

多策略改进鸽群优化算法的机器人路径规划

徐浩然¹, 王 雷¹, 王紫益¹, 张桐彬¹, 夏强强²

(1. 安徽工程大学机械与汽车工程学院, 芜湖 241000; 2. 长三角哈特机器人产业技术研究院, 芜湖 241000)

摘要: 基于传统鸽群优化算法在处理高维优化问题时存在的局限性, 提出一种改进的螺旋-动态鸽群优化 (Modified spiral-dynamic pigeon-inspired optimization, MSDPIO) 算法, 用于解决机器人路径规划问题中的收敛速度慢和易陷入局部最优解等问题。首先, 通过引入 Logistic 混沌映射初始化策略以扩大搜索范围; 其次, 设计螺旋搜索策略和动态反向学习策略以改进位置更新机制, 提高算法的收敛速度和解的质量。同时, 采用自适应余弦函数调整反向学习权重和改进地标操作提高算法的自适应能力与全局搜索能力。通过对 10 个 CEC2017 基准测试函数的实验, 有效地评估了算法的性能。最后, 将 MSDPIO 算法与基于改进的 B 样条曲线应用于不同尺度 (20 m × 20 m 与 40 m × 40 m) 地图的路径规划问题。仿真结果表明: 在小规模 20 m × 20 m 地图中较鸽群优化 (Pigeon-inspired optimization, PIO) 算法、多策略融合的天鹰优化 (Multi-strategy improved aquila optimizer, MSIAO) 算法、灰狼优化 (Grey wolf optimizer, GWO) 算法路径长度分别改进 1.46%、1.43%、1.47%; 在大规模 40 m × 40 m 地图中收敛性能最大提升 37.82%。

关键词: 机器人路径规划; 改进鸽群优化算法; 混沌映射; 动态反向学习; 螺旋搜索; B 样条曲线

中图分类号: TP242

文献标志码: A

文章编号: 1005-2615(2025)05-0900-12

Robot Path Planning Based on Multi-strategy Improved Pigeon-Inspired Optimization Algorithm

XU Haoran¹, WANG Lei¹, WANG Ziyi¹, ZHANG Tongbin¹, XIA Qiangqiang²

(1. School of Mechanical and Automotive Engineering, Anhui Polytechnic University, Wuhu 241000, China; 2. Yangtze River Delta Hart Robot Industry Technology Research Institute, Wuhu 241000, China)

Abstract: Based on the limitations of traditional pigeon swarm optimization algorithms in dealing with high-dimensional optimization problems, a modified spiral-dynamic pigeon-inspired optimization (MSDPIO) algorithm is proposed to solve problems such as slow convergence speed and susceptibility to local optima in robot path planning. Firstly, by introducing the logistic chaotic mapping initialization strategy, the search range is expanded. Secondly, spiral search strategies and dynamic reverse learning strategies are designed to improve the position update mechanism and enhance the convergence speed and resolution quality of the algorithm. At the same time, the adaptive cosine function is used to adjust the reverse learning weights and improve landmark operations to enhance the algorithm's adaptive and global search capabilities. The performance of the algorithm is effectively evaluated through experiments on ten CEC2017 benchmark test functions. MSDPIO algorithm and improved B-spline curves is applied to path planning problems on maps of different scales (20 m × 20 m and 40 m × 40 m). Simulation results show that in a small-scale 20 m × 20 m

基金项目: 安徽省高校优秀拔尖人才培养项目 (gxbjZD2022023); 安徽省机器视觉检测与感知重点实验室开放基金项目 (KLMVI-2024-HIT-15)。

收稿日期: 2025-03-26; **修订日期:** 2025-05-15

通信作者: 王雷, 男, 教授, 硕士生导师, E-mail: wangdalei2000@126.com。

引用格式: 徐浩然, 王雷, 王紫益, 等. 多策略改进鸽群优化算法的机器人路径规划[J]. 南京航空航天大学学报(自然科学版), 2025, 57(5): 900-911. XU Haoran, WANG Lei, WANG Ziyi, et al. Robot path planning based on multi-strategy improved pigeon-inspired optimization algorithm[J]. Journal of Nanjing University of Aeronautics & Astronautics (Natural Science Edition), 2025, 57(5): 900-911.

map, MSDPIO algorithm improves the path length by 1.46%, 1.43%, and 1.47% compared to pigeon-inspired optimization (PIO) algorithm, multi-strategy improved aquila optimizer (MSIAO) algorithm, and grey wolf optimizer (GWO) algorithm, respectively. The maximum convergence performance improvement in a large-scale $40\text{ m} \times 40\text{ m}$ map is 37.82%.

Key words: robot path planning; improved pigeon-inspired optimization algorithm; chaotic mapping; dynamic opposition-based learning; spiral search; B-spline curve

随着机器人技术的快速发展,路径规划作为机器人自主导航的关键技术之一,其性能直接影响机器人的工作效率和安全性。机器人路径规划的核心目标是在给定环境中寻找一条从起点到终点的最优或近似最优路径,该路径需同时满足避障和路径长度最短等多个约束条件。

目前,求解路径规划问题的方法主要分为两类:传统确定性算法和智能优化算法。传统算法包括Dijkstra算法^[1]、A*算法^[2]等,这类算法在简单环境下能快速得到最优解,在复杂动态环境中计算效率较低。智能优化算法如天鹰优化(Aquila optimizer, AO)^[3]算法,蜣螂优化(Dung beetle optimizer, DBO)^[4]算法,鸽群优化(Pigeon-inspired optimization, PIO)^[5]算法等由于其良好的全局搜索能力和较强的环境适应性,在路径规划领域得到广泛应用。其中,PIO算法相比其他算法具有需要调整的参数较少、收敛速度快、全局搜索能力较强且算法整体公式较少易于理解等优点,因此被广泛应用于路径规划领域。国内外诸多学者尝试对传统的鸽群优化算法进行改进,如李兆博等^[6]将鸽群优化算法用于多无人车协同编队避障控制,通过增加方向协同因子和队形保持因子,并结合变权重鸽群优化算法优化路径评价函数,显著提高了编队避障的稳定性和精确性。秦昌礼等^[7]针对粒子群(Particle swarm optimization, PSO)算法易陷入局部最优的问题,结合PIO算法的快速收敛能力,提出了一种改进的PSO-PIO混合算法。该算法通过分析PSO算法的局限性并优化栅格地图大小,有效避免了局部最优,全局路径长度缩短约3.8%,路径规划时间减少。刘昂等^[8]针对鸽群优化算法通过引入模拟退火准则、高斯扰动和引入对数S形传递函数对鸽群数量的步长进行优化,有效避免算法陷入局部最优,提高了种群多样性。Alazzam等^[9]提出了一种基于鸽群启发式优化的入侵检测系统特征选择算法。该算法利用鸽群算法的全局收敛优势,通过引入余弦相似度概念来二值化连续鸽群算法,以适应离散问题。

尽管上述改进方法在不同程度上提升了鸽群优化算法的性能,但这些改进仍然存在一些局限性。例如,算法的收敛性能依然受到初始种群分布

和搜索策略的显著影响,且在处理高维复杂问题时,算法的计算效率和鲁棒性仍有待进一步提升。传统鸽群优化算法通常采用随机初始化方式生成初始种群,这种方式容易导致种群分布不均,从而影响算法的收敛速度。同时,现有的搜索策略在全局探索与局部开发之间的平衡能力不足,容易陷入局部最优。尽管已有多种优化策略在其他智能算法中被证实具有良好效果,但这些策略在鸽群优化算法的改进研究中尚未得到充分应用。

