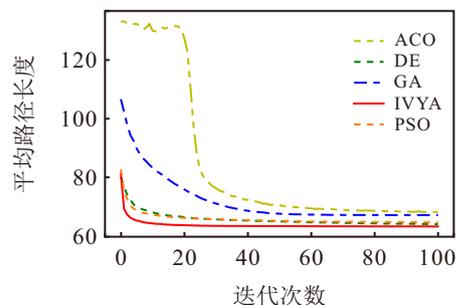
为验证所提出高斯引导机制在离散路径规划问题中的有效性,并评估常春藤算法(IVYA)作为基础算法的优势,设计对比实验1.实验选取了信息素引导的蚁群算法(ACO)、序列重组的遗传算法(GA),以及结合本文提出的高斯引导机制的常春藤算法(IVYA)、粒子群算法(PSO)和差分进化算法(DE).为保证对比的公正性,所有智能优化算法均采用未经改进的基础版本,并独立运行100次,取其统计平均值作为最终结果.表3给出了不同建模方式与基础算法的实验结果.图8展示了路径规划及收敛情况.

表3 不同建模方式与基础算法的对比数据

建模方式	基础算法	平均路径长度	路径长度标准差	平均迭代次数	平均拐点数量	时间/s
高斯引导机制	IVYA	62.9515	0.8068	29.24	17.51	2.14
高斯引导机制	PSO	65.1974	1.0049	48.19	19.13	2.17
高斯引导机制	DE	64.6468	1.1214	66.62	18.36	2.53
序列重组机制	GA	66.7926	2.3219	48.59	28.25	3.48
信息素机制	ACO	67.8281	2.9196	67.03	23.44	5.33



(a) 路径规划



(b) 收敛情况

图8 对比实验1

从表3结果可以看出,基于高斯引导机制建模的3种算法(IVYA、PSO、DE)在平均路径长度、路径长度标准差和平均拐点数量等指标上均优于序列重

组机制的 GA 和信息素机制的 ACO. 这体现了本文提出的高斯引导机制能够有效地将固定维度连续域优化算法应用于离散路径规划问题, 在解的质量和结果的稳定性方面均优于两类经典的组合优化扩展算法. 同时, 3 种基于高斯引导机制的算法在平均迭代次数和平均运行时间上明显低于 GA 与 ACO 算法, 表现出更高的计算效率. 这主要得益于高斯引导机制能够充分刻画已有路径邻域的引导信息, 强化局部优化能力, 使算法在较小种群规模下即可获得高质量解, 从而降低整体求解时间并提高收敛效率.

进一步比较 3 种采用高斯引导机制的基础算法可以发现: DE 虽具有较强的跳出局部最优的能力, 能够在优化中后期持续改善解的质量并最终超过 PSO, 但其整体收敛速度相对较慢; PSO 在优化初期收敛速度较快, 却易陷入局部最优而难以继续提升; 而 IVYA 同时兼具快速收敛与跳出局部最优的能力, 最终获得了最优的路径长度和最少的拐点, 表现出最佳的综合性能. 这些结果共同验证了高斯引导机制在路径规划问题中的有效性, 并突出了常春藤算法作为基础算法的优势.

### 3.3.2 对比实验 2

本实验将 I-IVYA 与 A\*、ACO 及改进蚁群算法 MF-ACO 进行对比, 实验地图及 MF-ACO 的参数设置均参考其原始文献 [15]. 图 9 展示了 4 种算法在 30 × 30 地图上的 100 次独立实验的平均收敛曲线及路径规划结果. 表 4 给出了本次实验的详细统计数据.

