

# 基于改进天牛群算法和三次样条插值的路径规划<sup>\*</sup>

欧阳鹏 尹先验 叶景贞

(江西理工大学电气工程与自动化学院 赣州 341000)

**摘要:**为更好地解决移动机器人路径规划问题,对天牛群算法的性能进行改进,并拓展其应用领域,提出了一种基于自适应精英变异的改进天牛群算法(AEM-BSO)。首先运用佳点集对天牛种群进行初始化操作,使种群分布更加均匀,降低陷入局部最优解的风险;其次使用非线性递减惯性权重策略,以提升算法的初期探索能力和收敛速度;然后引入精英变异策略,在每次迭代时对精英个体进行变异以产生新的、可能更优的候选解,避免早熟现象和陷入局部最优解。将AEM-BSO应用于求解移动机器人全局路径,借助三次样条插值方法,对规划得出的全局路径节点进行平滑化处理,使路径更贴合实际运动需求。最后在10个测试函数上和不同环境地图上评估AEM-BSO的有效性。实验结果表明,AEM-BSO在不同测试函数中具有较好的寻优精度和稳定性能,在机器人路径规划中路径长度较原始BSO算法、粒子群算法和蝙蝠算法分别减少了0.24%、18.12%与8.41%,标准差分别减少了25.8%、96.73%、14.13%,表明了AEM-BSO算法的有效性。

**关键词:**佳点集初始化;自适应精英变异;天牛群算法;三次样条插值;路径规划

中图分类号: TP24; TN96 文献标识码: A 国家标准学科分类代码: 510.80

## Path planning based on improved beetle swarm optimization algorithm and cubic spline interpolation

Ouyang Peng Kuang Xianyan Ye Jinzhen

(School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341000, China)

**Abstract:** To address the optimization requirements in mobile robot path planning and enhance the performance limitations of the beetle swarm optimization algorithm regarding convergence precision and application scope, this paper introduces an adaptive elite mutation-based beetle swarm optimization (AEM-BSO) algorithm. The methodological innovations manifest in three principal aspects. Firstly, the implementation of good point set initialization ensures uniform population distribution, effectively mitigating the risk of local optima entrapment. Subsequently, a non-linearly decreasing inertia weight strategy enhances global exploration capabilities during initial iterations while accelerating convergence rates in later stages. Furthermore, incorporation of elite mutation mechanisms that strategically perturb high-performing individuals during iterative processes to prevent premature convergence. For practical implementation in mobile robot navigation, cubic spline interpolation optimizes waypoint connections in generated paths, ensuring kinematic feasibility and smooth trajectory formation. Comprehensive validation across 10 benchmark functions and diverse environmental maps demonstrates the algorithm's superior optimization precision and robust stability. Experimental comparisons reveal that AEM-BSO achieves respective path length reductions of 0.24%, 18.12%, and 8.41% compared to primitive BSO, PSO and BA, accompanied by significant standard deviation decreases of 25.8%, 96.73%, and 14.13%. These quantitative improvements substantiate the proposed algorithm's effectiveness in balancing exploration-exploitation trade-offs and enhancing solution quality for complex path planning tasks.

**Keywords:** good point set initialized; adaptive elite mutation; beetle swarm optimization algorithm; cubic spline interpolation; path planning

## 0 引言

移动机器人当前的应用较为广泛,在工业生产、服务业、运输、仓储等领域都能见到其身影。路径规划问题一直都是移动机器人研究的核心。路径规划是指在具有障碍物的环境中根据某些性能指标,搜索出一条能够从起始点到目标点的无碰撞最优或近似最优路径。移动机器人的路径规划算法有 A<sup>\*</sup> 算法<sup>[1-2]</sup>、快速探索随机数(rapidly-exploring random trees, RRT)算法<sup>[3-4]</sup>、Dijkstra 算法<sup>[5]</sup>、粒子群算法 (particle swarm optimization, PSO)<sup>[6-9]</sup>、蝙蝠算法 (bat algorithm, BA)<sup>[10]</sup>、遗传算法(genetic algorithm, GA)<sup>[11]</sup>、蚁群算法 (ant colony optimization, ACO)<sup>[12-13]</sup>、天牛须算法 (beetle antennae search, BAS)<sup>[14-15]</sup>、深度学习<sup>[16]</sup>等。

天牛群算法 (beetle swarm optimization, BSO) 是一种智能优化算法,在群体智能优化算法中有较强的竞争性,常应用于连续函数优化问题,目前天牛群算法在路径规划领域应用较少。沈显庆等<sup>[17]</sup>将环境建模中的二维栅格法拓扑到三维,使用三维栅格法进行环境建模,将粒子群中的粒子替换成天牛提出天牛群算法并将其用于求解移动机器人的三维路径规划,该方法采用的是传统的天牛群算法,主要是面对的三维路径规划问题。吕昱呈等<sup>[18]</sup>提出一种融入变异交叉的改进天牛群算法,该方法主要是解决 TSP 问题。王蕊等<sup>[19]</sup>引入改进的 Tent 混沌序列生成初始解,加入高斯变异策略对天牛群算法进行改进,提高了驱鸟车的驱鸟效率。尽管现有的 BSO 算法在局部精细搜索能力上已有所改进,但其性能仍不够理想。与 PSO 算法和 BAS 算法相比,BSO 算法在收敛速度和寻优精度等关键性能指标上虽有一定程度的提升,但仍较大的优化空间。因此,本文对天牛群算法进行了改进,提出了一种自适应精英变异的改进天牛群优化算法 (adaptive elite mutation of improved beetle swarm optimization algorithm, AEM-BSO)。改进措施包括采用佳点集初始化天牛种群,调整算法的惯性权重变化策略,并在迭代过程中引入精英变异策略,保留优秀个体,并在后续迭代中对其进行微调,以产生新的、可能更优的候选解。这种策略有助于维持算法的多样性并避免过拟合到局部最优解。将改进后的天牛群算法应用于移动机器人的路径规划问题时,首先利用该算法寻找一条较优的路径,然后通过三次样条插值平滑路径,以消除路径中的锐角和突变,确保移动机器人运动的连续性和安全性。通过仿真对比实验,验证改进算法的可行性和有效性。

## 1 天牛群算法

天牛群算法是粒子群算法和天牛须搜索算法的结

合,将每个粒子都当成天牛个体进行搜索。天牛群搜索算法的收敛速度较快、收敛精度较高,其算法原理如下。

设  $f(x)$  为适应度函数,天牛群中的全部天牛个体在每次迭代时均对其左侧和右侧的适应度函数值进行比较,取两者的更优方向更新天牛群的位置。BSO 生成的更新率如式(1)所示。

$$vb_i = -\delta^t \cdot b \cdot sign(f(x_r) - f(x_l)) \quad (1)$$

式中: $\delta^t$  为  $t$  时刻天牛两须间距;  $b$  为标准化方向向量;  $x_l$  为左须位置;  $x_r$  为右须位置。

天牛群中天牛个体的速度和位置更新方法:

$$\begin{aligned} v_i^{t+1} &= \omega \cdot v_i^t + c_1 \cdot rand \cdot (Pbest_i^t - x_i^t) + c_2 \cdot rand \cdot \\ &(Gbest_i^t - x_i^t) + c_3 \cdot rand \cdot vb_i \end{aligned} \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$

