

文章编号: 2095-2163(2024)01-0134-06

中图分类号: TP301.6

文献标志码: A

# 融合哈里斯鹰与蚁群算法的机器人路径规划

李鹏飞, 包长春

(内蒙古工业大学 航空学院, 呼和浩特 010051)

**摘要:** 针对传统蚁群算法在静态环境进行路径规划时存在初始信息素匮乏, 收敛速度慢, 搜索不全面的问题, 提出了一种融合哈里斯鹰与蚁群算法的路径规划算法。基于栅格法对机器人工作地图进行建模, 使用哈里斯鹰算法对蚁群算法中初始参数信息素因子和启发函数因子进行训练, 并利用经哈里斯鹰算法优化过的蚁群算法对路径进行搜索, 改进信息素更新公式并设定信息素浓度阈值, 增加了融合算法的全局搜索能力及收敛速度。仿真实验结果表明, 融合算法相较于其它算法在最终解精度、转弯次数等方面有一定的优势, 验证了该算法的有效性和实用性。

**关键词:** 路径规划; 蚁群优化算法; 哈里斯鹰优化算法; 信息素浓度; 参数优化

## Robot path planning based on harris hawks optimization and ant colony optimization

LI Pengfei, BAO Changchun

(Institute of Aviation, Inner Mongolia University of Technology, Hohhot 010051, China)

**Abstract:** Aiming at the problems of traditional ant colony optimization algorithms in path planning in a static environment, such as lack of initial pheromones, slow convergence speed, and incomplete search, a path planning algorithm based on Harris Hawks optimization and ant colony algorithm was proposed. Map modeling based on the grid method for robots work, using the Harris Hawks optimization for initial parameters pheromone in ant colony algorithm and heuristic function factor for training, and by using the Harris Hawks optimization algorithm optimized ant colony algorithm to search path, improve the pheromone updating formula and setting threshold, the pheromone concentration increased the fusion algorithm is global search ability and convergence speed. The simulation results show that the fusion algorithm has certain advantages compared with other algorithms in the final solution accuracy, the number of turns and so on, which verifies the effectiveness and practicability of the algorithm.

**Key words:** path planning; ant colony optimization algorithm; Harris Hawk optimization algorithm; pheromone concentration; parameter optimization

## 0 引言

随着互联网、大数据以及人工智能等相关技术的飞速发展, 移动机器人在物流仓储、军事医疗、生产制造等领域具有广阔的应用前景和发展空间<sup>[1]</sup>。路径规划问题作为移动机器人的关键技术之一, 其主要功能是在满足一定约束条件下, 尽可能找到一条从初始位置到目标位置的最短无碰撞路径<sup>[2-3]</sup>。国内外专家学者针对路径规划问题提出了 A\* 算法<sup>[4]</sup>、遗传算法<sup>[5]</sup>、人工势场法<sup>[6]</sup>、粒子群算法<sup>[7]</sup>及蚁群算法<sup>[8]</sup>等多种经典算法, 取得了良好的成果。

蚁群算法具有正反馈机制、较强的搜索能力以及易于实现等优点, 还可以通过结合其他智能算法

的方式来提高计算性能, 因此被广泛应用于求解各种路径规划问题。传统蚁群算法存在初始信息素匮乏, 收敛速度慢等问题, 很多学者对此进行了深入研究。张成等<sup>[9]</sup>针对蚁群算法收敛速度慢、求解路径质量差等方面进行了优化, 但对初始信息素不足的问题并未涉及。杨岱川等<sup>[10]</sup>使用概率地图算法对蚁群算法寻径路线进行优化的方法, 提高了融合算法的收敛速度, 但在较大规模地图进行寻径时优化效果并不明显。Huailin Zhao<sup>[11]</sup>将 Dijkstra 算法融合蚁群算法进行求解, 先用 Dijkstra 算法计算出一条初始路径, 然后用蚁群算法对初始路径进行优化得到最终解, 使得路径长度得到缩短。

