

doi: 10.7690/bgzdh.2023.04.016

## 改进蚁群融合 DWA 算法的移动机器人路径规划

王 倩, 李俊丽, 杨立炜, 夏国锋, 杜凌浩  
(昆明理工大学信息工程与自动化学院, 昆明 650504)

**摘要:** 为实现机器人在动态环境下的自主导航, 基于蚁群算法规划出全局最优路径的情况下采用 DWA 算法进行局部避障。根据距离障碍栅格的远近计算邻接栅格的初始信息素, 提出初始信息素不均匀分配原则; 对启发式函数进行自适应调整的改进, 提高算法的搜索速率; 利用狼群法则改进信息素更新方式, 对最优、最差和普通层蚂蚁进行分类更新, 提高算法的寻优能力; 使用二次路径优化的方法, 有效减少路径长度, 提高路径的平滑度; 以蚁群算法全局规划路径的关键点为目标点, 采用 DWA 算法进行局部路径规划。仿真结果表明: 改进后的融合算法能减少最优路径长度, 减少路径转弯次数且有效躲避障碍物。

**关键词:** 路径规划; 蚁群算法; DWA 算法; 路径优化

**中图分类号:** TP242 **文献标志码:** A

## Mobile Robot Path Planning Based on Improved Ant Colony and DWA Algorithm

Wang Qian, Li Junli, Yang Liwei, Xia Guofeng, Du Linghao

(School of Information Engineering and Automation, Kunming University of Technology, Kunming 650504, China)

**Abstract:** In order to realize the autonomous navigation of mobile robot in dynamic environment, DWA algorithm is used to avoid local obstacles based on the global optimal path planned by ant colony algorithm. The initial pheromone of the adjacent grids is calculated according to the distance from the obstacle grid, and the principle of uneven distribution of the initial pheromone is proposed; the heuristic function is improved by adaptive adjustment to improve the search rate of the algorithm; the wolf group rule is used to improve the pheromone update mode, and the optimal, worst and ordinary layer ants are classified and updated to improve the optimization ability of the algorithm; The method of secondary path optimization is used to effectively reduce the length of the path and improve the smoothness of the path; Taking the key points of the global planning path of the ant colony algorithm as the target points, the DWA algorithm is used for local path planning. The simulation results show that the improved fusion algorithm can reduce the length of the optimal path, reduce the number of turns and effectively avoid obstacles.

**Keywords:** path planning; ant colony algorithm; DWA algorithm; path optimization

### 0 引言

路径规划<sup>[1]</sup>研究在智能机器人研究领域具有重要意义, 其主要目的是从某一点出发, 在考虑多种因素的情况下, 计算出到达目标点的有效最优路径。目前, 常见的全局路径规划算法主要包括蚁群算法<sup>[2]</sup>、A\*算法<sup>[3]</sup>、遗传算法<sup>[4]</sup>等。随着研究深入, 涌现出大量的新仿生智能优化算法, 例如天牛须搜索算法<sup>[5]</sup>、蛙跳算法<sup>[6]</sup>、麻雀搜索算法<sup>[7]</sup>等, 而局部路径规划算法有 DWA 算法和人工势场法等。

蚁群算法是一种智能仿生算法, 具有正反馈、并行性好以及适应性强<sup>[8]</sup>等优点, 其缺点是易陷入局部最优, 还可能出现易陷于死锁、前期搜索盲目和搜索效率低等问题。目前, 许多学者分别从改进初始信息素分布规则、信息素更新规则等方面对算法进一步优化。

对于蚁群算法的改进, 文献[9]由 A\* 算法得到启发, 改进节点状态转移规则, 增加算法搜索到最优路径的概率, 并引用狼群法则, 增加局部最优信息素, 减少局部最差信息素, 避免算法陷入局部最优。文献[10]结合自适应状态转移策略和自适应信息素策略, 将所提自适应多态蚁群算法成功运用于智能轮椅的路径规划邻域。文献[11]引入了 A\* 算法的评价函数和弯曲抑制算子, 改进启发式信息, 增加了全局路径的平滑度, 加快了算法收敛速度。文献[12]根据栅格地图中的 8 个不同方向选择, 指导路径的初始信息素分布, 加快了搜索速度, 减少了搜索初期耗费的时间。

动态窗口法是一种常见的局部路径规划算法, 近年来, 有许多关于动态窗口法的研究。文献[13]融合 A\* 与动态窗口法, 加入路径评价子函数, 使机

收稿日期: 2022-12-21; 修回日期: 2023-01-20

基金项目: 国家自然科学基金(61163051)

作者简介: 王 倩(1998—), 女, 重庆人, 硕士, 从事移动机器人路径规划研究。E-mail: 2678742447@qq.com.