针对以上存在的问题,本文提出一种改进的螺旋-动态鸽群优化(Modified spiral-dynamic pigeon-inspired optimization, MSDPIO)算法用于解决机器人路径规划问题。首先,引入Logistic混沌映射替代随机初始化,提高种群的多样性和分布的均匀性。其次,结合螺旋搜索策略改进个体位置的更新机制,增强算法的局部搜索能力。同时,通过动态反向学习策略提高算法跳出局部最优的能力,并采用自适应余弦函数动态调整反向学习权重,实现了算法在搜索过程中探索与利用的动态平衡,有效提高了收敛性能。为验证改进策略的有效性,本文对10个CEC2017基准测试函数以及机器人路径规划问题的解决进行仿真验证,结果表明,所提出改进算法的优化性能得到了明显改善。

1 鸽群优化算法

鸽群优化算法是一种基于模拟鸽群中导航行为的群体智能优化算法,它是由段海滨教授团队于2014年提出的。该算法通过仿生学原理的应用展现了大自然对解决复杂优化问题的启发,其最初被用于空中机器人路径规划,随后,在电力系统优化、图像处理等领域也展现出较好的优化能力。

鸽群优化算法借鉴了鸽子独特的导航能力,通过模拟两种关键的导航机制来求解全局优化问题,其核心机制包括以下两个阶段。

1.1 指南针和地图操作

在鸽群优化算法中,鸽子的行为依照迭代次数分为两个阶段。其中,阶段1占整个迭代次数 I_{MaxIter} 的比例为 R_{NcRate} ,一般取0.75,当迭代次数在 $[0, I_{\text{MaxIter}} \cdot R_{\text{NcRate}}]$ 中时则执行地图和指南针操作,当迭代次数

大于 $I_{\text{MaxIter}} \cdot R_{\text{NcRate}}$ 时则执行阶段 2, 即地标操作。

指南针操作: 通过衰减因子 λ 逐步降低历史速度权重, 引导种群向当前最优解 X_{best}^t 聚集, 其数学表达式如下

$$V_{\text{new}}^t = V_i^t \cdot e^{-\lambda t} + (X_{\text{best}}^t - X_i^t) \cdot \text{rand} \quad (1)$$

式中: V_{new}^t 表示第 t 次迭代所得新的速度; V_i^t 表示第 i 只鸽子的第 t 次迭代的速度; λ 表示衰减因子, 一般取 0.2; X_{best}^t 表示第 t 次迭代的全局最佳位置; X_i^t 表示第 i 只鸽子的第 t 次迭代的位置; rand 表示 $[0, 1]$ 之间的随机数。

地图操作: 模拟鸽子通过内在的认知, 对环境地形进行感知形成地图, 用于更新位置, 其数学模型为

$$X_{\text{new}}^t = X_i^t + V_i^t \quad (2)$$

式中 X_{new}^t 表示第 t 次迭代通过鸽子的速度及鸽子的位置相加所得的新的位置。

1.2 地标操作

当迭代次数大于 $I_{\text{MaxIter}} \cdot R_{\text{NcRate}}$ 时, 算法执行阶段 2, 淘汰前一半种群, 按前一半个体质心 X_c^t 位置更新位置, 其数学表达式如下

$$X_c^t = \frac{S_{\text{XF}}}{S_{\text{NF}}} \quad (3)$$

$$X_i^{t+1} = X_i^t + \text{rand} \cdot (X_c^t - X_i^t) \quad (4)$$

$$S_{\text{XF}} = \sum_{i=1}^{\frac{N}{2}} (X_i^t \cdot F(X_i)) \quad (5)$$

$$S_{\text{NF}} = \frac{N}{2} \cdot \sum_{i=1}^{\frac{N}{2}} F(X_i) \quad (6)$$

式中: X_c^t 表示第 t 次迭代下的种群质心位置; $F(X_i)$ 表示第 i 只鸽子的适应度; X_i^{t+1} 表示第 t 次迭代下根据质心所计算出的新的位置; N 为种群数量。

1.3 鸽群优化算法局限性分析

尽管传统鸽群算法通过指南针操作式(1)与地标操作式(3~6)的结合实现了全局搜索与局部开发的平衡, 但其数学模型中存在以下关键缺陷, 导致收敛速度受限且易陷入局部最优。

首先, 在种群初始化阶段上, 传统鸽群算法采用随机初始化, 导致种群分布不均, 这种初始化方式难以覆盖高维解空间, 易使算法早期陷入局部最优区域。

其次, 指南针操作具有过早收敛倾向, 地标操作会造成种群多样性损失。在传统鸽群算法的指南针操作中, 速度更新公式(式(1))中的速度衰减项 $e^{-\lambda t}$ 的指数衰减特性导致搜索后期速度趋近于 0 ($\lambda=0.2$ 时尤为显著), 使得算法过早依赖当前最优解 X_{best}^t , 削弱了全局探索能力。在地标操作(式(3~6))中淘汰后一半的种群仅更新前一半个体, 导致种群规模持续性衰减, 这种机制降低了种群的多样性, 使得算法

在复杂多峰问题中难以跳出局部最优。

最后, 鸽群优化算法在局部搜索能力方面存在不足: 由位置更新公式(2)可知, 仅依赖速度向量 V_i^t 的线性叠加, 缺乏对潜在最优解的局部精细化搜索能力, 导致算法在复杂地形中收敛精度不足。

2 改进鸽群优化算法

针对上述传统鸽群优化算法存在的缺陷, 本文提出以下改进策略以优化传统鸽群算法在种群初始化、指南针操作与地标操作上的缺陷。

2.1 混沌初始化策略

混沌现象是一种确定性非线性动力系统中普遍存在的复杂动力学行为, 其演化具有高度的不规则性, 不可预测性和内在的随机性。混沌系统对初值条件具有极其敏感的依赖性, 类似蝴蝶效应, 系统初始状态的微小差异经过有限次迭代后会导致完全不同的演化轨迹。这种特性使得混沌序列在维持确定性的同时又表现出类随机的行为, 其产生的序列具有良好的均匀性和遍历性, 能够确保初始解覆盖更多潜在最优区域, 从根源上避免早熟收敛。

常用于优化算法中的混沌映射包括 Logistic 映射^[10]、Tent 映射^[11]、Sine 映射^[12]、Chebyshev 映射^[13]等, 其中 Logistic 映射因其简单性和良好的混沌特性而被广泛应用。故本文采用 Logistic 混沌映射来初始化种群, 其迭代方程为

$$X_{n+1} = r \cdot X_n (1 - X_n) \quad (7)$$

式中: r 为控制变量, X_n 表示第 n 次迭代的混沌变量。

当 r 的值不同时系统会呈现出不同的特性, 当 $0 < r < 3$ 时系统最终收敛到固定点, 当 $3 < r < 3.57$ 时系统进入周期倍分岔区域, 当 $3.57 < r < 4$ 时系统进入混沌区域, 当 $r=4$ 时系统处于完全混沌状态,