式中: $v_i^{t+1}$  和  $x_i^{t+1}$  分别表示第  $t$  次迭代后第  $i$  只天牛的速度和位置;  $c_1$  和  $c_2$  均为学习因子;  $c_3$  为认知因子;  $rand$  为(0,1)内随机数;  $Pbest_i^t$  为个体最优位置;  $Gbest_i^t$  为全局最优个体位置;  $\omega$  为惯性权重,其更新方法为:

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) \frac{t}{t_{max}} \quad (4)$$

式中: $\omega_{max}$  和  $\omega_{min}$  分别为最大和最小惯性权重;  $t_{max}$  是最大迭代次数。

## 2 自适应精英变异的天牛群算法

本文针对天牛群算法易陷入局部最优的问题,提出了一种自适应精英变异策略的改进天牛群算法。该算法采用佳点集初始化天牛种群,优化算法的惯性权重调整机制,并引入精英变异策略,旨在保留优秀的个体,以此维持算法的多样性并防止算法过度拟合到局部最优解。

### 2.1 佳点集初始化

天牛群算法的初始化种群是通过随机初始化获得的。随机初始化生成的种群在整个空间上的分布是不均匀,易使算法陷入局部最优解。使用佳点集理论生成的样本点偏差较小,且样本的偏差只受样本大小影响,因此,使用佳点集初始化的种群在搜索空间分布更加均匀,从而降低算法陷入局部最优解的风险。

佳点集初始化种群的公式为:

$$x_j(i) = lb_j + (ub_j - lb_j) \cdot \{r_j^{(i)} \cdot k\} \quad (5)$$

式中: $ub_j$  和  $lb_j$  分别表示第  $j$  维的上界和下界;  $\{r_j^{(i)} \cdot k\}$  代表佳点集取小数部分。

假设天牛种群的规模为 100,分别采用随机初始化和佳点集初始化两种方法来构建天牛群。如图 1 所示,随机初始化产生的种群在多个区域未能实现覆盖,而佳点集初始化的种群分布则更为均匀,其在解空间中的覆盖范围也更为广泛。因此,采用佳点集初始化的种群可以有效降低算法陷入局部最优的风险。

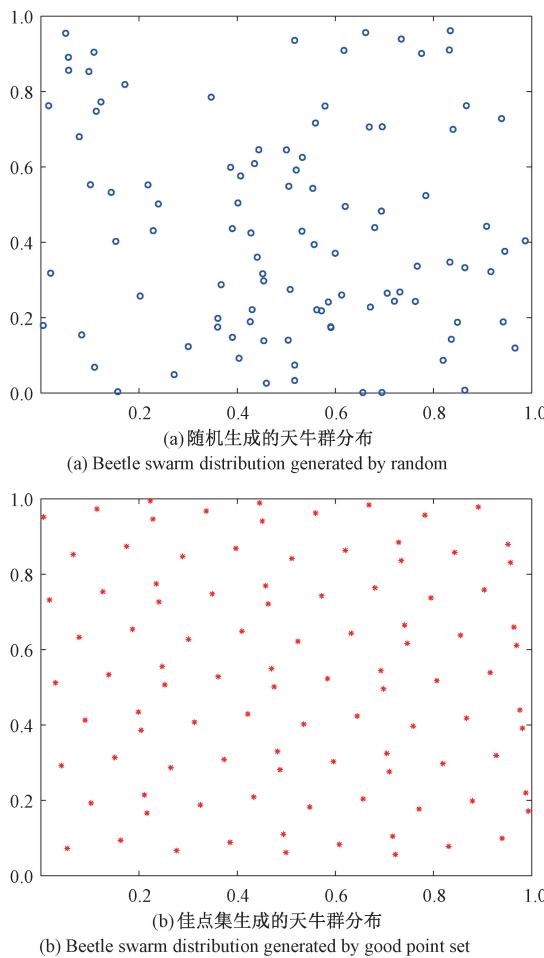


图1 随机生成和佳点集生成的天牛群分布

Fig. 1 Beetle swarm distribution generated by random and good point set

## 2.2 调整惯性权重

惯性权重可以影响天牛群算法的全局搜索能力与局部勘探能力。当惯性权重较大时算法趋向于进行全局搜索,此时种群的搜索范围扩大,较小时利于局部勘探,此时天牛个体在当前点附近进行精细搜寻,以避免过早收敛到局部最优解。当天牛的惯性权重随迭代次数增加而简单的线性减小时会降低搜索精度,因此提出一种非线性递减的惯性权重策略。

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \cdot \cos\left(\frac{\pi}{2} \sqrt{\frac{t}{t_{\max}}}\right) \quad (6)$$

非线性递减惯性权重使  $\omega$  值在搜索后期较小,加强天牛个体的寻优能力,以便获得更好的最优解。

## 2.3 精英变异策略

为了降低算法陷入局部最优的可能性,引入精英变异策略<sup>[20]</sup>,精英变异策略的加入是为了保留优秀个体,并在后续迭代中对其进行微调,以产生新的、可能更优的候选解。

将天牛种群中的全局最优个体  $Gbest$  作为种群的精英天牛个体,在每次迭代过程当中,都根据式(7)~(9)对  $Gbest$  进行变异操作获得新的精英天牛个体  $Gbest^*$ 。若  $Gbest^*$  的适应度值优于  $Gbest$  的适应度值,则用  $Gbest^*$  替代  $Gbest$  进行到后续迭代中。

$$G_1 = k_1 \frac{t}{t_{\max}} \quad (7)$$

$$G_2 = k_2 e^{k_3(f_{avg} - f_{Gbest})} \quad (8)$$

$$Gbest^* = Gbest \cdot [1 \pm (G_1 + G_2 + \mu)] \quad (9)$$

式中: $k_1, k_2, k_3$  为常数; $\mu$  为  $(0, 1)$  内的随机数; $f_{avg}$  为种群平均适应度值; $f_{Gbest}$  为种群最优适应度值。从式(7)~(9)可以看出,在算法迭代前期,  $t$  较小,因此变异主要受  $G_2$  影响,而在算法迭代后期,种群平均适应度值逐渐接近种群最优适应度值,因此变异主要受  $G_1$  影响。