由于传统的蚁群算法停留在根据经验进行比较

**基金项目:** 自治区直属高校基本科研业务费项目(JY20220355)。

**作者简介:** 李鹏飞(1996-), 男, 硕士研究生, 主要研究方向: 智能算法优化、路径规划。

**通讯作者:** 包长春(1980-), 男, 博士, 副教授, 主要研究方向: 动力学与控制、飞行器设计。Email: 1035934962@qq.com

**收稿日期:** 2023-01-05

和筛选参数的常规模式<sup>[12]</sup>, 这会影响算法的收敛速度和寻径效果。基于上述问题, 本文利用栅格法对机器人工作空间进行建模, 结合哈里斯鹰算法与蚁群算法的工作原理, 利用哈里斯鹰算法对蚁群算法中的初始参数  $\alpha, \beta$  进行训练, 将得到的最优组合代入蚁群算法, 基于栅格环境求解出最优路径; 同时对信息素挥发系数  $\rho$  的选取方式进行优化并设置信息素浓度阈值, 达到增强 HHO-ACO (Harris Hawks Optimization-Ant Colony Optimization) 算法全局搜索能力的目的。

### 1 环境建模

在进行路径规划之前, 将机器人作为质点进行处理, 使用栅格法<sup>[13]</sup>对机器人工作地图进行建模。栅格法由二进制的 0、1 作为栅格取值, 地图中的白色自由栅格用 0 来表示, 机器人可由此通过; 黑色障碍栅格用 1 来表示, 机器人需要绕行通过。将地图划分为  $N \times M$  个正方形栅格, 从左到右、从下到上将栅格依次编号为 1、2、 $\dots$ 、 $k$ , 实现每个序号与栅格中心坐标的一一对应, 栅格地图如图 1 所示。地图左下角第一个栅格的序号为 1, 则栅格 1 中心点的坐标为 (0.5, 0.5), 同理栅格 2 的中心点坐标为 (1.5, 0.5), 类似的, 得到栅格中心点坐标与序号的计算公式如下:

$$\begin{cases} x_i = \begin{cases} \left[ \frac{\text{mod}(k, M) - 1}{2} \right], & \text{mod}(k, M) \neq 0 \\ M - \frac{1}{2}, & \text{mod}(k, M) = 0 \end{cases} \\ y_i = \left[ \frac{k}{N} \right] - \frac{1}{2} \end{cases} \quad (1)$$

式中:  $k$  为栅格序号,  $N$  为栅格地图行数,  $M$  为栅格地图列数,  $\text{mod}$  表示取  $k/M$  的余数。

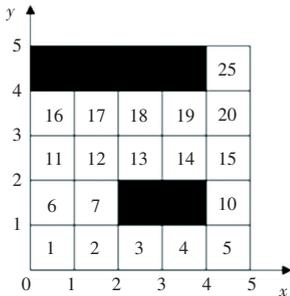


图 1 栅格地图  
Fig. 1 Raster map

## 2 蚁群算法及其改进

### 2.1 基本蚁群算法

蚁群算法(ACO)<sup>[14]</sup>是一种基于自然界中蚂蚁

觅食行为得到的群智能优化算法。算法主要思想是若干只蚂蚁在觅食经过的路径上留下信息素, 后面的蚂蚁将会大概率选择信息素浓度高的路径进行爬行, 不断迭代最终形成最优路径, 以此达到优化路径的目的。蚁群算法中的转移概率表达式用于引导蚂蚁从当前位置选择下一步位置, 算法中蚂蚁个数用  $k = 1, 2, \dots, m$  表示, 蚂蚁  $k$  在  $t$  时刻从当前  $i$  节点移动到下一  $j$  节点的可能性  $p_{ij}^k$  由如下的状态转移概率公式得到:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}(t)]^\beta}{\sum_{a \in allow_k} [\tau_{ia}(t)]^\alpha * [\eta_{ia}(t)]^\beta}, & j \in allow_k \\ 0, & \text{其他} \end{cases} \quad (2)$$