从路径质量来看, 4 种算法均能在 100 次实验中

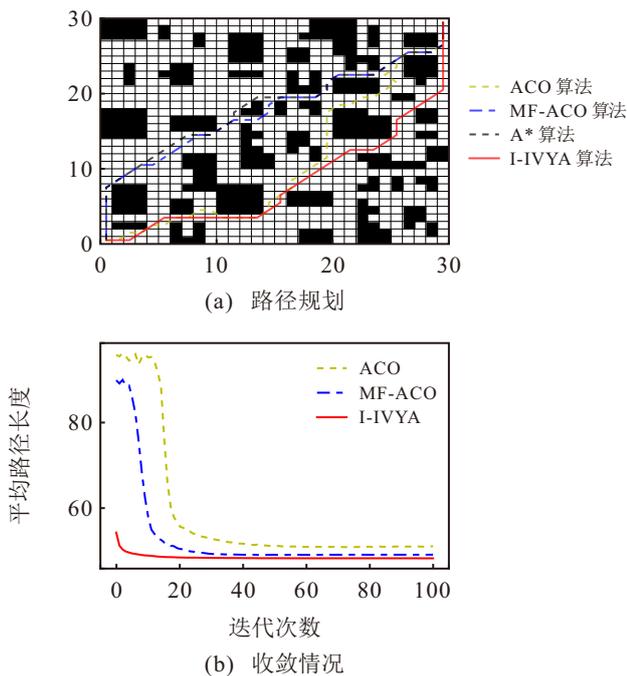


图9 对比实验 2

表4 30 × 30 地图实验对比数据

算法	A*	ACO	MF-ACO	I-IVYA
最短路径长度	48.0416	48.0416	48.0416	48.0416
平均路径长度	48.0416	50.1486	48.6772	48.0709
路径长度标准差	—	1.2494	0.5412	0.2141
平均迭代次数	—	49.08	28.19	16.71
平均拐点数	14	19.97	17.16	11.33
路径最优率/%	100	6	35	96

找到理论最短路径 (48.0416), 但 I-IVYA 的平均路径长度为 48.0709, 优于 ACO (50.1486) 和 MF-ACO (48.6772). 同时, I-IVYA 在 100 次实验中的最优路径获得率高达 96%, 远高于 ACO (6%) 和 MF-ACO (35%), 且 I-IVYA 的路径长度标准差最小, 仅为 0.2141. 这表明 I-IVYA 算法不仅在求解质量上表现出色, 而且在多次实验中展现出较高的稳定性, 能够在大多数情况下找到最优路径解. 在收敛效率方面, I-IVYA 同样表现突出, 其平均迭代次数为 16.71, 显著低于 ACO (49.08) 和 MF-ACO (28.19), 体现了更快的收敛速度. 此外, I-IVYA 生成的路径平均拐点数为 11.33, 相较于 ACO (19.97) 和 MF-ACO (17.16) 明显减少, 表明其路径更为平滑, 转弯次数更少, 更符合无人车的实际行驶需求. 该优势在路径可视化结果中也得到了直观印证.

### 3.3.3 对比实验 3

本实验将 I-IVYA 与 A\*、GA 和 萤火虫遗传算法 (GFA) 进行对比, 实验地图及 GFA 的参数设置均参考其原始文献 [16]. 图 10 展示了 4 种算法在该

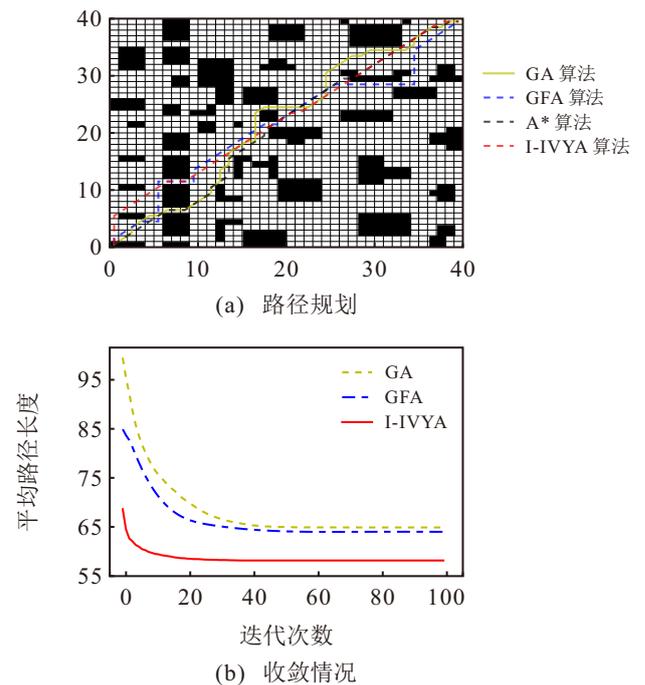


图10 对比实验 3

40 × 40 地图上 100 次独立实验的平均收敛曲线及对应的路径规划结果, 表 5 给出了本次实验的详细统计数据.