## 3 实验及结果分析

为验证本文提出的自适应精英变异天牛群算法的性能,同时验证非线性惯性权重的效果,记 AEM-BSO2 算法为采用线性递减的惯性权重更新的自适应精英变异天牛群算法,AEM-BSO2 算法的惯性权重采用式(4)进行更新,其余控制策略与 AEM-BSO 一致,将本文提出的 AEM-BSO 分别与 AEM-BSO2、BSO、PSO、BA 进行比较。测试运行环境为 MATLAB R2024b, 所用计算机 CPU 为 Intel Core(TM) i7-7700, 内存为 8 G。5 种算法的种群规模均设置为 100, 最大迭代次数为 1 000,  $\omega_{\max} = 0.9$ ,  $\omega_{\min} = 0.4$ 。其中 BSO 算法参数设置为  $c_1, c_2, c_3$  均为 1.49; PSO 算法参数设置为  $c_1$  和  $c_2$  为 1.49; 本文所用参数设置为  $c_1, c_2, c_3$  均为 1.49。

本文对 10 个基准测试函数进行对比实验,分别在维度  $D=30$  和  $D=60$  下进行测试,重复执行 100 次,统计最优值和平均值。

### 1) Sphere 函数

Sphere 函数为单峰函数,  $x_i$  的取值范围为  $[-100, 100]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优值 0。

$$f_1 = \sum_{i=1}^D x_i^2 \quad (10)$$

### 2) Step 函数

Step 函数为单峰函数,  $x_i$  的取值范围为  $[-100, 100]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优值 0。

$$f_2 = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2 \quad (11)$$

### 3) Rosenbrock 函数

Rosenbrock 函数为单峰函数,  $x_i$  的取值范围为  $[-50, 50]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优值 0。

$$f_3 = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (12)$$

## 4) Quadric 函数

Quadric 函数为单峰函数,  $x_i$  的取值范围为  $[-100, 100]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优点 0。

$$f_4 = \sum_{i=1}^D \left( \sum_{j=1}^i x_j^2 \right) \quad (13)$$

## 5) Schwefel P2. 2 函数。

Schwefel P2. 2 函数为单峰函数,  $x_i$  的取值范围为  $[-10, 10]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优点 0。

$$f_5 = \sum_{i=1}^D |x_i| + \prod_{i=1}^D |x_i| \quad (14)$$

## 6) Elliptic 函数

Elliptic 函数为单峰函数,  $x_i$  的取值范围为  $[-100, 100]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优点 0。

$$f_6 = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2 \quad (15)$$

## 7) Rastrigin 函数。

Rastrigin 函数为多峰函数,  $x_i$  的取值范围为  $[-5.2, 5.2]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优点 0。

$$f_7 = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10] \quad (16)$$

## 8) Griewank 函数。

Griewank 函数为多峰函数,  $x_i$  的取值范围为  $[-600, 600]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优点 0。

表 1 5 种算法在 10 个测试函数上的结果比较

Table 1 Comparison of the results of the five algorithms on 10 test functions

测试 函数	统计 指标	D=30					D=60				
		AEM-BSO	AEM-BSO2	BSO	PSO	BA	AEM-BSO	AEM-BSO2	BSO	PSO	BA
单峰 函数	最优值	<b>2.68×10<sup>-174</sup></b>	4.91×10 <sup>-159</sup>	6.76×10 <sup>-2</sup>	3.15×10 <sup>0</sup>	3.44×10 <sup>0</sup>	<b>1.94×10<sup>-171</sup></b>	9.95×10 <sup>-163</sup>	2.00×10 <sup>2</sup>	3.71×10 <sup>2</sup>	1.53×10 <sup>1</sup>
	平均值	<b>1.98×10<sup>-154</sup></b>	1.99×10 <sup>-143</sup>	4.82×10 <sup>0</sup>	3.09×10 <sup>1</sup>	5.03×10 <sup>0</sup>	<b>2.45×10<sup>-156</sup></b>	1.32×10 <sup>-145</sup>	4.81×10 <sup>2</sup>	8.32×10 <sup>2</sup>	2.23×10 <sup>1</sup>
	最优值	<b>0</b>	<b>0</b>	4.00×10 <sup>1</sup>	5.20×10 <sup>1</sup>	4.00×10 <sup>0</sup>	<b>0</b>	<b>0</b>	6.20×10 <sup>2</sup>	7.61×10 <sup>2</sup>	2.30×10 <sup>1</sup>
	平均值	<b>2.00×10<sup>-2</sup></b>	2.50×10 <sup>-1</sup>	1.48×10 <sup>2</sup>	1.94×10 <sup>2</sup>	6.94×10 <sup>0</sup>	<b>0</b>	<b>0</b>	1.11×10 <sup>3</sup>	1.40×10 <sup>3</sup>	3.30×10 <sup>1</sup>
	最优值	<b>2.91×10<sup>-5</sup></b>	1.07×10 <sup>-2</sup>	9.54×10 <sup>1</sup>	3.65×10 <sup>2</sup>	5.22×10 <sup>2</sup>	<b>4.76×10<sup>1</sup></b>	4.81×10 <sup>1</sup>	8.22×10 <sup>3</sup>	3.92×10 <sup>4</sup>	3.27×10 <sup>3</sup>
	平均值	<b>1.29×10<sup>-1</sup></b>	3.66×10 <sup>-1</sup>	5.97×10 <sup>2</sup>	3.96×10 <sup>3</sup>	1.57×10 <sup>3</sup>	<b>4.94×10<sup>1</sup></b>	4.96×10 <sup>1</sup>	1.01×10 <sup>5</sup>	2.09×10 <sup>5</sup>	6.15×10 <sup>3</sup>
双峰 函数	最优值	<b>4.23×10<sup>-170</sup></b>	4.69×10 <sup>-157</sup>	2.79×10 <sup>1</sup>	1.36×10 <sup>2</sup>	9.15×10 <sup>0</sup>	<b>2.08×10<sup>-170</sup></b>	1.81×10 <sup>-163</sup>	6.01×10 <sup>2</sup>	1.25×10 <sup>3</sup>	2.19×10 <sup>2</sup>
	平均值	<b>2.00×10<sup>-154</sup></b>	2.32×10 <sup>-143</sup>	1.67×10 <sup>2</sup>	3.60×10 <sup>2</sup>	1.54×10 <sup>1</sup>	<b>1.91×10<sup>-152</sup></b>	5.90×10 <sup>-142</sup>	1.70×10 <sup>3</sup>	2.57×10 <sup>3</sup>	3.73×10 <sup>2</sup>
	最优值	<b>1.26×10<sup>-179</sup></b>	1.69×10 <sup>-166</sup>	2.43×10 <sup>-3</sup>	5.01×10 <sup>-2</sup>	3.61×10 <sup>0</sup>	<b>1.52×10<sup>-173</sup></b>	1.52×10 <sup>-163</sup>	1.84×10 <sup>0</sup>	4.33×10 <sup>0</sup>	1.98×10 <sup>1</sup>
	平均值	<b>3.52×10<sup>-157</sup></b>	1.88×10 <sup>-148</sup>	7.29×10 <sup>-2</sup>	4.30×10 <sup>-1</sup>	1.60×10 <sup>2</sup>	<b>7.34×10<sup>-132</sup></b>	6.93×10 <sup>-127</sup>	5.90×10 <sup>0</sup>	9.55×10 <sup>0</sup>	6.38×10 <sup>16</sup>
	最优值	<b>1.26×10<sup>-163</sup></b>	2.67×10 <sup>-156</sup>	6.03×10 <sup>4</sup>	1.20×10 <sup>5</sup>	5.47×10 <sup>5</sup>	<b>1.98×10<sup>-164</sup></b>	1.01×10 <sup>-152</sup>	1.07×10 <sup>6</sup>	1.63×10 <sup>6</sup>	4.30×10 <sup>6</sup>
	平均值	<b>7.38×10<sup>-149</sup></b>	2.48×10 <sup>-139</sup>	2.74×10 <sup>5</sup>	7.58×10 <sup>5</sup>	2.57×10 <sup>6</sup>	<b>1.96×10<sup>-149</sup></b>	1.60×10 <sup>-140</sup>	3.30×10 <sup>6</sup>	6.68×10 <sup>6</sup>	8.82×10 <sup>6</sup>
	最优值	<b>1.69×10<sup>1</sup></b>	3.08×10 <sup>1</sup>	2.11×10 <sup>1</sup>	2.23×10 <sup>1</sup>	2.02×10 <sup>2</sup>	<b>3.48×10<sup>1</sup></b>	4.78×10 <sup>1</sup>	1.13×10 <sup>2</sup>	1.30×10 <sup>2</sup>	5.35×10 <sup>2</sup>
	平均值	<b>5.12×10<sup>1</sup></b>	6.12×10 <sup>1</sup>	5.38×10 <sup>1</sup>	5.26×10 <sup>1</sup>	2.50×10 <sup>2</sup>	<b>1.08×10<sup>2</sup></b>	1.12×10 <sup>2</sup>	1.68×10 <sup>2</sup>	1.97×10 <sup>2</sup>	6.39×10 <sup>2</sup>
	最优值	<b>0</b>	2.08×10 <sup>-1</sup>	2.89×10 <sup>-1</sup>	8.60×10 <sup>-1</sup>	2.03×10 <sup>-1</sup>	<b>0</b>	<b>0</b>	2.84×10 <sup>0</sup>	3.63×10 <sup>0</sup>	4.62×10 <sup>-1</sup>
	平均值	<b>1.77×10<sup>-3</sup></b>	2.66×10 <sup>-1</sup>	9.35×10 <sup>-1</sup>	1.27×10 <sup>0</sup>	2.68×10 <sup>-1</sup>	<b>1.75×10<sup>-3</sup></b>	2.44×10 <sup>-3</sup>	5.19×10 <sup>0</sup>	8.13×10 <sup>0</sup>	5.60×10 <sup>-1</sup>
	最优值	<b>-4.44×10<sup>-16</sup></b>	<b>-4.44×10<sup>-16</sup></b>	2.53×10 <sup>0</sup>	3.11×10 <sup>0</sup>	3.01×10 <sup>0</sup>	<b>-4.44×10<sup>-16</sup></b>	<b>-4.44×10<sup>-16</sup></b>	5.88×10 <sup>0</sup>	6.03×10 <sup>0</sup>	3.91×10 <sup>0</sup>
	平均值	<b>2.58×10<sup>-15</sup></b>	<b>2.58×10<sup>-15</sup></b>	5.66×10 <sup>0</sup>	5.48×10 <sup>0</sup>	5.19×10 <sup>0</sup>	<b>2.43×10<sup>-15</sup></b>	2.72×10 <sup>-15</sup>	7.44×10 <sup>0</sup>	7.54×10 <sup>0</sup>	8.29×10 <sup>0</sup>
$f_7 \sim f_{10}$	最优值	<b>0</b>	<b>0</b>	1.74×10 <sup>1</sup>	2.70×10 <sup>1</sup>	2.81×10 <sup>1</sup>	<b>0</b>	<b>0</b>	1.81×10 <sup>2</sup>	3.59×10 <sup>2</sup>	9.06×10 <sup>1</sup>
	平均值	<b>0</b>	1.10×10 <sup>-1</sup>	4.03×10 <sup>1</sup>	6.25×10 <sup>1</sup>	3.39×10 <sup>1</sup>	<b>0</b>	<b>0</b>	4.57×10 <sup>2</sup>	7.10×10 <sup>2</sup>	1.08×10 <sup>2</sup>