式中:  $\alpha$  和  $\beta$  分别表示蚂蚁在寻找道路时信息素和启发信息的重要程度,  $allow_k$  表示蚂蚁在栅格  $k$  时可行路径的集合,  $\tau_{ij}(t)$  为信息素的浓度,  $\eta_{ij}(t)$  为启发信息影响因子, 计算公式为

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \quad (3)$$

其中,  $d_{ij}$  为节点  $i$  与节点  $j$  之间的欧几里得距离, 公式为

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4)$$

蚂蚁移动时会更新路径上遗留的信息素, 方便下一代蚂蚁趋向于走全局最优路径, 其信息素更新公式如下:

$$\tau_{ij}(t+1) = \tau_{ij}(t) * (1 - \rho) + \Delta\tau_{ij}(t) \quad (5)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (6)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{蚂蚁 } k \text{ 经过边 } (i, j) \\ 0 & \text{其他} \end{cases} \quad (7)$$

其中,  $\rho$  是信息素挥发系数;  $\Delta\tau_{ij}(t)$  是信息素增量;  $Q$  是信息素强度;  $L_k$  是蚂蚁  $k$  到达目标点走过路径总长度。

### 2.2 蚁群算法改进

#### 2.2.1 改进信息素更新方式

蚁群算法实现过程中, 最重要的一点就是对信息素进行更新, 这主要是模拟实际环境中信息素会随着时间而不断的挥发和积累。在基本蚁群算法的信息素更新公式中, 信息素挥发系数  $\rho$  通常设定为常量, 这将会导致算法在前期进行全局搜索和后期实现快速收敛存在时间分配的问题。因此, 需要引入新的信息素挥发更新公式进行改进。改进后的公式如下:

$$\tau_{ij}(t+1) = \tau_{ij}(t) * [1 - \rho(t+1)] + \Delta\tau_{ij}(t+1) \quad (8)$$

$$\rho(t+1) = \begin{cases} \frac{D_{\max} - D}{D_{\max}} * \frac{1}{e^{1.1-\rho(t)}}, & \rho \geq \rho_{\min} \\ \rho_{\min}, & \text{其他} \end{cases} \quad (9)$$

其中,  $\rho_{\min}$  是  $\rho$  的最小值;  $D$  为当前的迭代次数;  $D_{\max}$  是最大迭代次数。信息素挥发系数  $\rho$  随着迭代次数的增加而减小, 当减到最小值后就变为固定值  $\rho_{\min}$ 。

引入新的更新策略, 不仅可以保证算法前期的全局搜索能力, 且同时提高算法的收敛速度。

### 2.2.2 信息素限制策略

为了避免算法过早达到局部收敛, 形成算法停滞的情况。通过设定信息素浓度阈值来进行改进, 在信息素更新阶段时施加最大范围的控制力, 在信息素挥发阶段时施加最小范围的控制力。在上下限范围内, 信息素浓度限制公式如下:

$$\tau_{ij}(t) = \begin{cases} \tau_{\max}, & \tau_{ij}(t) > \tau_{\max} \\ \tau_{ij}(t), & \tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max} \\ \tau_{\min}, & \tau_{ij}(t) < \tau_{\min} \end{cases} \quad (10)$$

对信息素浓度阈值进行设定, 可以防止某条路径上的信息素浓度过高或者过低, 增加算法路径选择的多样性。

## 3 哈里斯鹰算法

哈里斯鹰算法<sup>[15]</sup>是2019年Heidari等人模拟北美洲的哈里斯鹰群在搜寻、追捕猎物过程中形式多样的攻击策略而形成的一种用于求解各种优化问题的无梯度优化算法, 该算法的主要特点是收敛速度较快、所需参数较少、全局搜索能力较强。算法主要包括搜索阶段和开发阶段, 其构建过程如下。

搜索阶段的主要公式为:

$$X(t+1) =$$

$$\begin{cases} X_{rand}(t) - r_1 | X_{rand}(t) - 2r_2X(t) |, & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)), & q < 0.5 \end{cases} \quad (11)$$