器人在进行局部路径规划中躲避障碍物后回到全局路径上, 实现实时最优的路径规划。文献[14]结合改进蚁群与动态窗口法进行全局动态路径规划, 实现了在规划的全局最优路径上有效躲避动态障碍物。

大量研究发现, 蚁群算法易出现收敛性差和易陷于局部最优问题, 笔者分别从蚁群算法改进和路径二次优化 2 方面对移动机器人规划路径优化。1) 利用障碍栅格的占比来计算各栅格初始信息素的分配, 障碍栅格越多的环境初始信息素越小, 反之, 初始信息素越大, 以此提高算法收敛速度; 改进启发函数, 降低启发式信息在后期对路径选择的影响; 引入狼群法则对信息素更新方式进行改进, 加大最优蚂蚁的影响, 减弱最差蚂蚁的影响。2) 对规划路径进行二次优化, 删除冗余点, 有效减少最优路径长度, 提高路径平滑度。此外, 针对蚁群算法无法躲避动态障碍物的问题, 采用结合动态窗口法的避障能力, 在全局最优路径上进行动态路径规划。

## 1 蚁群算法原理

蚁群优化是个体活动之间以信息素为媒介的交流, 开始时蚂蚁随机行走, 寻找到达目的地的路径, 一边寻路一边在经过的路径上释放信息素, 其他蚂蚁会跟随其足迹。信息素释放后会随时间挥发, 较长的路径比较短的路径具有更高的信息素挥发量, 最终蚂蚁会收敛到信息素浓度最高的最优路径<sup>[15]</sup>。

蚂蚁  $k$  在  $t$  时刻通过状态转移概率的计算选择  $i$  点到  $j$  点的行进方向。通过公式计算出  $i$  点邻接栅格的可选路径的转移概率  $P_{ij}^k(t)$ , 利用轮盘赌规则确定转移栅格  $j$  点的转移概率, 如式(1)所示:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{j \in \text{allowed}} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, j \in \text{allowed} \\ 0, \text{其他} \end{cases} \quad (1)$$

式中:  $\text{allowed}$  为蚂蚁下一步可选节点的集合;  $P_{ij}^k(t)$  为节点转移概率;  $\tau_{ij}^k(t)$  为路径上的信息素浓度;  $\eta_{ij}^k(t)$  为启发式函数。

$$\eta_{ij}(t) = 1/d_{ij} \quad (2)$$

式中  $d_{ij}$  为节点  $i$  和节点  $j$  之间的欧氏距离。

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3)$$

为防止过多的信息素对后续蚂蚁选择路径造成错误的引导, 每一代蚂蚁完成寻路之后, 路径上的

信息都会按照式(4)进行更新:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (4)$$

式中:  $\rho$  为信息素挥发系数;  $1-\rho$  为信息素残留因子;  $\Delta\tau_{ij}(t, t+1)$  为本次循环  $i, j$  间的信息素增量。

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k; \quad (5)$$

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, \text{若蚂蚁} k \text{经过的路径}(i, j); \\ 0, \text{其他} \end{cases} \quad (6)$$

式中:  $\Delta\tau_{ij}^k$  为蚂蚁  $k$  在走过路径释放的信息素;  $Q$  为信息素初始值;  $L_k$  为蚂蚁  $k$  走过路径总长度。

## 2 改进蚁群算法

### 2.1 改进初始信息素分配原则

传统蚁群算法中, 在初始阶段环境中的信息素分布为相同的值, 会导致蚂蚁在初始路径搜索时存在一定的盲目性, 从而影响算法的搜索效率, 增加算法搜索时间。得到文献[16]的启发, 提出一种改进初始信息素分布方法, 该方法根据该节点周围的障碍物多少来计算初始信息素的大小, 具体实现如下:

$$\tau_{ij}(0) = \gamma * f(j); \quad (7)$$

$$f(j) = 1 - C_U(\text{allowed}_j)/8 \quad (8)$$

式中:  $\gamma$  为增强系数, 可以根据实际情况取值;  $f(j)$  为避障系数;  $C_U$  为补集符号;  $U$  为当前栅格的邻接栅格集合。使用  $f(j)$  函数可以计算当前栅格的 8 个邻接栅格中自由栅格的占比, 当该栅格的邻域栅格中障碍物栅格的数量越少时, 该栅格的初始信息素越大; 反之, 初始信息素就越小。以此引导蚂蚁前进, 避免蚂蚁对障碍物过多的地形寻路, 加快算法的收敛速度。