对于  $f_7 \sim f_{10}$  这些双峰函数, 从表 1 可以看出, AEM-BSO 算法和 AEM-BSO2 算法能够有效地跳出局部最优, 成功找到全局最优解, 在所有比较算法中表现最为出色。

$$f_8 = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (17)$$

## 9) Ackley 函数。

Ackley 函数为多峰函数,  $x_i$  的取值范围为  $[-32, 32]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优点 0。

$$f_9 = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) \quad (18)$$

## 10) Bohachevsky 函数。

Bohachevsky 函数为多峰函数,  $x_i$  的取值范围为  $[-50, 50]$ , 在  $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$  时有最优点 0。

$$f_{10} = \sum_{i=1}^{D-1} [x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_i)] + 0.7 \quad (19)$$

表 1 为 5 种算法在 10 个标准函数上分别在维度  $D=30$  和  $D=60$  时求得的最优点和平均值的比较结果。其中,  $f_1 \sim f_6$  为单峰函数, 本文采用的 AEM-BSO 算法在  $D=30$  和  $D=60$  时在这些函数上均取得了最优点和平均值的最佳表现, 且所得结果的精度较高, AEM-BSO2 算法的最优点和平均值率低于 AEM-BSO 算法。

尽管在函数  $f_7$  上的表现尚有提升空间, 但 AEM-BSO 算法的结果仍然优于其他对比算法。以上可以看出在单峰和多峰函数下 AEM-BSO 算法的效果都是比较优秀的, 但

亦显示出在高维情况下,AEM-BSO 针对局部极值点特别多的部分函数仍可能陷入局部极值。同时从 AEM-BSO 和 AEM-BSO2 的对比可以看出采用本文所用非线性递减惯性权重更新方法略优于线性递减的惯性权重更新策略。