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (12)$$

式中:  $X_{rand}$  为当前种群中随机选择的个体,  $X_{rabbit}$  为当前最优个体,  $X_m$  为种群个体的平均位置,  $r_1, r_2, r_3, r_4$  均为  $0 \sim 1$  之间的随机数,  $UB, LB$  分别为种群的上下边界,  $N$  为种群数量。

探索阶段到开发阶段的转换公式为

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \quad (13)$$

其中,  $E$  代表控制算法全局搜索和局部搜索的逃逸能量因子;  $E_0$  是  $-1 \sim 1$  的随机数;  $t$  为当前迭代次数;  $T$  为最大迭代次数。

开发阶段软包围公式有:

$$X(t+1) = \Delta X(t) - E | JX_{rabbit}(t) - X(t) | \quad (14)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (15)$$

$$J = 2(1 - r_5) \quad (16)$$

开发阶段的硬包围公式为

$$X(t+1) = X_{rabbit}(t) - E | \Delta X(t) | \quad (17)$$

开发阶段渐进软包围公式有:

$$Y = X_{rabbit}(t) - E | JX_{rabbit}(t) - X(t) | \quad (18)$$

$$Z = Y + S \times LF(D) \quad (19)$$

$$X(t+1) = \begin{cases} Y, & F(Y) < F(X(t)) \\ Z, & F(Z) < F(X(t)) \end{cases} \quad (20)$$

开发阶段渐进硬包围公式为:

$$Y = X_{rabbit}(t) - E | JX_{rabbit}(t) - X_m(t) | \quad (21)$$

$$Z = Y + S \times LF(D) \# \quad (22)$$

$$X(t+1) = \begin{cases} Y, & F(Y) < F(X(t)) \\ Z, & F(Z) < F(X(t)) \end{cases} \quad (23)$$

开发阶段中,  $\Delta X$  为最优个体和当前个体的差值;  $r_5$  是  $0 \sim 1$  之间的随机数;  $J$  为  $X_{rabbit}$  逃跑过程中的跳跃距离;  $D$  为求解问题的维度;  $S$  为与  $D$  相同维度的随机向量;  $LF$  的全称为 Levy 飞行函数。

## 4 哈里斯鹰-蚁群融合算法设计

从概率转移公式可以看出, 信息素因子  $\alpha$  表示蚂蚁寻找道路时信息素的重要程度, 蚂蚁选择之前路径的可能性随  $\alpha$  的增加而变大, 这会减小算法随机性的可能。启发函数因子  $\beta$  表示蚂蚁寻找道路时启发式信息的重要程度, 算法的收敛速度会因为  $\beta$  过大而加快, 这会增加算法陷入局部最优的可能。  $\alpha$  和  $\beta$  作为蚁群算法在实现过程中重要的两个参数, 在进行参数选择时主要依靠人们的工作经验以及重复调试, 而参数选取的是否合适最终又会影响算法的整体性能。

因此, 利用哈里斯鹰算法对蚁群算法参数  $\alpha$  和  $\beta$  进行训练, 解决初始参数选取不准确的问题, 以此提高算法的精度。具体做法是: 在栅格地图内先利用哈里斯鹰算法训练得到蚁群算法所需初始参数  $\alpha$  和  $\beta$ , 再用改进蚁群算法求解最优路径。HHO-ACO 算法流程如图 2 所示。

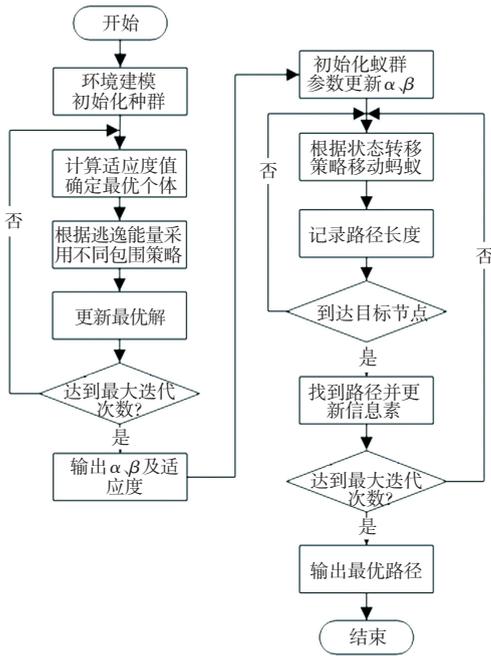


图 2 HHO-ACO 算法流程

Fig. 2 HHO-ACO algorithm flow chart

融合算法步骤为如下:

**Step 1** 建立静态栅格地图, 哈里斯鹰种群初始化, 计算初始适应度, 保留当前最优个体;

**Step 2** 更新猎物逃逸能量, 根据逃逸能量和生成的随机数, 执行搜索或开发行为中对应的位置更新策略;

**Step 3** 将位置更新后的个体适应度值与猎物的适应度值进行比较, 将数值更小的个体所在位置设定为后续猎物位置; 重复实验, 直到满足最大迭代次数时算法结束, 并输出此时的  $\alpha$  和  $\beta$  取值, 否则返回 Step 2;

**Step 4** 初始化蚁群算法中的参数, 将上一步得到的数值赋值给信息素因子  $\alpha$  和启发函数因子  $\beta$ ;

**Step 5** 使用式(2)移动蚂蚁, 计算路径长度, 到达目标节点并更新路径; 如不满足则返回 Step 4;

**Step 6** 使用式(8)-式(10)中的更新方法进行信息素的更新; 重复实验直到满足最大迭代次数时输出计算得到的最优路径, 否则返回 Step 4.

### 5 实验结果分析

为验证本文算法的实际效果, 在构建的不同复杂程度 ( $20 \times 20, 50 \times 50$ ) 二维栅格地图中进行仿真对比实验。实验所用操作环境为 windows10, 仿真软件为 MATLAB R2020b。实验前对算法的初始参数进行设定, 哈里斯鹰算法中的种群规模设为 30, 最

大迭代次数设为 3。蚁群算法中的种群规模设为 50, 最大迭代次数设为 200,  $Q = 1, \rho = 0.1, \alpha$  和  $\beta$  由哈里斯鹰算法训练得到, 训练结果见表 1。

表 1 参数范围及训练结果

Table 1 Parameter range and training results		
参数	范围	训练结果
$\alpha$	1-10	3.101 4
$\beta$	1-10	7.664 8

#### (1) 实验一

在  $20 \times 20$  的栅格地图中进行实验, 起点坐标为 (1, 1), 终点坐标为 (20, 20)。将蚁群 (ACO) 算法、哈里斯鹰 (HHO) 算法、文献 [16] 中的遗传蚁群 (GA-ACO) 算法与本文所提哈里斯鹰-蚁群 (HHO-ACO) 算法进行对比分析。路径规划对比结果及最优路径收敛曲线分别如图 3、图 4 所示, 4 种算法在路径长度、转弯次数、迭代次数、搜索时间的对比结果见表 2。