如图 1 所示, 对  $15 \times 15$  地图的边缘初始信息素浓度设置较低是为避免蚂蚁往边缘寻路导致偏离目标点。其他栅格的初始信息素按照式(7)、(8)进行计算, 可以看到障碍栅格较少的地方初始信息素浓度高(图示颜色深), 障碍物较多的位置, 初始信息素浓度低(图示颜色浅)。

### 2.2 改进启发函数

蚁群算法中, 启发式信息只考虑待选节点与目标点之间的欧氏距离, 会导致蚂蚁在靠近目标点的区域, 受到较大吸引力, 然而在远离目标点的区域, 蚂蚁不能受到目标点的引导, 会导致前期蚂蚁寻路

的随机性；因此，为了提高搜索启发性，在启发函数中增加当前节点与下一节点间的欧氏距离。此外，为削弱启发式信息在后期对路径选择的影响，加快算法的收敛速度，改进后的启发函数如式(9)：

$$\eta_{ij}(t) = \frac{1}{d_{ij} + d_{jd}} * (1 + \frac{Nc_{max} - Nc}{Nc_{max}}) \quad (9)$$

式中： $Nc_{max}$  为最大迭代次数； $Nc$  为当前迭代次数。

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2			8.788.75					7.5	10	10	10	10	2						
2	6.256.256.257.5	6.256.255						7.5	10	10	10	10	2						
2	6.25	6.25	8.7510	8.758.758.7510				7.5	10	10	10	10	2						
2		5	8.758.758.758.7510					7.5	10	10	10	10	2						
2	8.25	7.5	10	10	7.5			7.5	10	10	10	8.757.5	2						
2	7.5	7.5	10	8.755				5	8.7510	8.756.25									
2	7.5		5	7.5	7.5				8.7510	8.75									
2	7.5					6.257.5	6.257.5	8.7510	8.757.5	6.252									
2	8.757.5	5				6.258.7510		10	10	10	8.757.5	7.5	2						
2	10	10	8.757.5			8.7510	8.758.758.757.5						2						
2	10	10	10	8.758.758.7510	7.5			7.5	6.25		5	2							
2	10	10	10	10	10	10	10	7.5		6.256.25		6.252							
2	10	10	10	10	10	10	10	8.757.5		7.5	7.5		2						
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

图 1 初始信息素分布

### 2.3 改进信息素更新算法

传统蚁群算法中，信息素更新对蚂蚁进行分类无差别处理，可能导致最差蚂蚁留下信息素对算法产生较大影响，最优蚂蚁的优势不能发挥出来；因此，本文中算法引入狼群法则，模拟“强者生存”的狼群更新规则和“由强到弱”的食物分配顺序，身体强壮的狼会更容易猎到食物也会优先获得食物，而病弱瘦小的狼随着狼群的发展难以生存下去，经过迭代，狼群的生存能力会逐渐提高。

笔者将狼群法则与信息素更新进行结合：找到局部最优蚂蚁生成的路径和局部最差蚂蚁生成的路径，分别对最优路径和最差路径进行信息素更新，加强局部最优解对后续循环的指导作用，而减弱局部最差解的误导作用，对于除了最优解和最差解之外的普通路径，则引入局部信息素挥发系数  $\lambda$ ，各个路径按照式(10)–(14)进行信息素更新：