表5 40 × 40 地图实验对比数据

算法	A*	GA	GFA	I-IVYA
平均路径长度	58.0832	66.8763	66.2650	58.4297
路径长度标准差	—	1.8576	0.4502	0.6681
平均拐点数	13	25.78	13.55	7.49
平均迭代次数	—	45.76	51.22	27.65

从路径长度来看, I-IVYA 算法在保持较短路径方面表现最优, 平均路径长度为 58.4297, 显著低于 GA 算法 (66.8763) 和 GFA 算法 (66.2650), 这表明 I-IVYA 在全局搜索过程中能更高效地获取高质量路径解. 尽管 I-IVYA 的路径长度标准差为 0.6681, 略高于 GFA 算法的 0.4502, 但在求解质量和收敛速度上, I-IVYA 仍具有明显优势. I-IVYA 的平均路径长度更接近理论最优值, 同时在路径平滑性方面表现突出, 平均拐点数仅为 7.49, 远低于 GA (25.78) 和 GFA (13.55), 这表明 I-IVYA 生成的路径更平滑、连续性更强, 更符合实际应用中对轨迹可行性和导航安全性的要求. 此外, I-IVYA 在收敛速度上也优于其他算法, 平均迭代次数为 27.65, 明显少于 GA (45.76) 和 GFA (51.22), 这意味着 I-IVYA 能在更少的迭代次数内接近最优解, 显著提升了计算效率.

### 3.3.4 对比实验 4

本实验将 I-IVYA 与 A\*、PWGO 和 IVYA 进行对比, 实验地图与 PWGO 的参数配置严格参考其原始文献 [17]. 考虑到 PWGO 在 100 次迭代内未能有效收敛, 为对比完全收敛的效果, 本实验将所有算法的最大迭代次数设置为 500. 图 11 展示了 4 种算法在 50 × 50 地图上进行 100 次独立实验的平均收敛曲线及路径规划结果, 表 6 给出了本次实验的详细统计数据.

I-IVYA 算法在路径优化质量方面展现出显著优势, 其平均路径长度为 73.8108, 接近 A\*算法的 72.8111, 优于原始 IVYA 算法 (75.6267) 和 PWGO 算法 (75.29), 验证了改进策略在提升路径质量方面的有效性. 改进后的常春藤算法具备更强的搜索能力, 能够更高效地寻优近似最优路径. 在路径平滑性方面, I-IVYA 同样表现优异, 平均拐点数为 13.84, 明显低于 PWGO (21)、原始 IVYA (14.69) 及 A\*算法 (15), 表明所生成路径更为平滑, 有助于提高无人车在实际行驶中的控制稳定性与跟踪精度. 在运行

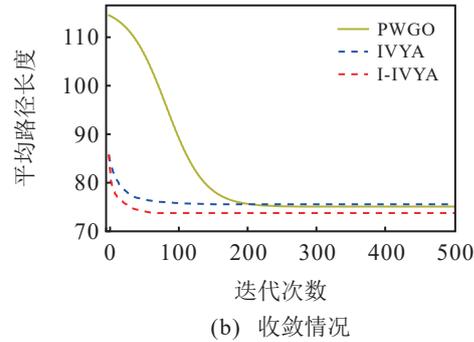
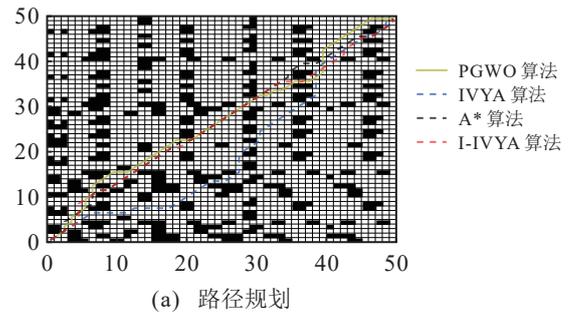


图11 对比实验 4

表6 50 × 50 地图实验对比数据

算法	A*	PWGO	IVYA	I-IVYA
平均路径长度	72.8111	75.29	75.6267	73.8108
平均拐点数	15	21	14.69	13.84
运行时间/s	0.371	5.283	3.804	3.635

时间方面, A\*算法表现最优, 仅需 0.371 s. 其次为 I-IVYA 算法, 平均运行时间为 3.635 s, 优于 PWGO (5.283 s), 也略优于原始 IVYA (3.804 s). 尽管 A\*算法在运行时间上占据绝对优势, 但其在路径平滑性和综合优化能力方面不如 I-IVYA, 难以满足复杂环境中对路径的多重需求. 相比之下, I-IVYA 在兼顾运行效率的同时, 提供了更优的路径质量和更强的综合性能.