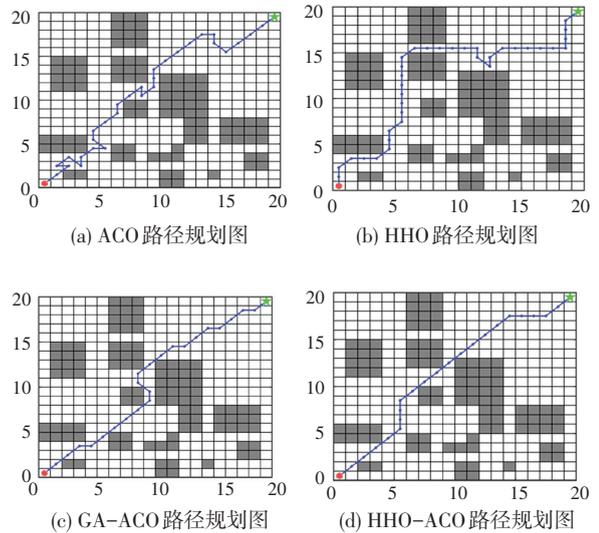


图 3 路径规划实验对比

Fig. 3 Comparison diagram of path planning experiment

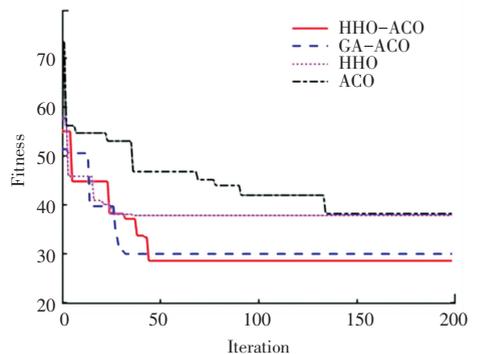


图 4 最优路径收敛曲线

Fig. 4 Optimal path convergence curve

表2 计算结果对比

Table 2 Comparison of calculation results

算法	路径长度	转弯次数	迭代次数	搜索时间
ACO	38.284 3	19	134	3.652 149
HHO	37.899 5	15	24	0.803 410
GA-ACO	30.041 6	12	32	2.437 991
HHO-ACO	28.627 4	4	44	11.617 646

由表2得出,所选4种算法都能顺利找到可行路径。相比而言,ACO算法找到的路径长度最长,转弯次数以及迭代次数最多,在实际应用中会消耗更多的燃料。HHO算法得到的路径长度、转弯次数和ACO算法差别不大,但在迭代次数和搜索时间上分别提高了82%和78%,可见哈里斯鹰算法在收敛速度和收敛时间上明显优于蚁群算法。

相比原始ACO算法,HHO-ACO算法搜索的路径长度从38.284 3减少到28.627 4(减少25.2%);转弯次数由19次减少到4次(减少78.9%);迭代次数由134次减少到44次(减少67.1%)。同时,HHO-ACO算法相比HHO算法在路径长度上减少24.5%,转弯次数上减少73.3%,可以看出HHO-ACO算法在路径长度和转弯次数上明显优于ACO和HHO算法。HHO-ACO算法与文献[16]中的GA-ACO算法相比,路径长度从30.041 6减小到28.627 4(减小4.7%),转弯次数减小66.6%,可以看出HHO-ACO算法在路程减短的同时大幅减少转弯次数。尽管HHO-ACO算法在搜索时间上不如其余3种算法,但在路径长度以及转弯次数上有显著提高。综合来看,本文算法在处理简单地图上具有优势。

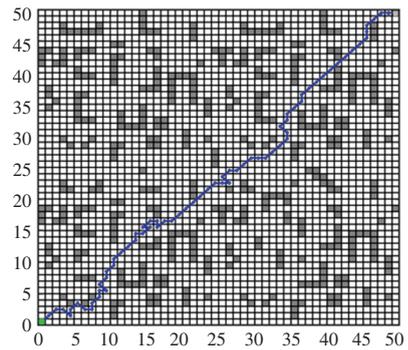
## (2) 实验二

为进一步验证本文算法在较大规模地图路径规划的有效性,在 $50 \times 50$ 的复杂栅格地图上进行测试,并与文献[16]中遗传蚁群(GA-ACO)算法进行了对比实验,路径规划实验结果如图5所示。GA-ACO与HHO-ACO在最优路径长度、平均路径长度、转弯次数、平均迭代次数的对比结果见表3。

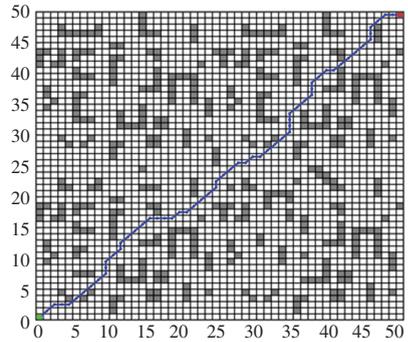
表3 GA-ACO、HHO-ACO对比结果

Table 3 GA-ACO、HHO-ACO comparison result

算法	最优路径长度	平均路径长度	转弯次数	平均迭代次数
GA-ACO	80.568 5	82.588 6	45次	142
HHO-ACO	75.397 1	76.997 1	25次	100.75