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + (1 + b/w)\Delta\tau_{ij}(t+1); \quad (10)$$

$$\Delta\tau_{ij}(t+1) = \Delta\tau_{ij}^b(t+1) + \Delta\tau_{ij}^w(t+1) + \Delta\tau_{ij}^*(t+1); \quad (11)$$

$$\Delta\tau_{ij}^b(t+1) = \tau_{ij}(t) + Q/L + A(L_w/L_b); \quad (12)$$

$$\Delta\tau_{ij}^w(t+1) = \tau_{ij}(t) + Q/L - B(L_w/L_b); \quad (13)$$

$$\Delta\tau_{ij}^*(t+1) = (1 - \lambda)\tau_{ij}(t + C(Q/L)). \quad (14)$$

式中： $b$  为最优蚂蚁的数量，找到最优路径的蚂蚁定义为最优蚂蚁； $w$  为最差蚂蚁的数量，走过路径

为最差路径的蚂蚁定义为最差蚂蚁； $A$ 、 $B$ 、 $C$  分别为最优、最差和普通蚂蚁的信息素增强因子权重，可根据实际情况取值； $L_b$ 、 $L_w$  分别为最优路径长度和最差路径长度。

### 2.4 最优路径二次优化处理

在路径规划中，除路径长短以外还可以通过路径安全系数、能耗大小、路径平坦程度、路径平滑度等作为衡量路径好坏的标准；因此，在蚂蚁找到最优路径的基础上，为使机器人规划路径的转弯次数少，最优路径更短，笔者应用二次优化概念来优化路径，减少机器人行驶中的转弯节点数量。笔者在改进蚁群算法求出全局最优路径的基础上，对路径中的节点进行选择、判断和平滑连接。基本原则为：

1) 如果连接节点的连线之间无障碍物且连接后长度小于最短路径，则进行连接，并以这个节点为起始点继续和后面的节点进行连接。

2) 当连线之间存在障碍物时，则说明原本路径为最短安全路径，优化无效，继续判断后续其他节点，判断、连接完毕以后得到长度和转折点减少的路径。

如图 2 所示，实线轨迹为本文中改进算法规划的路径，虚线轨迹为路径去除多余转折点优化后的路径。通过对比可以看出二次优化的有效性，优化后的路径比优化前更短，转弯次数更少。

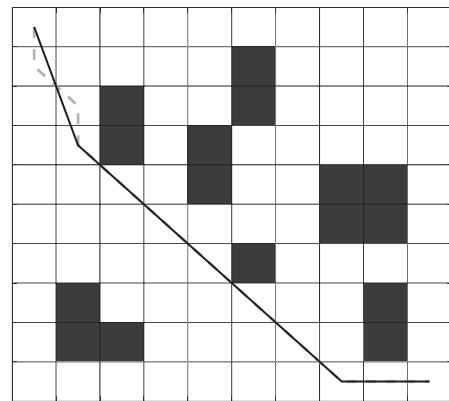


图 2 优化路径与原始路径对比

## 3 动态窗口法原理

动态窗口法具有良好的避障能力，可以通过在线实时采样当前线速度与角速度模拟下一段时间内的轨迹，通过评价函数找出最优线速度、角速度，驱使机器人按照最优采样速度的模拟轨迹进行移动。

### 3.1 机器人运动模型

笔者采用文献[17]提出的运动学模型:

$$x_{t+\Delta t}=x_t+v\Delta t\cos(\theta_t)-v_y\Delta t\sin(\theta_t); \quad (15)$$

$$y_{t+\Delta t}=y_t+v\Delta t\sin(\theta_t)+v_y\Delta t\cos(\theta_t); \quad (16)$$

$$\theta_{t+\Delta t}=\theta_t+\omega\Delta t. \quad (17)$$

式中:  $v, \omega$  分别为速度和角速度;  $(x_{t+\Delta t}, y_{t+\Delta t})$  为从当前位置  $(x_t, y_t)$  移动后的坐标;  $\theta_{t+\Delta t}$  为从当前角度  $\theta_t$  转动后的角度。

### 3.2 速度采样

1) 速度限制, 保证机器人以最大减速度停止都不会撞到最近的障碍物:

$$V_a = \left\{ (v, \omega) \mid v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{v}_b} \cap \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{\omega}_b} \right\}. \quad (18)$$

式中  $\text{dist}(v, \omega)$  函数用于评价机器人在当前模拟轨迹与最近静态障碍物的距离。

2) 加速度限制, 在当前状态下, 加减速变化都有上下限制, 因此下一时刻状态的加减速变化也存在限制:

$$V_d = \left\{ (v, \omega) \mid v \in [v_a - \dot{v} \cdot t, v_a + \dot{v} \cdot t] \cap \omega \in [\omega_a - \dot{\omega} \cdot t, \omega_a + \dot{\omega} \cdot t] \right\}. \quad (19)$$