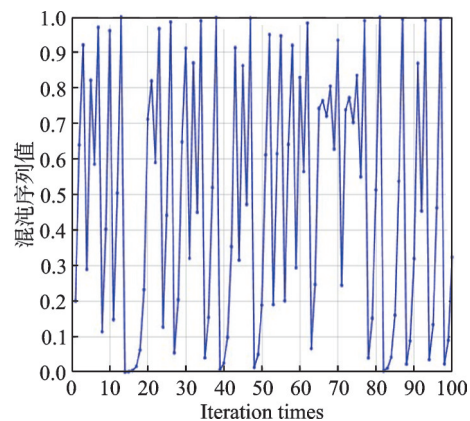


图1 时间序列图像

Fig.1 Time series image

最终的长期行为会在 $[0,1]$ 区间上均匀分布。故本文采用 $r=4$,初始值 X_1 为 $[0,1]$ 内的随机数。图1和2分别展示了随迭代进行所生成点的时间演化过程以及各点在迭代时, X_n 与 X_{n+1} 在 $[0,1]$ 中的分布情况。相较于随机种群初始化方式提高了种群的初始多样性和分布均匀性,有助于扩大初始搜索范围。

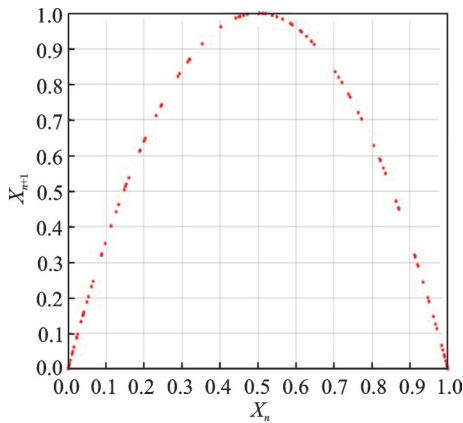


图2 空间关系图

Fig.2 Spatial relationship

在算法的实现中,首先,进行20次预迭代以避免混沌序列的瞬态过程导致混沌序列生成没有混沌特性;其次,将稳定的混沌序列生成在 $[0,1]$ 区间上;最后,将所生成的序列映射到搜索空间中作为初始种群。映射关系如下

$$X_i^k = l_{b_k} + (u_{b_k} - l_{b_k}) \cdot X_n \quad (8)$$

式中: X_i^k 表示第 i 个个体在第 k 维的位置; l_{b_k} 、 u_{b_k} 分别为第 k 维的搜索上下界。

2.2 螺旋搜索策略

螺旋搜索是一种模拟自然界中生物螺旋运动的搜索策略^[14],它通过在搜索空间中生成螺旋轨迹来探索潜在的最优解。该策略具有较强的局部搜索能力和较高的搜索精度,能够有效地在解空间中进行细致的搜索。并且,螺旋搜索策略可以应用于各种优化算法中,特别是在需要进行局部搜索和提高搜索精度的场景中。

在传统鸽群算法中,第1阶段的位置更新主要依赖于速度的更新,而速度的更新方式缺乏局部精细搜索的能力。因此,在传统鸽群算法的地图操作中引入螺旋搜索策略,可以帮助算法提高局部搜索能力,以弥补其在局部搜索能力上的不足。通过螺旋搜索策略的引入,算法能够在局部区域内进行更加细致和深入的搜索,从而提高算法的收敛速度和解的质量。

首先,改进速度更新方式(式(1))使速度的更

新适应后续改进,同时增强了算法在搜索后期的收敛能力,其表达式如下

$$V_{\text{new}}^t = V_i^t \cdot (1 - e^{-\lambda t}) + (X_{\text{best}}^t - X_i^t) \cdot \text{rand} \quad (9)$$

同时,改进后的位置更新公式为

$$X_{\text{new}}^t = X_i^t + V_i^t + S_f \quad (10)$$

$$S_f = (y - x) \cdot \text{rand} \quad (11)$$

$$x = R \cdot \sin \varphi \quad (12)$$

$$y = R \cdot \cos \varphi \quad (13)$$

式中: S_f 表示螺旋因子,通过结合螺旋轨迹的坐标差和随机数来增加搜索的随机性和多样性; x 表示计算螺旋轨迹在 x 轴上的坐标值; y 表示计算螺旋轨迹在 y 轴上的坐标值。

$$\varphi = -\rho \cdot t \cdot t_0 + \varphi_0 \quad (14)$$

$$R = r_0 + u + t_0 \quad (15)$$

式中: φ 表示计算螺旋轨迹的角度; t 为当前迭代次数; R 表示计算螺旋轨迹的半径; t_0 表示搜索空间的维度,用于计算螺旋轨迹的各个维度上的值; ρ 和 u 为固定值,分别为0.056 5和0.005; φ_0 表示螺旋轨迹的起始角度,本文取 1.5π ; r_0 为螺旋轨迹的起始半径,大小取值范围为 $[0,1]$,本文取0.5。

通过图3可以看出基于上述公式对速度计算后,将螺旋因子作为局部算子引入到位置更新公式中,使得算法在搜索过程中搜索性能更高,降低算法陷入局部最优解的可能性。同时,通过螺旋因子计算能充分利用速度参数信息,充分挖掘可能产生最优解的区域,从而提高了算法的局部开发能力和解的质量。

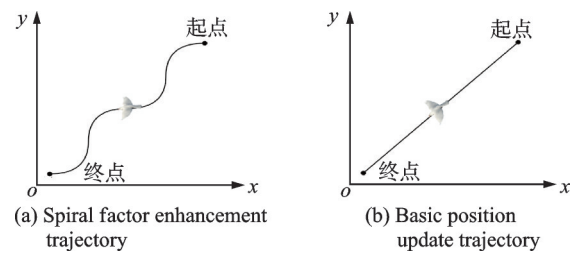


图3 不同搜索方式轨迹

Fig.3 Trajectories of different search methods

2.3 动态反向学习策略

为了提高鸽群算法的全局搜索能力,通过生成反向点来提高算法的探索能力和搜索效率。

动态反向学习策略^[15]是基于反向学习^[16]思想发展而来的。根据Tizhoosh^[16]首次提出的智能算法优化方案,通过在搜索空间中生成反向点来增加种群的多样性,从而提高算法的全局搜索能力。动态反向学习原始解与反向解的生成关系如图4所示,其演示了算法在迭代过程中生成反向解的动态

区域。

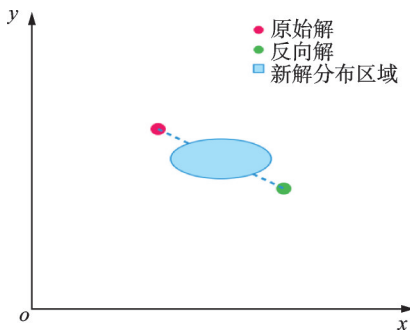


图4 反向解生成区域

Fig.4 Reverse solution generation region

动态反向学习策略的反向解生成公式为

$$X^o = l_{b_s} + u_{b_s} - X \quad (16)$$

$$X^{Do} = X + \omega \cdot \text{rand} \cdot (\text{rand} \cdot X^o - X) \quad (17)$$

式中: X^o 表示传统反向解; X^{Do} 表示动态反向解的坐标; X 表示原始解的坐标; ω 表示权重因子, 用来控制反向解生成的强度。

本文所设定动态权重因子 ω 如式(18)所示, 该动态权重因子使得算法在早期能够增强全局探索, 后期侧重局部开发, 所得到的图像如图5所示。

$$\omega = 0.5 \cdot \left(1 + \cos \left(t \cdot \frac{\pi}{I_{\text{MaxIter}}} \right) \right)^2 \quad (18)$$