(a) GA-ACO 路径规划图



(b) HHO-ACO 路径规划图

图5 路径规划实验对比图

Fig. 5 Comparison diagram of path planning experiment

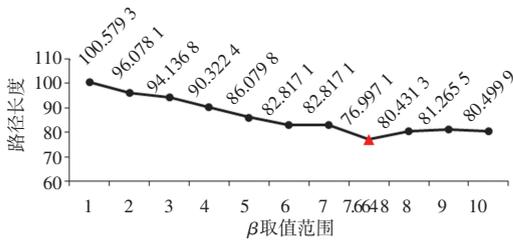
由表3得出,对于最优路径长度,HHO-ACO相对于GA-ACO算法减少了6.42%,说明本文算法拥有更好的全局寻优能力,找到的路径更短。对于平均路径长度,HHO-ACO相对于GA-ACO算法减少了6.77%,本文算法在稳定性以及搜索效率上更高。对于转弯次数,HHO-ACO相对于GA-ACO算法减少了44.44%,本文算法的路径平滑度更优,减少在实际应用中对机器人的操作和燃料消耗。对于平均迭代次数,HHO-ACO相对于GA-ACO算法减少了29%,本文算法能够更快速的趋于稳定,提高了算法的稳定性。

## (3) 实验三

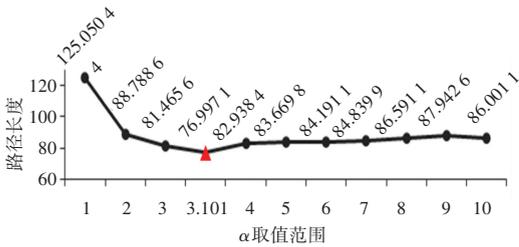
为进一步验证信息素因子 $\alpha$ 和启发函数因子 $\beta$ 的选取对于算法性能的影响,运用本文所提算法在 $50 \times 50$ 栅格地图环境下,采用控制变量法进行实验。即分别固定 $\alpha$ 和 $\beta$ 的取值,记录另一个参数在上下限范围内变化的路径长度值,路径长度随 $\alpha, \beta$ 变化结果如图6所示。

由图6(a)可知,当 $\alpha = 3.101 4$ , $\beta$ 取值在1~10范围内变化时,路径长度呈现出先减后增的趋势;在 $\beta = 7.664 8$ 时取得极小值76.997 1,此时路径长度最短。同理,在图6(b)中,当 $\beta$ 固定为7.664 8, $\alpha$ 取值变化时,路径长度在 $\alpha = 3.101 4$ 时取得最小值

76.997 1。综上,当  $\alpha = 3.101 4$ ,  $\beta = 7.664 8$  时,改进算法性能达到最佳,验证了本文算法的有效性。



(a) 路径长度随  $\beta$  变化折线图



(b) 路径长度随  $\alpha$  变化折线图

图 6 路径长度变化折线图

Fig. 6 Line graph of path length varying

### 6 结束语

针对传统蚁群算法实现过程存在的不足,本文提出了一种融合哈里斯鹰与蚁群算法的路径规划算法。采用哈里斯鹰算法优化蚁群算法中的初始参数信息素因子和启发函数因子,提升了算法的精度。同时改进信息素更新公式并设定信息素浓度阈值,提高了算法的收敛性。仿真结果表明,本文算法较原始 ACO 算法和 GA-ACO 算法在转弯次数减少 40% 以上,在大范围地图环境下效果更为明显,验证了本文所提算法的有效性。

另外,本文未能有效降低算法的空间复杂度,使得算法的运行时间相较对比算法更长,这也是后期需要解决的问题。

(上接第 133 页)

### 参考文献

[1] CHEN Lulu, WEI Hong, FERRYMAN J. A survey of human motion analysis using depth imagery [J]. Pattern Recognition Letters, 2013,34(15): 1996.