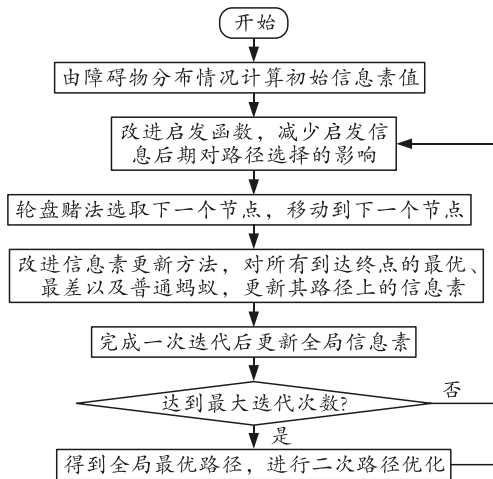


图 3 融合算法流程

3) 由于机器人本身的动力学限制, 角速度与线速度存在上下限标准:

$$V_m = \{v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\}. \quad (20)$$

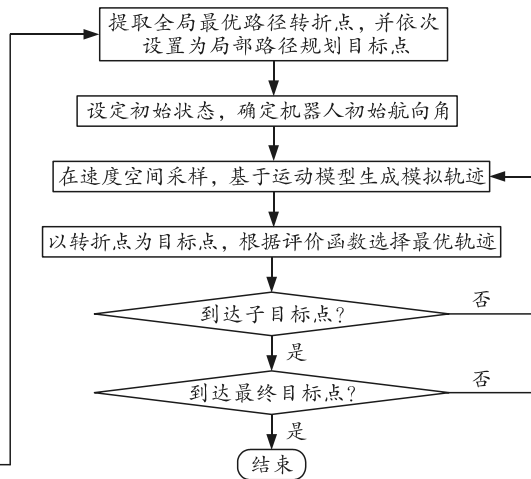
由上述 2 个限制条件的交集生成动态窗口, 根据评价函数, 朝着目标方向前进, 其评价函数为:

$$G(v, \omega) = \rho[\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega)]. \quad (21)$$

式中:  $\text{heading}(v, \omega)$  函数用于评价机器人当前模拟轨迹与目标之间的角度差;  $\text{velocity}(v, \omega)$  函数用于评价当前模拟轨迹的速度大小;  $\alpha, \beta, \gamma$  为权重系数,  $\rho$  为归一化参数。

## 4 融合算法的全局动态路径规划

改进蚁群算法可以进行全局路径规划但是无法避免环境中出现的未知障碍物, 动态窗口法具有良好的避障能力, 但是该算法在应用过程中无法估计全局最优, 从而导致机器人在操作过程中易陷于局部最优。笔者选择将蚁群算法与动态窗口法融合, 以改进蚁群算法二次路径优化后的路径转折点为动态窗口法的中间目标点进行局部路径规划, 以保证动态规划路径的全局最优, 同时能实时避障。融合算法的流程如图 3 所示。



## 5 仿真实验

本文中的实验参数初始值如表 1 所示。

表 1 实验参数设置

参数	传统蚁群算法	本文中算法
Nmax	100	100
m	50	50
a	1	1
β	15	13.7
Q	1	10
ρ	0.95	0.35
λ		0.05

算法运行环境为: Windows11 64 位; Matlab 2019a; 处理器 R5-5600H; 内存 16 GB。

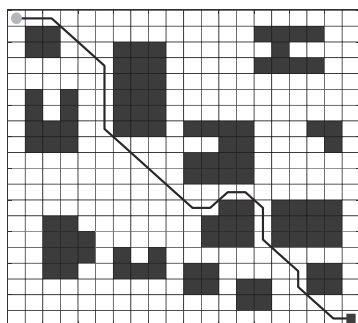
### 5.1 改进蚁群算法实验分析

为验证在常规环境中, 本文中算法对比传统蚁群算法的优越性, 在 20×20 的环境地图上进行对比实验。实验结果如表 2 所示, 2 种算法的路径规划结果及收敛曲线变化趋势如图 4—5 所示, 图 5(a) 中虚线路径为本文中改进算法规划路径, 实线路径