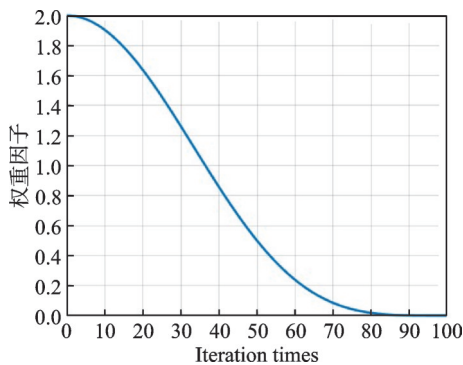


图5 动态权重因子函数曲线

Fig.5 Dynamic weight factor function curve

在实际算法实现中, 动态反向学习策略被应用于算法的两个主要阶段。在地图操作阶段和地标操作阶段中, 每当个体位置完成更新后, 算法会立即为其生成对应的动态反向解坐标。通过贪婪选择策略, 对原始解和动态反向解进行适应度评估, 保留适应度更优的位置作为个体的新位置; 同时根据式(17)计算新位置的反向速度。这种更新机制显著增强了算法的自适应能力和鲁棒性, 使种群能够更灵活地响应搜索空间的变化。同时, 由于速度影响着后续的位置更新, 这种机制也为搜索过程引入了额外的动态性。

2.4 地标阶段改进

在传统鸽群优化算法的地标阶段, 传统方法是按照适应度值对种群进行排序, 仅选择适应度较高的前一半个体计算质心位置并更新这些个体, 而直接抛弃后一半的个体。这种方法导致种群数量不断减少, 难以维持种群多样性。针对这一问题, 本文提出改进方案: 在计算出质心位置后利用该位置信息对整个种群进行动态更新, 即每个个体都根据质心位置和自身位置进行位置调整, 即对式(5,6)进行调整得到式(19,20)。

$$S_{\text{XF}} = \sum_{i=1}^N (X_i' \cdot F(X_i)) \quad (19)$$

$$S_{\text{NF}} = N \cdot \sum_{i=1}^{\frac{N}{2}} F(X_i) \quad (20)$$

该机制避免种群规模缩减, 通过全局信息共享提升收敛稳定性, 让所有个体都有机会向更优位置移动来维持种群多样性, 有助于提高算法的全局搜索能力。

2.5 算法流程

本文改进算法具体流程如下:

(1) 初始化各参数。设置迭代次数 $t=0$, 种群数量 N , 最大迭代次数 I_{MaxIter} 以及阶段转换因子 R_{NcRate} , 定义鸽群算法以及螺旋搜索和反向学习参数。

(2) 生成初始鸽群。在解空间中生成混沌序列, 通过式(8)映射到上下界内作为初始鸽群。计算适应度将初始鸽群适应度排序并记录当前最优解。

(3) 执行指南针和地图操作: 当迭代次数在 $[0, I_{\text{MaxIter}} \cdot R_{\text{NcRate}}]$ 时, 计算局部算子(螺旋因子与动态反向学习反向位置)分别用于各个鸽子位置更新以及学习反向位置。

(4) 重新排序更新当前迭代次数的全局最优解。迭代次数 $t=t+1$, 当前迭代次数若未超过 $I_{\text{MaxIter}} \cdot R_{\text{NcRate}}$ 则执行步骤(3)。

(5) 执行地标操作: 当迭代次数超过 $I_{\text{MaxIter}} \cdot R_{\text{NcRate}}$ 时, 对所有鸽子进行排序, 计算前一半鸽群的质心位置后按式(19,20)对所有鸽子进行位置更新并计算反向位置。

(6) 所有鸽子按照适应度值重新进行排序, 更新全局最优解。迭代次数 $t=t+1$, 当前迭代次数若未达到 I_{MaxIter} 则执行步骤(5)。

(7) 迭代次数达到最大迭代次数则输出最终全局最优解。

鸽群算法与改进鸽群算法的流程图分别如图6和7所示。

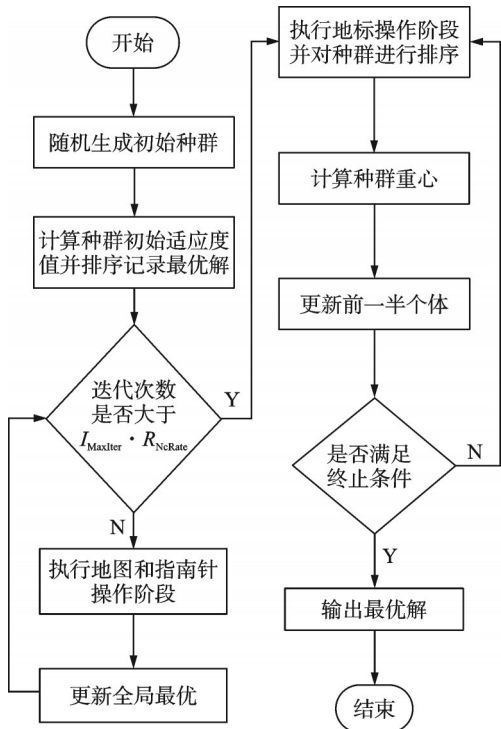


图 6 PIO 算法流程图

Fig.6 Flow chart of PIO algorithm

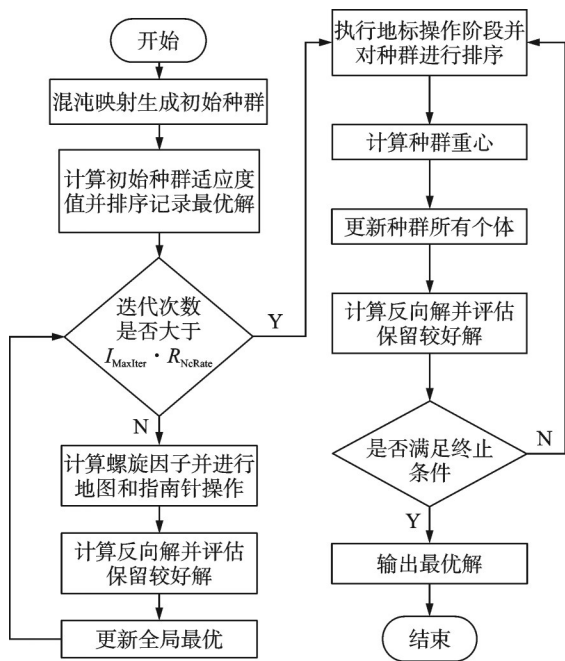


图 7 MSDPIO 算法流程图

Fig.7 Flow chart of MSDPIO algorithm

3 改进算法性能测试与分析

3.1 实验环境配置与测试函数

实验所采用的相关计算机配置为:CPU 为 Intel(R) Core(TM) i5-12600KF, 计算机主频率为 3.70 GHz, 16 G 内存, 64 位 Win10 操作系统;所用软件为 Matlab2024a。为了验证改进鸽群算法的有效性, 本文选取对比算法为 PIO 算法^[5]、AO 算

法^[3]、灰狼算法 (Grey wolf optimizer, GWO)^[17]、DBO 算法^[4]、加入螺旋搜索的鸽群 (Spiral pigeon-inspired optimization, SPIO) 算法和加入动态反向学习的鸽群 (Dynamic pigeon inspired optimization, DPIO) 算法。