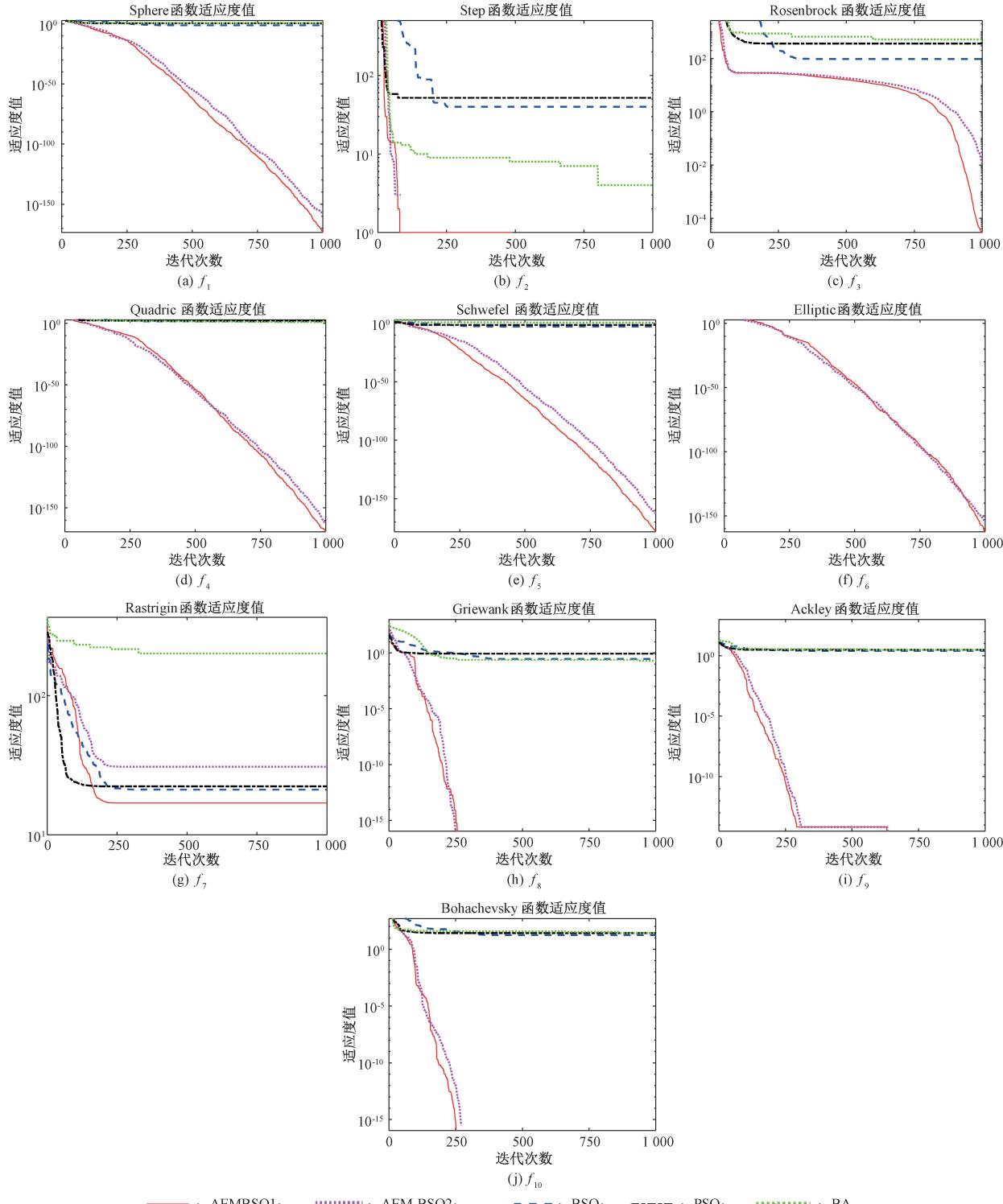


图 2 10 个测试函数最优迭代曲线

Fig. 2 10 Optimal iteration curves of test functions

## 4 基于 AEM-BSO 和三次样条插值的路径规划

三次样条插值法是常用的数值分析方法, 可通过给定的一组散点数据来拟合出一条光滑的连续函数曲线。其基本思想是用低次多项式逼近一段小区间内的数据, 使整个曲线连续光滑。使用三次样条插值进行路径平滑, 可以确保移动机器人的运动的连续性和安全性。

### 4.1 三次样条插值

三次样条插值的定义: 对于已知的  $n$  个数据节点  $(x_i, y_i)$ , 满足  $x_1 < x_2 < \dots < x_{n-1} < x_n$ , 如果函数  $S(x)$  满足以下条件:

- 1) 在每个子区间  $[x_{i-1}, x_i]$ , ( $i = 1, 2, \dots, n$ ), 均为三次多项式;
  - 2) 在两个相邻的子区间的连接处拥有相同的一阶导数和二阶导数;
  - 3) 经过所有的给定的数据点, 即  $S(x_i) = y_i$ 。
- 则称  $S(x)$  为  $n$  个数据节点的三次样条插值函数。

### 4.2 构造插值路径

将三次样条插值每一个分段的交点作为路径节点, 将一条路径的所有路径节点作为单个天牛个体, 假设一条路径上共有  $m$  个路径节点, 路径节点坐标可表示成  $R = [(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \dots, (x_{im}, y_{im})]$ , 可得路径节点横纵坐标的集合分别为  $R_x = \{x_{i1}, x_{i2}, \dots, x_{im}\}$  和  $R_y = \{y_{i1}, y_{i2}, \dots, y_{im}\}$ , 将路径节点横纵坐标集合  $R_x$  和  $R_y$  合并作为天牛个体坐标:

$$x = R_x \cup R_y = [x_{i1}, x_{i2}, \dots, x_{im}, y_{i1}, y_{i2}, \dots, y_{im}] \quad (20)$$

故该天牛个体的维数为  $2m$ , 由此可知天牛群的维数是路径节点个数的 2 倍。再结合已知的起始位置坐标和目标位置坐标, 便可通过三次样条插值的方法获得  $n$  个插值点的坐标, 插值点坐标的连线就是规划的全局路径曲线。

### 4.3 路径评价

移动机器人路径规划的曲线通常是尽可能短的, 并且不能与障碍物有交点。因此可以构建带有罚函数的路径评价适应度函数:

$$z = L(1 + M \cdot \xi) \quad (21)$$

式中:  $L$  表示规划的路径总长度;  $M$  为惩罚系数, 一般设置为 100;  $\xi$  表示规划的路径和  $\xi$  个障碍物之间的距离小于安全距离  $d_{safe}$ , 若  $\xi=0$ , 则  $z=L$ , 此时路径评价适应度的大小就是路径长度。

### 4.4 路径规划流程

步骤 1) 创建环境模型, 初始化移动机器人的起始位置和目标位置, 初始化工作空间;

步骤 2) 初始化算法参数, 包括种群规模、最大迭代次数、种群维数、插值点个数等参数, 使用佳点集初始化天牛群位置  $x$ , 初始速度  $v$ , 求出个体最优解和全局最优解;

步骤 3) 进行算法迭代, 根据式(6)更新惯性权重, 根据式(1)~(3)更新天牛的位置  $x$  和速度  $v$ ;

步骤 4) 利用三次样条插值的方法, 求出插值点坐标, 结合起始位置和目标位置获得全部路径节点, 计算路径节点之间的距离, 得到路径总长度  $L$ , 确定  $\xi$  的值;

步骤 5) 根据式(21)求出路径适应度值, 更新个体最优  $Pbest$  和全局最优  $Gbest$ ;

步骤 6) 根据式(7)~(9)进行精英变异生成  $Gbest^*$ , 若  $Gbest^*$  的路径评价小于  $Gbest$  则替换全局最优个体为  $Gbest^*$ , 否则不变;