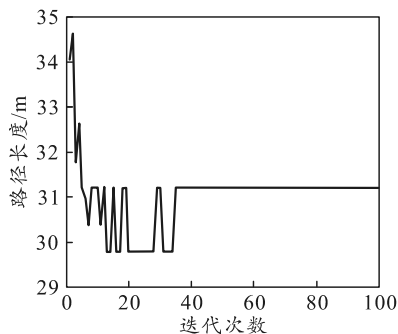
为改进算法加二次优化后的路径。

表 2 20×20 环境算法性能对比

最优路径的各项 指标	传统蚁群算法	本文中算法	
		二次优化前	二次优化后
路程距离/m	31.80	29.80	28.92
收敛迭代次数	35.00	8.00	8.00
转折点次数	12.00	10.00	8.00

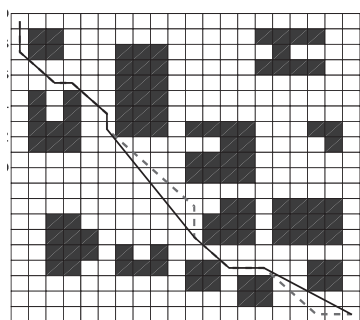


(a) 路径轨迹

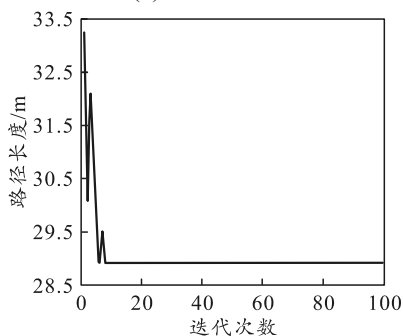


(b) 收敛速度

图 4 传统蚁群算法路径规划



(a) 路径轨迹



(b) 收敛速度

图 5 本文中算法路径规划

结果表明：传统蚁群算法的路径长度为 31.80，经历 35 次迭代后收敛于最优路径，转弯次数为 12，而本文中算法的路径长度较短，为 29.80，二次优化后的路径长度可达 28.92，说明二次优化路径有效减少了 2.95% 的路径长度；且本文中算法在运行 8 次后快速收敛于最优路径，转折点数由优化前的 10 次降为 8 次，说明本文中迭代收敛速度较优，最优路径长度以及转折点个数较优化前效果更好。

### 5.2 融合蚁群与动态窗口法实验分析

为验证算法的避障能力，设置 20×20 的动态环境，改进蚁群算法的参数参考表 1，动态窗口法中机器人最高速度为 1.5 m/s，最大旋转速度为 20 rad/s，最大加速度为 0.2 m/s<sup>2</sup>，最大角加速度为 50 rad/s<sup>2</sup>，速度分辨率为 0.02 m/s，转速分辨率为 1 rad/s<sup>2</sup>，设置评价函数权重  $\alpha=0.15$ ， $\beta=0.1$ ， $\gamma=0.3$ 。在规划出的全局最优路径上分别加入几个未知静态障碍物以及一个未知动态障碍物进行实验，实验结果如图 6—7 所示。图中虚线表示改进蚁群算法规划出的全局最优路径，实线表示动态路径规划的路线，较大灰色栅格无运动轨迹，为未知静态障碍物，较小灰色栅格表示未知动态障碍物，箭头为移动障碍物的运动方向。

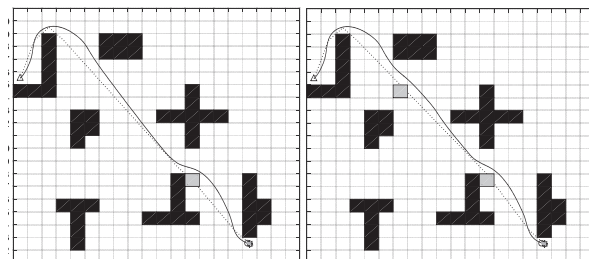


图 6 躲避未知静态障碍物

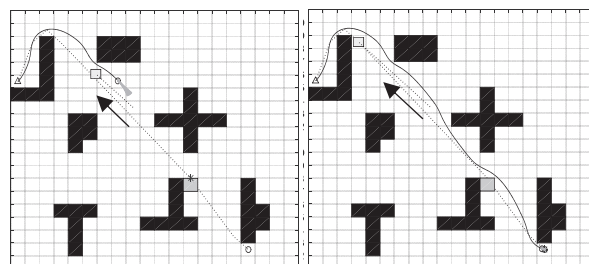


图 7 躲避未知动态障碍物

结果表明：图 6 由于全局信息的指导，机器人可以躲避随机加入的未知静态障碍物，躲避之后会回到全局最优路径上继续行走；图 7 中设置动态障碍物与机器人运动方向相反，两者在中途相遇，但机器人成功避开移动障碍物。说明了动态窗口法在局部避障中的有效性。