选择以上算法原因包括以下几点: 首先, 进行基础算法的对比, 如 PIO、AO、GWO、DBO 等均为近年提出具有代表性的智能优化算法, 且广泛用于机器人路径规划领域; 其次, 加入改进策略逐步验证, 通过对比 PIO、SPIO 和 DPIO 算法, 可逐步验证螺旋搜索和动态反向学习策略对算法性能的独立贡献, 为最终 MSDPIO 算法提供理论依据。最后, 考虑算法特性覆盖, 所选算法涵盖不同搜索机制, 能够全面反映改进算法在不同搜索模式下的竞争力。为保证实验的公平有效, 所有算法的种群规模和迭代次数分别设置为 30 与 500, 重复运行 10 次, 各算法参数也严格与其原文章相对应。

CEC2017 基本测试^[18]函数是一组用于评估优化算法性能的标准化基准函数, 这些测试函数被广泛用于评估优化算法的性能, 比较不同算法在各种问题类型上的表现。函数分为以下 3 类: 单峰函数 (Unimodal functions, UF) 包括 $F_1 \sim F_3$, 简单多峰函数 (Simple multimodal functions, SMF) 包括 $F_4 \sim F_{10}$, 混合函数 (Hybrid functions, HF) 包括 $F_{11} \sim F_{20}$, 组合函数 (Composition functions, CF) 包括 $F_{21} \sim F_{30}$ 。

本文所选 10 个测试函数如表 1 所示, 所选函数求解维度均为 10 维, 包含 UF、SMF、HF 及 CF。所选具体函数及其函数特征和函数求解适应度最优值如表 1 所示。

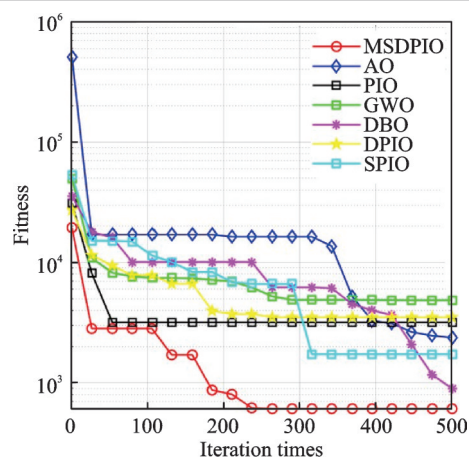
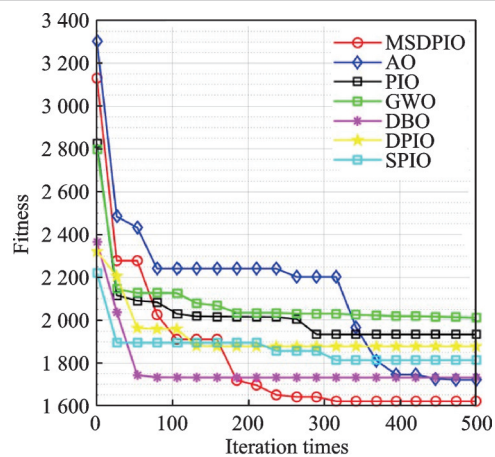
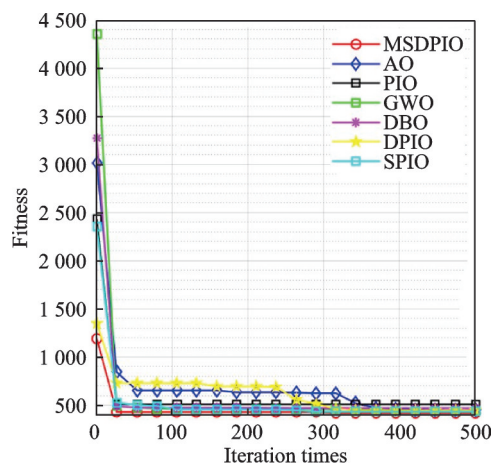
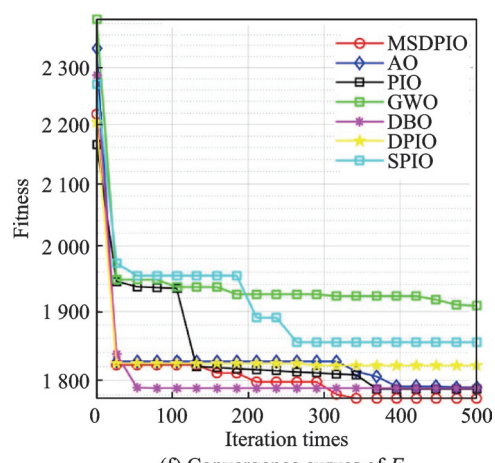
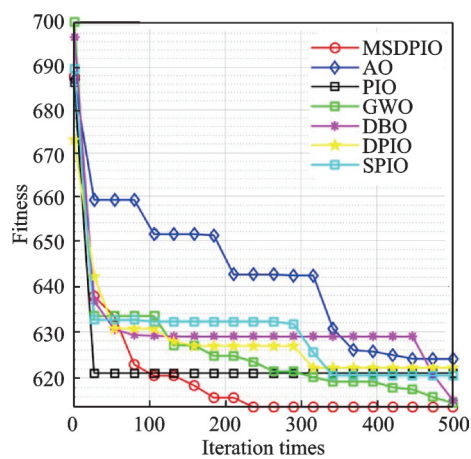
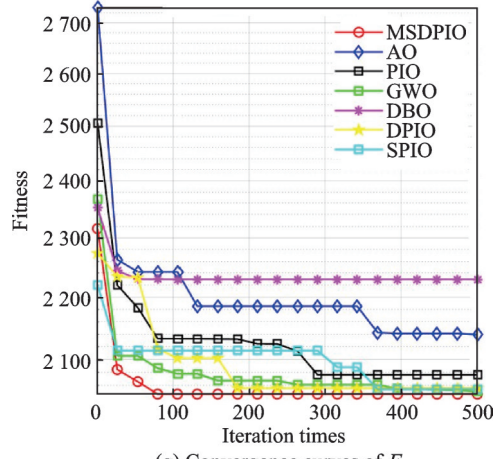
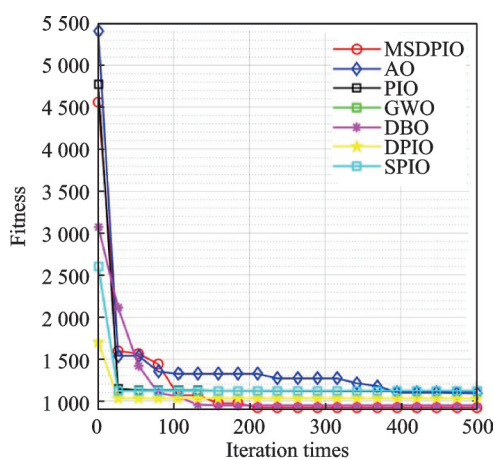
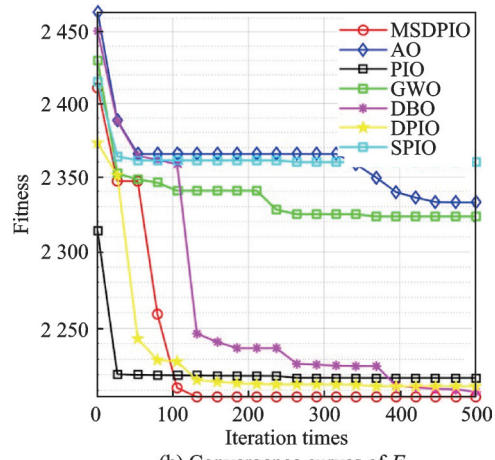
表 1 所选函数相关信息