步骤 7) 判断算法是否达到最大迭代次数, 若满足则停止搜索输出全局最优路径; 否则返回步骤 3) 继续迭代。

## 5 路径规划效果测试

测试环境地图设置为  $100 \times 100$ , 起始坐标点  $Begin$  为  $[10, 10]$ , 目标坐标点  $Goal$  为  $[90, 90]$ , 惩罚系数  $M=100$ , 安全距离为 1.5, 最大迭代次数为 500, 重复执行 100 次。简单环境下障碍物个数为 10, 复杂环境下障碍物数量为 30。

### 5.1 插值方法比较

为验证三次样条插值方法的有效性, 将本文提出的 AEM-BSO 算法分别和线性插值 (linear)、最近邻插值 (nearest) 以及三次样条插值 (spline) 进行结合, 由 AEM-BSO 算法规划路径节点, 分别使用 3 种插值方法构造插值路径, 分别在简单环境和复杂环境下进行路径规划。

表 2 为 AEM-BSO 算法结合 3 种插值方法在简单环境和复杂环境下规划出的路径长度和单次运行时间耗时的比较结果。通过分析表 2 的数据, 可以观察到, 在简单环境中, 采用三次样条插值方法规划出的路径长度是最短的。而在复杂环境下, 虽然线性插值方法得到的路径长度最短, 但三次样条插值与线性插值规划出的路径长度相差无几, 3 种插值算法单次运行所用时间基本一致。因此, 三次样条插值的综合性最好。

图 3 所示为在简单环境和复杂环境下, 不同插值方法进行路径规划。从图 3 可以发现, 在这两种环境下, 无论是线性插值还是最近邻插值规划出的路径都存在多个拐点, 而相比之下, 三次样条插值规划的路径则显得最为平滑。路径中的拐点过多对于移动机器人的运动是不利的, 因为它们可能导致机器人运动的不连续。

性和不稳定性。相反,平滑的曲线更适合作为移动机器人的路径点,因为它能够确保机器人运动的流畅性和安全性。

**表 2 不同环境下 3 种插值方法路径长度和耗时**

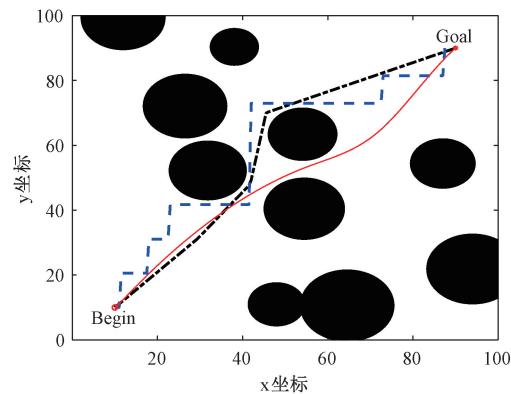
**Table 2 Path length and time consumption of three interpolation methods in different environments**

算法	简单环境			复杂环境		
	最短 长度	平均 长度	单次耗 时/s	最短 长度	平均 长度	单次耗 时/s
线性插值	120. 68	121. 08	2. 62	115. 24	139. 67	5. 20
最近邻插值	158. 81	159. 28	2. 60	159. 01	159. 76	4. 89
三次样条插值	114. 86	146. 29	2. 64	116. 17	160. 19	4. 94

因此,本文选择将 AEM-BSO 算法与三次样条插值方法相结合的方式用于移动机器人的路径规划,以期获得更优的路径规划结果。

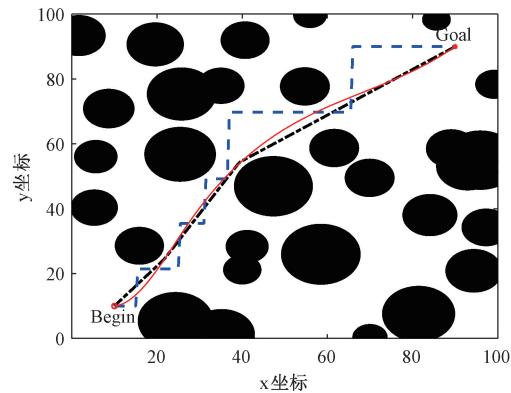
## 5.2 简单环境和复杂环境的效果比较

为评估 AEM-BSO 算法与三次样条插值在不同环境下的路径规划效果,对 AEM-BSO、BSO、PSO 和 BA 这 4 种算法进行仿真对比实验,分别采用 4 种对比算法规划全局路径节点,而后用三次样条插值平滑曲线,分别在简单环境和复杂环境下进行路径规划。所有算法的参数均按照第 3 节的设定进行配置。表 3 为 4 种算法结合三次样条插值在简单环境和复杂环境下规划出的路径长度和单次运行所耗时间的对比分析。图 4 和 5 为不同环境下的迭代曲线和路径规划。



(a) 简单环境下不同插值路径规划图

(a) Path planning diagrams with different interpolation in simple environments



(b) 复杂环境下不同插值路径规划图

(b) Path planning diagrams with different interpolation in complex environments

----：线性插值（linear）；---：最近邻插值（nearest）；  
—：三次样条插值（spline）；

图 3 不同插值方法路径规划

Fig. 3 Different interpolation path planning diagram

**表 3 不同环境下 4 种算法路径长度和耗时**

**Table 3 Four algorithm path lengths and time consumption in different environments**

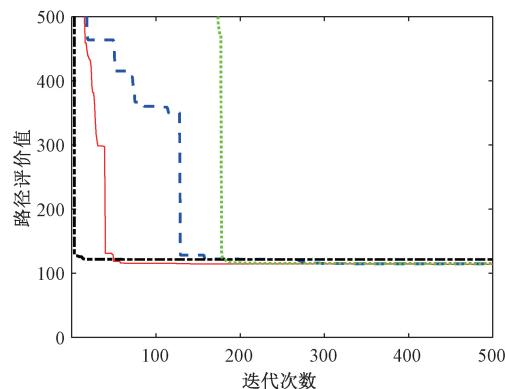
算法	简单环境				复杂环境			
	最短长度	平均长度	标准差	单次耗时/s	最短长度	平均长度	标准差	平均耗时/s
AEM-BSO	114. 11	121. 65	1. 50	2. 35	115. 94	158. 51	8. 57	4. 60
BSO	115. 00	209. 06	99. 04	2. 35	116. 22	162. 20	11. 55	4. 62
PSO	121. 67	361. 94	247. 32	0. 78	141. 59	279. 31	262. 14	1. 54
BA	115. 48	310. 81	609. 00	0. 79	126. 59	165. 53	9. 98	1. 58

根据表 3 的数据,可以明显观察到,在简单环境和复杂环境下,AEM-BSO 算法规划出的最短路径长度均为所有算法中最短,并且其平均路径长度也是所有算法中最小的,在简单环境 AEM-BSO 算法的最短路径长度较 BSO、PSO、BA 分别减少了 0.77%、6.21%、1.29%,标准差分别减少了 98.49%、99.39%、99.75%;在复杂环境下 AEM-BSO 算法的最短路径长度较 BSO、PSO、BA 分别减少了 0.24%、18.12%、8.41%,标准差分别减少了 25.8%、96.73%、14.13%。AEM-BSO 和 BSO 算法的单次运行所用时间相差不大,均比 PSO 和 BA 要更久一些。