### 3.4 实验总结

本节通过多组实验验证了所提出高斯引导机制、改进策略的有效性及其在路径规划任务中的性能优势. 低值偏好验证实验表明, 两处偏好调整有效克服了原算法的偏好局限性, 使搜索更加均衡. 消融实验结果显示, 低值偏好改进、生长速率螺旋衰减策略和基于 DTW 检测的局部陷阱逃离策略均能提升算法性能, 其中低值偏好改进与局部陷阱逃离策略对求解质量和稳定性提升尤为显著. 对比实验 1 验证了所提出高斯引导机制能够有效地将固定维度的连续域优化算法应用于离散路径规划问题, 并在求解质量和效率上超越了基于组合优化拓展的 GA、ACO 两个经典算法. 对比实验 2 ~ 对比实验 4 表明, 本文提出的 I-IVYA 算法在不同环境下均表现优异.

I-IVYA 算法在与 MF-ACO、GFA、PGWO 等近年来表现优秀的改进型路径规划算法的对比中,不仅在不同规模和复杂度的环境下找到更高质量的路径解,而且在路径平滑性、收敛速度及稳定性方面展现出显著优势.这些特性提升了算法的实际适用性,为复杂环境下的无人车路径优化提供了一种高效、稳健的解决方案.

## 4 结论

针对常春藤算法等固定维连续域优化方法难以直接处理维度不固定且存在连贯性约束的离散路径规划问题,本文提出了一种基于高斯引导机制的改进常春藤优化算法.高斯引导机制作为一种应用建模方式,将离散的路径搜索优化转化为对引导参数的优化,通过迭代优化引导点位置和分布宽度,引导路径生成算法,不断改进路径解.此外,本文针对路径规划的应用特性,对常春藤算法进行专门优化:充分利用离散特性降低计算量;低值偏好修正提升了搜索的均衡性;螺旋衰减策略针对路径规划引导信息提供了更加可控且合适的引导分布宽度衰减方式;基于 DTW 算法的局部最优逃脱机制通过有效检测路径趋同,精确识别优化停滞问题,从而避免局部陷阱内的无效探索.这些改进有效提升了搜索的均衡性和全局探索能力,同时增强了求解质量与算法的稳定性.

仿真对比实验表明,所提出的高斯引导建模方式在路径优化中具备更强的搜索表达能力,其求解效果显著优于传统组合优化建模方法,如序列重组的 GA 遗传算法和信息素驱动的 ACO 蚁群算法.并且在优秀的改进算法对比中,本文提出的 I-IVYA 算法在最优路径长度、平均路径长度、收敛速度及路径平滑性方面均优于 MF-ACO、GFA、PGWO 等智能优化算法.因此,本文所提出方法不仅提升了常春藤算法在路径规划任务中的适用性,而且为连续域智能优化算法在离散路径规划领域的应用提供了新的方法思路.未来工作将聚焦于计算效率的提升,计划采用 GPU 加速和多线程等并行计算技术,进一步增强算法性能与工程适配能力.