[2] ESPINO J C, STANCIULESCU B, FORIN P. Rail and turnout detection using gradient information and template matching [C]// Proceedings of 2013 IEEE International Conference on Intelligent Rail Transportation Proceedings. IEEE, 2013: 233-238.

[3] WENPENG M, HAYASHI H. Research of intelligent search engine based on computer vision [C]//Proceedings of 2013 IEEE/ACIS 12<sup>th</sup> International Conference on Computer and

### 参考文献

[1] 林韩熙,向丹,欧阳剑,等. 移动机器人路径规划算法的研究综述[J]. 计算机工程与应用,2021,57(18):38-48.

[2] 王梓强,胡晓光,李晓筱,等. 移动机器人全局路径规划算法综述[J]. 计算机科学,2021,48(10):19-29.

[3] 张松灿,普杰信,司彦娜,等. 蚁群算法在移动机器人路径规划中的应用综述[J]. 计算机工程与应用,2020,56(8):10-19.

[4] 姜媛媛,张阳阳. 改进 8 邻域节点搜索策略 A\* 算法的路径规划[J]. 电子测量与仪器学报,2022,36(5):234-241.

[5] 李培英. 基于改进遗传算法的移动机器人路径规划[J]. 国外电子测量技术,2022,41(6):38-44.

[6] 孙鹏耀,黄炎焱,潘尧. 基于改进势场法的移动机器人路径规划[J]. 兵工学报,2020,41(10):2106-2121.

[7] 文展,唐康健,李文藻. 一种改进粒子群优化算法在车辆路径问题的应用研究[J]. 重庆邮电大学学报(自然科学版),2020,32(5):891-897.

[8] 毛文平,李帅永,谢现乐,等. 基于自适应机制改进蚁群算法的移动机器人全局路径规划[J/OL]. 控制与决策:1-9[2022-10-11].

[9] 张成,凌有铸,陈孟元. 改进蚁群算法求解移动机器人路径规划[J]. 电子测量与仪器学报,2016,30(11):1758-1764.

[10] 杨岱川,文成林. 基于蚁群和改进 PRM 算法的多目标点路径规划[J]. 杭州电子科技大学学报(自然科学版),2017,37(3):63-67.

[11] ZHAO H, NIE Z, ZHOU F, et al. A compound path planning algorithm for mobile robots [C]//Proceedings of 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA). IEEE, 2021: 1-5.

[12] 雷超帆,赵华东,江南. 融合粒子群与蚁群算法的机器人路径规划[J]. 重庆理工大学学报(自然科学),2020,34(1):235-241.

[13] 瞿新豪,丁云飞,谢亚琴. 基于栅格地图改进的智能路径选择算法[J]. 电子测量技术,2022,45(5):86-93.

[14] COLORNI A, DORIGO M, MANIEZZO V. Distributed optimization by ant colonies [C]// Proc of the First European Conference on Artificial Life the MIT Press, 1991: 134-144.

[15] HEIDARI A A, SEYEDALI M, HOSSAM F, et al. Harris hawks optimization: algorithm and applications [J]. Future Generation Computer Systems, 2019, 97(12): 849-872.

[16] 何娟,涂中英,牛玉刚. 一种遗传蚁群算法的机器人路径规划方法[J]. 计算机仿真,2010,27(3):170-174.

Information Science (ICIS). IEEE,2013:95-100.

[4] PRASUHN L, OYAMADA Y, MOCHIZUKI Y, et al. A HOG-based hand gesture recognition system on a mobile device [C]// Proceedings of 2014 IEEE International Conference on Image Processing. Paris:IEEE, 2014:3974.

[5] SANDLER M, HOWARD A, ZHU Menglong, et al. MobileNetV2: inverted residuals and linear bottlenecks [C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018:4512.

[6] 赵大兴. 高精度视觉测量系统中检测方法的研究[J]. 制造业自动化,2014,36(5):25-27.

[7] 杨丹,赵海滨,龙哲. 图像处理实例详解[M]. 北京:清华大学出版,2013:321.