## 6 结束语

笔者提出改进蚁群的优化算法进行路径规划方法, 引入了初始信息素不均匀分配原则, 根据障碍栅格占据邻域 8 个栅格的占比对各个栅格分配不同的初始信息素, 降低蚁群的盲目搜索, 提高算法的搜索速率; 针对传统蚁群算法全局搜索能力弱的问题, 本文中的信息素更新策略使用了狼群法则的概念, 对最优、最差以及普通蚂蚁 3 类蚂蚁, 按 3 种方法更新信息素, 提高了算法的寻优能力; 还使用二次优化路径的方法, 有效减少了路径长度; 最后在全局信息的引导下, 引入动态窗口法进行局部路径规划, 成功躲避随机加入的未知静态障碍物与动态障碍物。通过实验对比分析, 验证了改进算法规划路径的有效性和可靠性。未来, 将在复杂环境下对动态窗口法的应用进一步探索。

### 参考文献:

- [1] 朱大奇, 颜明重. 移动机器人路径规划技术综述[J]. 控制与决策, 2010, 25(7): 961-967.
- [2] 柳长安, 鄢小虎, 刘春阳, 等. 基于改进蚁群算法的移动机器人动态路径规划方法[J]. 电子学报, 2011, 39(5): 1220-1224.
- [3] 李冲, 张安, 毕文豪. 基于方向约束的 A\*算法[J]. 控制与决策, 2017, 32(8): 1395-1402.
- [4] 刘二辉, 姚锡凡, 蓝宏宇, 等. 基于改进遗传算法的自动导引小车动态路径规划及其实现[J]. 计算机集成制造系统, 2018, 24(6): 1455-1467.
- [5] WU Q, LIN H, JIN Y, et al. A new fallback beetle antennae search algorithm for path planning of mobile robots with collision-free capability[J]. Soft Computing, 2020, 24(3): 2369-2380.
- [6] 徐晓晴, 朱庆保. 基于蛙跳算法的新型机器人路径规划算法[J]. 小型微型计算机系统, 2014, 35(7): 1631-1635.
- [7] LIU G, SHU C, LIANG Z, et al. A Modified Sparrow Search Algorithm with Application in 3d Route Planning for UAV[J]. Sensors, 2021, 21(4): 1224.
- [8] 马卫, 朱庆保. 求解函数优化问题的快速连续蚁群算法[J]. 电子学报, 2008(11): 2120-2124.
- [9] 朱颢东, 孙振, 吴迪, 等. 基于改进蚁群算法的移动机器人路径规划[J]. 重庆邮电大学学报(自然科学版), 2016, 28(6): 849-855.
- [10] JIAO Z, MA K, RONG Y, et al. A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs[J]. Journal of Computational Science, 2018, 25: 50-57.
- [11] DAI X, LONG S, ZHANG Z, et al. Mobile robot path planning based on ant colony algorithm with A\*heuristic method[J]. Frontiers in neurorobotics, 2019, 13: 15.
- [12] 李龙澍, 喻环. 改进蚁群算法在复杂环境中机器人路径规划上的应用[J]. 小型微型计算机系统, 2017, 38(9): 2067-2071.
- [13] 劳彩莲, 李鹏, 冯宇. 基于改进 A\*与 DWA 算法融合的温室机器人路径规划[J]. 农业机械学报, 2021, 52(1): 14-22.
- [14] 杨周, 刘海滨. 基于改进蚁群与动态窗口法的 AGV 动态路径规划[J/OL]. 计算机工程与应用: 1-11[2021-11-29]. <http://kns.cnki.net/kcms/detail/11.2127.tp.20211108.1558.014.html>.
- [15] LIU J, YANG J, LIU H, et al. An improved ant colony algorithm for robot path planning[J]. Soft Computing, 2017, 21(19): 5829-5839.
- [16] 李理, 李鸿, 单宁波. 多启发因素改进蚁群算法的路径规划[J]. 计算机工程与应用, 2019, 55(5): 219-225, 250.
- [17] SEDER M, PETROVIC I. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles[C]//Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007: 1986-1991.