## 参考文献 (References)

- [1] Liu L X, Wang X, Yang X, et al. Path planning techniques for mobile robots: Review and prospect[J]. *Expert Systems with Applications*, 2023, 227: 120254.
- [2] Tang G, Tang C Q, 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.
- [3] Zammit C, van Kampen E J. Comparison between A\* and RRT algorithms for 3D UAV path planning[J]. *Unmanned Systems*, 2022, 10(2): 129-146.
- [4] Zhang D X, Chen C, Zhang G Y. AGV path planning based on improved A-star algorithm[C]. 2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference. Chongqing, 2024: 1590-1595.
- [5] 孙家玮, 余明晖, 杨大鹏, 等. 基于 CL-RRT 与 MPC 的舰载机牵引系统路径规划[J]. *系统工程与电子技术*, 2024, 46(5): 1745-1755.  
(Sun J W, Yu M H, Yang D P, et al. Path planning of carrier aircraft traction system based on CL-RRT and MPC[J]. *Systems Engineering and Electronics*, 2024, 46(5): 1745-1755.)
- [6] 孙哲, 马胜男, 解相朋, 等. 基于仿生算法的多式联运路径规划方法综述[J]. *控制与决策*, 2025, 40(2): 375-386.  
(Sun Z, Ma S N, Xie X P, et al. Bio-inspired optimization-based path planning algorithms in multimodal transportation: A survey[J]. *Control and Decision*, 2025, 40(2): 375-386.)
- [7] 肖鹏, 田润澜, 李赫, 等. 基于改进蚁群算法的无人机通信侦察航迹规划[J]. *控制与决策*, 2025, 40(11): 3239-3252.  
(Xiao P, Tian R L, Li H, et al. UAV communication reconnaissance path planning based on improved ant colony algorithm[J]. *Control and Decision*, 2025, 40(11): 3239-3252.)
- [8] 张恒, 何丽, 袁亮, 等. 基于改进双层蚁群算法的机器人路径规划[J]. *控制与决策*, 2022, 37(2): 303-313.  
(Zhang H, He L, Yuan L, et al. Mobile robot path planning using improved double-layer ant colony algorithm[J]. *Control and Decision*, 2022, 37(2): 303-313.)
- [9] Ab Wahab M N, Nazir A, Khalil A, et al. Improved genetic algorithm for mobile robot path planning in static environments[J]. *Expert Systems with Applications*, 2024, 249: 123762.
- [10] Li Y, Zhao J Y, Chen Z H, et al. A robot path planning method based on improved genetic algorithm and improved dynamic window approach[J]. *Sustainability*, 2023, 15(5): 4656.
- [11] Jain M, Saihjal V, Singh N, et al. An overview of variants and advancements of PSO algorithm[J]. *Applied Sciences*, 2022, 12(17): 8392.
- [12] 刘志华, 张冉, 郝梦男, 等. 基于改进 T 分布烟花-粒子群算法的 AUV 全局路径规划[J]. *电子学报*, 2024, 52(9): 3123-3134.  
(Liu Z H, Zhang R, Hao M N, et al. AUV global path planning based on improved T-distribution fireworks-particle swarm optimization algorithm[J]. *Acta Electronica Sinica*, 2024, 52(9): 3123-3134.)
- [13] Ghasemi M, Zare M, Trojovský P, et al. Optimization based on the smart behavior of plants with its engineering applications: Ivy algorithm[J]. *Knowledge-*

- Based Systems, 2024, 295: 111850.
- [14] 王合彬, 汪冠骅, 杨代虎, 等. 改进常春藤优化算法在天文干涉阵列设计中的应用[J]. 集成电路应用, 2024, 41(11): 78-79.  
(Wang H B, Wang G H, Yang D H, et al. Application of improved ivy optimization algorithm in the design of astronomical interference arrays[J]. Application of IC, 2024, 41(11): 78-79.)
- [15] 冯振辉, 肖人彬. 基于混合反馈机制的扩展蚁群算法[J]. 控制与决策, 2022, 37(12): 3160-3170.  
(Feng Z H, Xiao R B. Extended ant colony algorithm based on mixed feedback mechanism[J]. Control and Decision, 2022, 37(12): 3160-3170.)
- [16] 魏书鑫, 王群京, 李国丽, 等. 萤火虫算法结合遗传算法的移动机器人路径规划[J]. 制造业自动化, 2024, 46(10): 69-82.  
(Wei S X, Wang Q J, Li G L, et al. Firefly algorithm combined with genetic algorithm for mobile robot path planning[J]. Manufacturing Automation, 2024, 46(10): 69-82.)
- [17] 薛阳, 倪大斌, 卢秋红, 等. 基于 PGWO 算法的移动机器人路径规划[J]. 控制与决策, 2025, 40(4): 1395-1401.  
(Xue Y, Ni D B, Lu Q H, et al. Path planning of mobile robot based on PGWO algorithm[J]. Control and Decision, 2025, 40(4): 1395-1401.)

### 作者简介

沈波 (2001-), 男, 硕士生, 主要研究方向为机器人智能路径规划和导航, E-mail: [symbol23441@163.com](mailto:symbol23441@163.com);

杜鹏桢 (1982-), 男, 副教授, 博士, 主要研究方向为机器人技术、智能计算, E-mail: [dupengzhen@njust.edu.cn](mailto:dupengzhen@njust.edu.cn).