Table 1 Selected function related information

所选函数	函数特征	最优值
F_3	UF	300
F_4	SMF	400
F_6	SMF	600
F_9	SMF	900
F_{16}	HF	1 600
F_{17}	HF	1 700
F_{20}	HF	2 000
F_{21}	CF	2 100
F_{26}	CF	2 600
F_{29}	CF	2 900

3.2 实验结果分析

实验结果如图 8 与表 2 所示。图 8 中横坐标表示迭代次数, 纵坐标表示最优适应度值,

(a) Convergence curves of F_3 (e) Convergence curves of F_{16} (b) Convergence curves of F_4 (f) Convergence curves of F_{17} (c) Convergence curves of F_6 (g) Convergence curves of F_{20} (d) Convergence curves of F_9 (h) Convergence curves of F_{21}

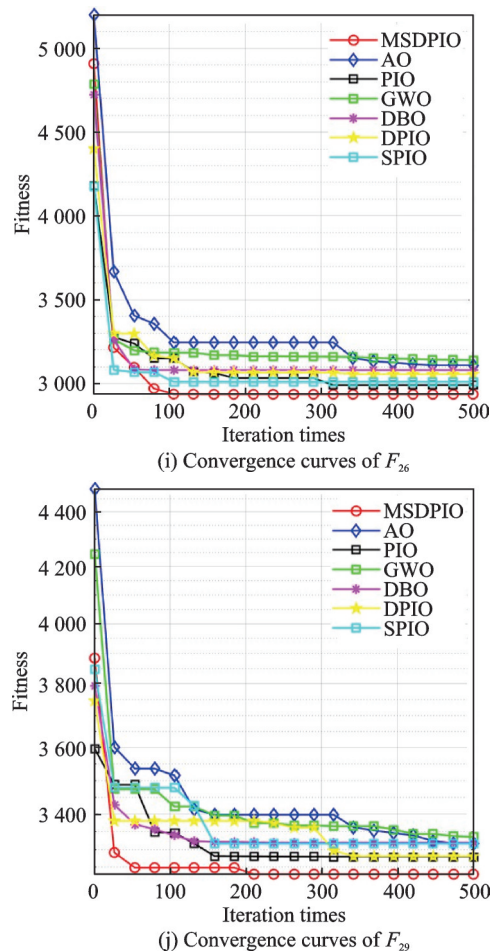


图 8 各函数收敛曲线

Fig. 8 Convergence curves of each function

表 2 中分别记录不同算法在各函数测试中的平均适应度值和标准差,用于评估不同算法的性能。

其中,MSDPIO 在所有测试场景中均展现出显著优势。在 $UF(F_3)$ 与 $SMF(F_4 \sim F_9)$ 中,PIO、GWO 等算法因早熟收敛(如 F_3 中 PIO 平均适应度 $4.17E+03$ 与 MSDPIO 的 $6.65E+02$)或探索效率不足(如 F_6 中 GWO 未完全收敛)难以逼近理论最优解,而 MSDPIO 通过 Logistic 混沌初始化扩大搜索范围,显著提升收敛精度(F_9 平均适应度 $9.29E+02$,较 PIO 优化 20.6%)。在混合函数($F_{16} \sim F_{20}$)测试中,在 F_{20} 中 MSDPIO 平均适应度值($2.08E+03$)与标准差($1.52E+01$)均优于对比算法,动态反向学习策略通过自适应权重调节,平衡了全局探索与局部开发。组合函数($F_{21} \sim F_{29}$)中,MSDPIO 算法在 F_{26} 的适应度($3.00E+03$)较 PIO 算法提升了 7.7%,其螺旋搜索策略增强了复杂问题的求解能力。

此外,MSDPIO 算法在所有函数中标准差最低(如 F_6 中 $3.00E+00$ 与 AO 的 $7.82E+00$),验证了多策略融合下算法对鲁棒性的提升。与 SPIO 和 DPIO 算法相比,MSDPIO 算法在 F_9 中平均适应度较 SPIO 算法降低 22.6%,证明单一策略改进效果有限,而螺旋搜索、动态反向学习与混沌初始化的协同作用实现了性能突破。

表 2 各算法优化结果对比

Table 2 Comparison of optimization results of each algorithm

函数	指标	MSDPIO	PIO	AO	GWO	DBO	SPIO	DPIO
F_3	Average	$6.65E+02$	$4.17E+03$	$2.95E+03$	$1.14E+03$	$1.12E+03$	$4.87E+03$	$3.27E+03$
	STD	$2.12E+02$	$1.60E+03$	$1.76E+02$	$2.35E+02$	$1.74E+03$	$2.05E+03$	$1.74E+03$
F_4	Average	$4.15E+02$	$4.56E+02$	$4.21E+02$	$6.19E+02$	$4.25E+02$	$4.39E+02$	$4.46E+02$
	STD	$5.75E+00$	$3.18E+01$	$2.54E+01$	$9.09E+01$	$3.50E+01$	$1.51E+01$	$2.00E+01$
F_6	Average	$6.11E+02$	$6.22E+02$	$6.23E+02$	$6.59E+02$	$6.13E+02$	$6.24E+02$	$6.20E+02$
	STD	$3.00E+00$	$5.83E+00$	$7.82E+00$	$1.33E+01$	$7.98E+00$	$5.45E+00$	$3.87E+00$
F_9	Average	$9.29E+02$	$1.17E+03$	$1.01E+03$	$3.35E+03$	$9.64E+02$	$1.20E+03$	$1.08E+03$
	STD	$2.21E+01$	$9.66E+01$	$5.97E+01$	$1.27E+03$	$3.94E+01$	$1.31E+02$	$9.07E+01$
F_{16}	Average	$1.71E+03$	$1.83E+03$	$1.82E+03$	$2.25E+03$	$1.84E+03$	$1.80E+03$	$1.80E+03$
	STD	$7.75E+01$	$1.26E+02$	$1.22E+02$	$1.33E+02$	$1.74E+02$	$1.07E+02$	$8.13E+01$
F_{17}	Average	$1.77E+03$	$1.80E+03$	$1.79E+03$	$1.99E+03$	$1.79E+03$	$1.82E+03$	$1.79E+03$
	STD	$1.37E+01$	$2.60E+01$	$2.74E+01$	$9.43E+01$	$4.71E+01$	$2.54E+01$	$1.41E+01$
F_{20}	Average	$2.08E+03$	$2.10E+03$	$2.14E+03$	$2.27E+03$	$2.10E+03$	$2.10E+03$	$2.09E+03$
	STD	$1.52E+01$	$2.79E+01$	$5.20E+01$	$8.31E+01$	$7.05E+01$	$1.18E+01$	$1.77E+01$
F_{21}	Average	$2.20E+03$	$2.33E+03$	$2.29E+03$	$2.39E+03$	$2.26E+03$	$2.30E+03$	$2.21E+03$
	STD	$1.71E+00$	$5.84E+01$	$5.23E+01$	$1.70E+01$	$6.78E+01$	$7.38E+01$	$2.91E+00$
F_{26}	Average	$3.00E+03$	$3.25E+03$	$3.08E+03$	$4.14E+03$	$3.13E+03$	$3.06E+03$	$2.97E+03$
	STD	$2.90E+01$	$5.08E+02$	$2.39E+02$	$5.63E+02$	$1.79E+02$	$7.61E+01$	$2.52E+01$
F_{29}	Average	$3.23E+03$	$3.24E+03$	$3.27E+03$	$3.44E+03$	$3.30E+03$	$3.30E+03$	$3.03E+03$
	STD	$2.46E+01$	$3.23E+01$	$5.30E+01$	$1.26E+02$	$8.88E+01$	$7.11E+01$	$3.83E+01$