这一结果表明 AEM-BSO 算法在路径的最优化和平均性能上的表现是不错的,但运算速度会比 PSO 和 BA 略慢。

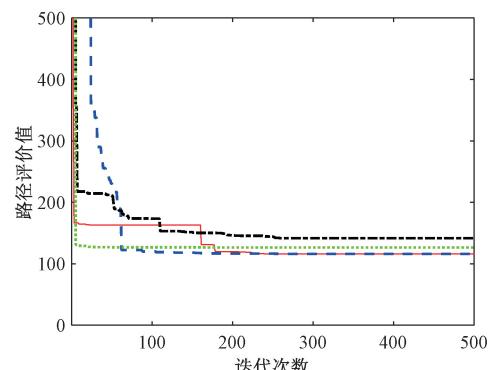
从图 4 可以看出本文算法可以较快的收敛,图 4(b)在复杂环境下,能够多次跳出局部最优,从而找到最短路径。由图 5 可以看出在简单环境和复杂环境下,4 种算法都能够避开障碍物规划出一条无碰撞的路径,并且本文算法规划的路径都是较优的。

综上可以看出 AEM-BSO 能够比较快的收敛并规划出最短路径,要优于对比算法。



(a) 简单环境下路径评价迭代曲线

(a) Iterative curve of path evaluation in simple environment



(b) 复杂环境下路径评价迭代曲线

(b) Iterative curve of path evaluation in complex environment

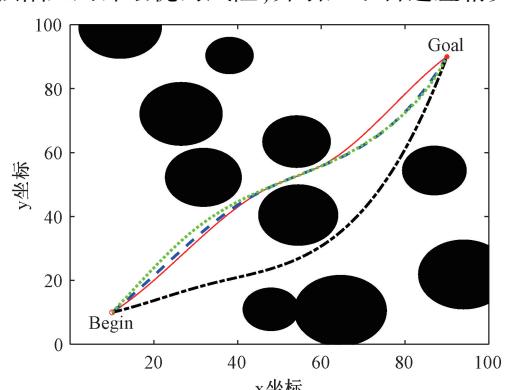
—: AEMBSO; - - : BSO; - - - : PSO; - - - - : BA

图 4 不同环境下路径评价迭代曲线

Fig. 4 Iterative curve of path evaluation in different environments

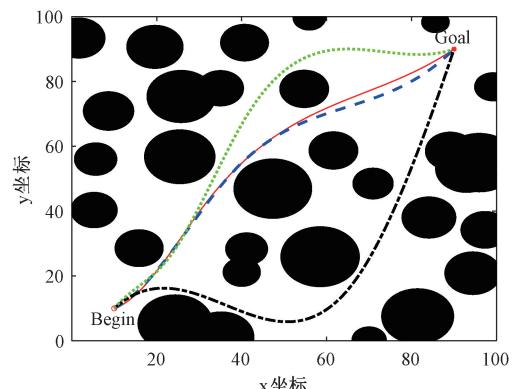
## 6 结 论

本文提出了一种自适应精英变异的天牛群算法, 并将其与三次样条插值相结合, 应用于移动机器人的路径规划问题。通过采用佳点集初始化天牛种群, 有效降低了算法陷入局部最优的风险, 并引入了自适应精英变异



(a) 简单环境下路径规划图

(a) Path planning diagram in simple environment



(b) 复杂环境下路径规划图

(b) Path planning diagram in complex environment

—: AEMBSO; - - : BSO; - - - : PSO; - - - - : BA

图 5 不同环境下路径规划

Fig. 5 Path planning diagram in different environments

策略。该策略旨在保留优秀的个体, 并在后续迭代中对其进行细致调整, 以生成新的、可能更优的候选解。这不仅有助于维持算法的多样性, 还能避免算法过度拟合到当前的局部最优解。在将改进的天牛群算法应用于移动机器人路径规划时, 首先利用该算法寻找一条较优的路径, 然后通过三次样条插值技术平滑路径, 消除路径中的锐角和突变, 确保移动机器人运动的连续性和安全性。为了验证算法的性能, 在简单环境和复杂环境中进行了仿真实验。实验结果表明, 结合三次样条插值的改进天牛群算法能够有效解决移动机器人的路径规划问题。与传统的天牛群算法及其他优化算法相比, 改进后的算法在避免陷入局部最优和提高路径平滑度方面展现出了更优越的性能, 为移动机器人的路径规划提供了一种高效且可靠的解决方案。未来将进一步将算法用于实际场景, 并在此基础上研究动态障碍物场景下算法的适用性。

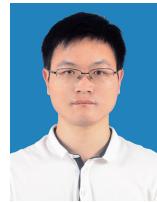
## 参考文献

- [1] 齐凤莲, 王晓庆, 张帼英. 改进 A\* 算法的 AGV 避障路径规划研究 [J]. 机床与液压, 2023, 51(9):34-39.  
QI F L, WANG X Q, ZHANG G Y. Research on AGV obstacle avoidance path planning based on improved A\* algorithm [J]. Machine Tool & Hydraulics, 2023, 51(9):34-39.
- [2] 杨国, 吴晓, 肖如奇, 等. 改进 A\* 算法的安全高效室内全局路径规划 [J]. 电子测量与仪器学报, 2024, 38(7):131-142.  
YANG G, WU X, XIAO R Q, et al. Improved A\* algorithm for secure and efficient indoor global path planning [J]. Journal of Electronic Measurement and Instrumentation, 2024, 38(7):131-142.

- [ 3 ] 骆海涛,孙嘉泽,高鹏宇,等. 基于改进 RRT \* 算法的智能轮椅全局路径规划研究 [J]. 仪器仪表学报, 2023,44(10):303-313.
- LUO H T, SUN J Z, GAO P Y, et al. Intelligent wheelchair global path planning research based on the improved RRT \* algorithm [J]. Chinese Journal of Scientific Instrument, 2023,44(10):303-313.
- [ 4 ] 冯垚,周志峰,沈亦纯,等. 基于改进 RRT 算法的避障路径规划 [J]. 工程设计学报, 2023,30(6):707-716.
- FENG Y, ZHOU ZH F, SHEN Y CH, et al. Obstacle avoidance path planning based on improved RRT algorithm [J]. Chinese Journal of Engineering Design, 2023,30(6):707-716.
- [ 5 ] 董丽莎. 基于改进 Dijkstra 算法的轮式移动机械臂无碰撞轨迹研究 [J]. 制造业自动化, 2022,44(8):66-69.
- DONG L SH. Research on collision free trajectory of wheeled mobile manipulator based on improved Dijkstra algorithm [J]. Manufacturing Automation, 2022,44(8):66-69.
- [ 6 ] 高岳林,武少华. 基于自适应粒子群算法的机器人路径规划 [J]. 郑州大学学报(工学版), 2020,41(4):46-51.
- GAO Y L, WU SH H. Robot path planning based on adaptive particle swarm optimization [J]. Journal of Zhengzhou University (Engineering Science), 2020,41(4):46-51.
- [ 7 ] 欧阳海滨,全永彬,高立群,等. 基于混合遗传粒子群优化算法的层次路径规划方法 [J]. 郑州大学学报(工学版), 2020,41(4):34-40.
- OUYANG H B, QUAN Y B, GAO L Q, et al. Hierarchical path planning method for mobile robots based on hybrid genetic particle swarm optimization algorithm [J]. Journal of Zhengzhou University (Engineering Science), 2020,41(4):34-40.
- [ 8 ] 杨教,陆安江,彭熙舜,等. 基于改进粒子群算法的三维路径规划研究 [J]. 电子测量技术, 2023,46(12):92-97.
- YANG J, LU AN J, PENG X SH, et al. Research on 3D path planning based on improved particle swarm optimization [J]. Electronic Measurement Technology, 2023,46(12):92-97.
- [ 9 ] 张红柱,蒋奇. 基于改进离散粒子群优化算法的路径规划 [J]. 计算机仿真, 2022,39(9):462-466,496.
- ZHANG H ZH, JIANG Q. Path planning based on improved discrete particle swarm optimization algorithm [J]. Computer Simulation, 2022,39(9):462-466,496.
- [ 10 ] 刘景森,吉宏远,李煜. 基于改进蝙蝠算法和三次样条插值的机器人路径规划 [J]. 自动化学报, 2021,47(7):1710-1719.
- LIU J S, JI H Y, LI Y. Robot path planning based on improved bat algorithm and cubic spline interpolation [J]. Acta Automatica Sinica, 2021,47(7):1710-1719.
- [ 11 ] 胡小建,杨智. 基于混合遗传算法的多拣货小车路径规划研究 [J]. 合肥工业大学学报(自然科学版), 2022,45(12):1715-1722.
- HU X J, YANG ZH. Research on path planning of multi picking car based on hybrid genetic algorithm [J]. Journal of Hefei University of Technology (Natural Science), 2022,45(12):1715-1722.
- [ 12 ] 徐万福,孙渊. 基于多步长蚁群算法的移动机器人路径规划 [J]. 组合机床与自动化加工技术, 2023(6):18-21,26.
- XU W F, SUN Y. Research of robot path planning based on multi-step ant colony algorithm [J]. Modular Machine Tool & Automatic Manufacturing Technique, 2023(6):18-21,26.
- [ 13 ] 牛晶,申传艳,张利鹏,等. 基于改进 ACO-DWA 算法的轮式植保机器人避障路径研究 [J]. 电子测量与仪器学报, 2024,38(5):188-200.
- NIU J, SHEN CH Y, ZHANG L P, et al. Research on obstacle avoidance path of wheeled plant protection robot based on improved ACO-DWA algorithm [J]. Journal of Electronic Measurement and Instrumentation, 2024,38(5):188-200.
- [ 14 ] 赵辉,郝梦雅,王红君,等. 基于改进 A~\* 算法与天牛须搜索算法的农业机器人路径规划方法 [J]. 科学技术与工程, 2019,19(31):185-190.
- ZHAO H, HAO M Y, WANG H, et al. Research on path planning method of agricultural robot based on improved A~\* algorithm and beetle antennae search algorithm [J]. Science Technology and Engineering, 2019,19(31):185-190.
- [ 15 ] 王槐彬,彭雪,夏小云,等. 基于改进天牛须搜索的自导引机器人路径规划 [J]. 机械设计与研究, 2021,37(5):27-30,46.
- WANG H B, PENG X, XIA X Y, et al. Path planning of automated guided vehicle based on improved beetle antennae search [J]. Machine Design and Research, 2021,37(5):27-30,46.

- [16] NA F, FEI F, GUANGLIN X, et al. Deep reinforcement learning based AGV self-navigation obstacle avoidance method [J]. Instrumentation, 2022, 9(4): 11-16.
- [17] 沈显庆, 孙启智. BSO 算法在移动机器人三维路径规划中的应用 [J]. 黑龙江科技大学学报, 2019, 29(6): 747-751.
- SHEN X Q, SUN Q ZH. Three dimensional path planning of mobile robot based on BSO algorithm [J]. Journal of Heilongjiang University of Science & Technology, 2019, 29(6): 747-751.
- [18] 吕昱呈, 莫愿斌. 融入变异交叉的改进牛顿法求解 TSP 及工程应用 [J]. 计算机应用研究, 2021, 38(12): 3662-3666.
- LYU Y CH, MO Y B. Improved beetle antennae search algorithm with mutation crossover in TSP and engineering application [J]. Application Research of Computers, 2021, 38(12): 3662-3666.
- [19] 王蕊, 李金洺, 史玉龙, 等. 基于视觉的机场无人驱鸟车路径规划算法 [J]. 北京航空航天大学学报, 2024, 50(5): 1446-1453.
- WANG R, LI J M, SHI Y L. Vision-based path planning algorithm of unmanned bird-repelling vehicles in airports [J]. Journal of Beijing University of Aeronautics and Astronautics, 2024, 50(5): 1446-1453.
- [20] 康岚兰, 董文永, 宋婉娟, 等. 无惯性自适应精英变异反向粒子群优化算法 [J]. 通信学报, 2017, 38(8): 66-78.
- KANG L L, DONG W Y, SONG W J, et al. Non-inertial opposition-based particle swarm optimization with adaptive elite mutation [J]. Journal on Communications, 2017, 38(8): 66-78.

## 作者简介



**欧阳鹏**(通信作者), 2020 年于江西理工大学获得硕士学位, 现任江西理工大学实验师, 主要研究方向为移动机器人路径规划。

E-mail: timeinpalm@ qq. com

**Ouyang Peng** (Corresponding author)

received his M. Sc. degree from Jiangxi University of Science and Technology in 2020. Now he is an experimentalist in Jiangxi University of Science and Technology. His main research interests include mobile robot path planning.



**邝先验**, 2014 年于华南理工大学获得博士学位, 现任江西理工大学教授, 硕士生导师, 主要研究方向为智能控制理论与应用、系统建模与仿真、智能交通系统、图像处理与应用。

E-mail: xianyankuang@ 163. com

**Kuang Xiyan** received Ph. D. from South China University of Technology. Now he is a professor and M. Sc. tutor in Jiangxi University of Science and Technology. His main research interests include intelligent control theory and application, system modeling and simulation, intelligent transportation system, image processing and application.



**叶景贞**, 2020 年于昆明理工大学获得硕士学位, 现任江西理工大学讲师, 主要研究方向为计算机视觉与图像处理, 机器学习与数据挖掘。

E-mail: jzye2013@ 163. com

**Ye Jingzhen** received his M. Sc. degree

from Kunming University of Science and Technology in 2020. Now he is a lecturer in Jiangxi University of Science and Technology. His main research interests include computer vision and image processing, machine learning and data mining.