4 机器人路径规划仿真实验

机器人路径规划,因其极高的应用价值在越来越多的领域被学者们所重视,为了进一步突出本文改进算法的实用性,本节将利用一种基于改进B样条曲线^[19]的机器人路径规划方法通过不同环境进行仿真分析。该方法通过优化中间控制点的位置,生成满足运动约束的平滑路径,与前文所述改进策略协同控制路径点生成,提高路径的质量和可行性。

4.1 环境建模

本节通过建立不同大小的环境模型进行实验仿真分析。分别建立两种不同复杂度的地图模型。其中,图9(a)所示为20 m×20 m的小地图模型,地图小且复杂度低;而图9(b)所示为40 m×40 m的大地图模型,其地图大、障碍物较多、复杂度高。同时,起点和终点,左下角蓝色小正方形以及右上角红色小正方形分别代表起点和终点,黑色轮廓圆圈表示障碍物,其余位置皆为可行区域。本文所用MSDPIO算法的各参数设置如表3所示,其中 r 为混沌映射中控制系数。

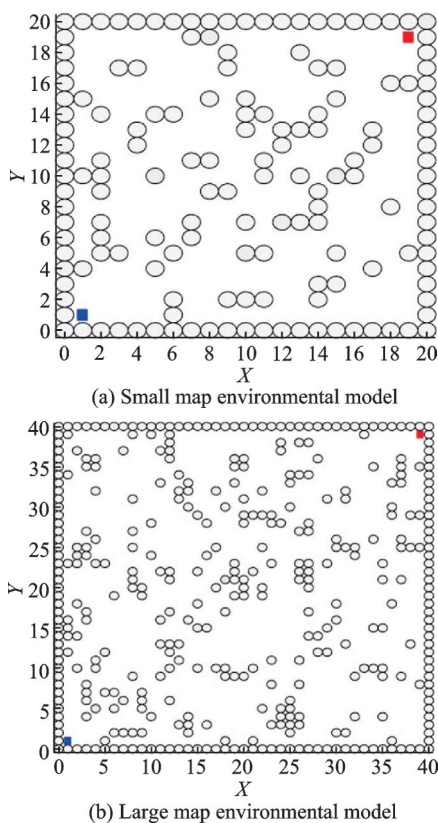


图9 Different environmental map models

4.2 基于改进策略的B样条

将起点坐标、终点坐标以及 N_1 个中间控制点作为基本路径描述单元。每个控制点的位置都作

表3 MSDPIO 算法参数设置

Table 3 MSDPIO algorithm parameter settings

参数	数值
r	4
φ_0/rad	1.5π
r_0/m	0.5
λ	0.2

为优化变量,通过优化算法进行调整。为了保证路径的平滑性,采用改进三次B样条曲线对控制点序列进行插值(插值采样点本文设置为500),并与前文所提改进策略融合,确保了路径能够精确通过起终点。

本节基于改进B样条曲线生成机器人路径,通过以下4方面优化实现平滑性与避障能力的协同提升。

4.2.1 动态节点向量生成

传统B样条采用均匀节点向量,难以适应复杂环境下的路径控制点分布。本文提出动态节点向量生成方法:采用基于控制点分布的改进节点向量,其数学表达式为

$$U_i = [0, t_1, t_2, \dots, t_{N-3}, 1] \quad (21)$$

$$t_k = \sum_{i=1}^{N_1} \|P_i - P_{i-1}\| / \sum_{i=1}^n \|P_i - P_{i-1}\| \quad (22)$$

式中: P_i 为第 i 个控制点的坐标; n 为控制点间插值点数量,取值为采样点数量减1。

通过控制点间距比例调整节点向量,与混沌初始化策略协同;Logistic映射生成均匀分布的控制点(式(7~8)),确保 t_k 动态反映路径的空间分布特性,避免传统均匀分布节点导致路径扭曲。

4.2.2 主方向投影优化

对于中间控制点,为增强路径方向性与平滑性,引入主方向投影修正控制点位置

$$d = \frac{P_e - P_s}{\|P_e - P_s\|} \quad (23)$$

$$P'_i = P_s + (d \cdot (P_i - P_s)) \cdot d + (P_i - P_r) \quad (24)$$

式中: P_r 为点在主方向上的投影; P_e 、 P_s 分别为终点和起点控制点。

与螺旋搜索策略协同:螺旋因子(式(11~15))在控制点更新中引入路径沿主方向螺旋延伸,避免冗余迂回。

4.2.3 局部平滑处理

为进一步增强B样条曲线对路径的平滑能力,对优化后的控制点进行加权平滑处理即计算前后控制点的中点使当前点向中点细微调整

$$P_i^{\text{new}} = \beta P_i + (1 - \beta) \cdot \left(\frac{P_{i-1} + P_{i+1}}{2} \right) \quad (25)$$

式中 $\beta=0.7$ 表示平滑权重系数。

通过局部平滑处理与动态反向学习策略协同,通过自适应权重(式(18))动态调整反向解的生成范围,结合局部平滑权重(式(25)),在避免急转弯的同时,维持路径对动态障碍的响应灵活性。

4.2.4 改进B样条路径生成

综合上述改进,路径点计算公式为

$$P(u) = \sum_{i=0}^n N_{i,3}(u, U_i) P_i' \quad (26)$$

式中: $N_{i,3}(u, U_i)$ 为三次B样条基函数通过递归定义; $P(u)$ 为参数 u 处的路径点。

最终路径由式(26)生成,融合混沌初始化、螺旋搜索与动态反向学习的协同优化,实现平滑、最短路径与避障能力的多目标平滑。

4.3 路径长度与违反程度评估

路径评估采用多目标优化策略,综合考虑路径长度和约束违反程度。

在路径长度计算中,若对采样点进行积分计算可以得到较为精确的距离,但其计算量较大所需时间较长且计算结果误差与计算两点之间欧氏距离相差可忽略不计。故本文在尽量消除计算误差并使时间效率更高的前提下,采用基于起终点之间的直线距离计算基准长度,将实际路径长度归一化后与基准长度比较,作为路径长度评价指标。基准长度以及路径长度计算公式为

$$L_{\text{base}} = \sqrt{(X_e - X_s)^2 + (Y_e - Y_s)^2} \quad (27)$$

$$L = \sum_{i=1}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (28)$$

$$D = L/L_{\text{base}} \quad (29)$$

式中: X_e, X_s, Y_e, Y_s 分别表示终点和起点的X轴坐标和Y轴坐标; (x_i, y_i) 表示路径上第 i 个采样点的坐标; (x_{i-1}, y_{i-1}) 表示路径上第 $i-1$ 个采样点的坐标; D 表示计算最终适应度的路径长度。

其次,通过计算路径与障碍物之间的最小距离,评估路径的约束违反程度。路径违反程度计算公式为

$$V = \max\left(\frac{1 - d'}{R_e}, 0\right) \quad (30)$$

式中: $\max$ 表示取最大值, d' 表示中间采样点到每个障碍物中心的坐标, R_e 为障碍物的半径大小。

算法采用软约束方法处理障碍物避障问题即通过计算路径点与障碍物的最小距离,构建基于距离的惩罚函数。当路径点进入障碍物的影响范围时,相应的惩罚值将增加,从而引导优化算法搜索可行解。这种处理方式既保证了约束的满足,又避免了硬约束可能带来的计算困难。适应度计算公式为

式为

$$\text{Fitness} = A \cdot D + B \cdot V \quad (31)$$

式中 A, B 分别表示权重系数,可根据实际需要自由定义。

4.4 不同环境下实验结果及分析

接下来本文将通过两种不同大小的地图对MSDPIO算法进行仿真实验以验证算法的优势,为保证实验的公平有效,所有算法在小环境中种群规模为50,大环境中种群规模为80,迭代次数均为100次,实验相关配置如3.1节所述。两种地图模型分别重复运行10次。选择若干算法进行对比,包括多策略融合的天鹰优化(Multi-strategy improved aquila optimizer, MSIAO)算法^[20],AO算法^[3],PIO算法^[5],GWO算法^[17]和DBO算法^[4]。

与3.1节所选算法不同,调整的原因包括:首先,MSIAO算法是近年来针对复杂路径规划问题提出的改进算法,其多策略融合思想与本文MSDPIO算法的设计目标一致,引入MSIAO算法可验证MSDPIO算法与同类多策略改进算法的性能差异,体现本文改进方法的通用性和竞争力。其次,保留PIO、AO、GWO和DBO算法作为对比组,可确保实验结果的连贯性,同时验证改进算法在不同问题场景(基准函数测试与实际路径规划)中的鲁棒性。最后,SPIO和DPIO算法在第3节中已用于验证局部改进策略的有效性,而本节聚焦于多策略融合后的综合性能。为避免冗余,未将SPIO和DPIO算法纳入路径规划仿真实验,但通过MSDPIO算法与原始PIO算法的对比,仍能体现多策略融合的整体优势。并且所选算法满足算法收敛速度、避障能力等特殊要求。

在难度较小的地图中,中间控制节点个数为 $N_1=3$,在难度较大的地图模型中,中间控制节点个数为 $N_1=4$ 。各算法最优路径生成结果和平均适应度迭代曲线如图10所示。

在仿真实验中,通过对不同地图模型下的实验结果进行结果分析,本文所提改进策略相较于其他优化算法寻优能力更佳,收敛效率显著提升;各算法路径规划最终数据如表4、5所示。

在表4、5中,平均适应度值为算法在10次独立运行中的平均值,反映算法的综合优化能力,即路径长度与避障性能的加权和;最短路径长度为10次运行中生成的归一化后的最短路径长度,体现算法的最优路径搜索能力。平均适应度改进率为MSDPIO算法较对比算法的适应度值提升百分比,路径长度改进率同理。

小地图测试中,MSDPIO算法的平均收敛适应度值为28.354,较PIO算法的平均适应度值28.693

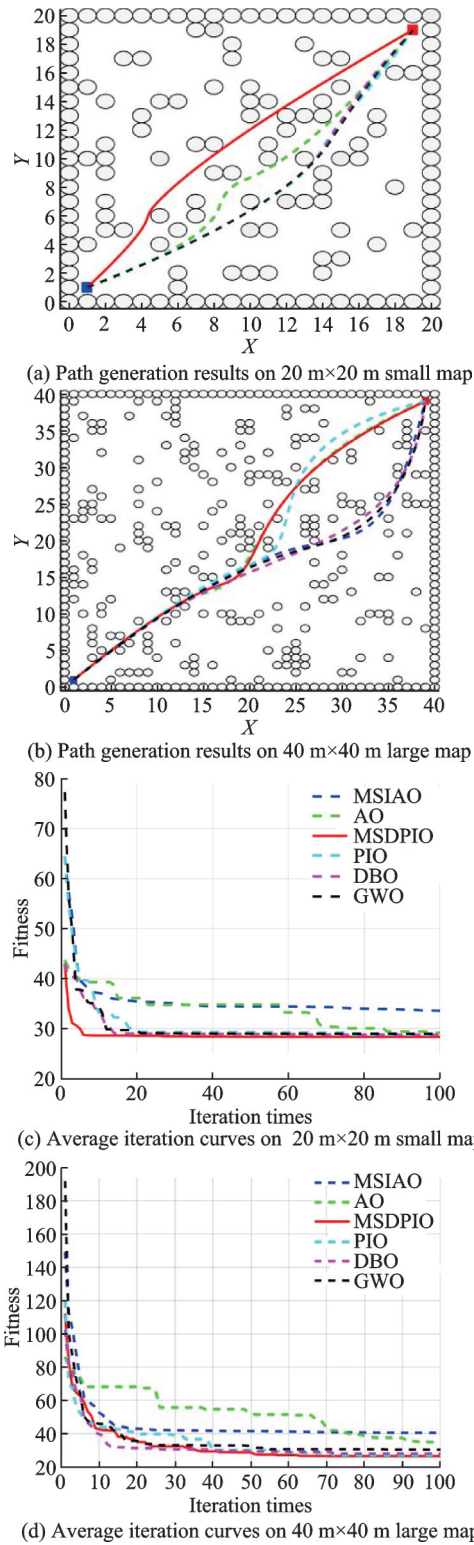


图 10 不同地图运行结果

Fig.10 Different map operation results

提升 1.18%，对比 AO 算法 (27.029 m) 最短路径长度 (25.767 m) 缩短 4.67%。大地图测试中，对比 MSIAO 算法，其平均适应度改进率为 37.82%，最短路径长度 (54.921 m) 相比于 GWO 算法 (56.661 m) 提升了 3.07%。上述改进率结果表明，MSDPPIO 算法在混沌初始化、螺旋搜索与动态反向学习等协同作用下，在高维复杂问题中仍能保持稳定的收敛性能。

表 4 小地图路径规划数据

Table 4 Small map path planning data

算法	平均收敛 适应度	最短路径 长度/m	平均收敛适应 度改进率/%	路径长度 改进率/%
MSDPPIO	28.354	25.767	—	—
PIO	28.693	26.149	1.17	1.46
AO	29.157	27.029	2.75	4.67
MSIAO	33.582	26.141	15.57	1.43
DBO	28.763	26.149	1.42	1.46
GWO	28.968	26.151	2.12	1.47

表 5 大地图路径规划数据

Table 5 Large map path planning data

算法	平均收敛 适应度	最短路径 长度/m	平均收敛适应 度改进率/%	路径长度 改进率/%
MSDPPIO	25.319	54.921	—	—
PIO	27.131	56.092	6.68	2.09
AO	35.019	55.504	27.70	1.05
MSIAO	40.721	56.681	37.82	3.11
DBO	28.121	56.091	9.96	2.09
GWO	30.576	56.661	17.19	3.07

5 结 论

本文针对传统鸽群优化算法易陷入局部最优、收敛效率低的问题，提出 MSDPIO 算法。通过 Logistic 混沌初始化 (提升种群分布均匀性)、螺旋搜索策略 (增强局部开发能力) 与动态反向学习 (自适应平衡探索与开发) 的多策略协同优化，结合地标阶段全局更新机制 (维持种群多样性)，显著提升了算法全局性能。实验表明，MSDPPIO 算法在 CEC2017 基准测试中收敛精度较 PIO 算法均有较大提升，在机器人路径规划中，融合改进 B 样条曲线和 MSDPIO 算法生成的路径长度较 PIO、AO 等算法缩短 1.43%~4.67%，收敛速度最大可提升 37.82%，验证了其在复杂场景下的高效性与实用性。

参考文献:

[1] CAI Z Z, SELEZNEVA S M, YANG M. Dijkstra algorithm based minimum acceleration/snap quadrotor UAV trajectory planning[J]. Journal of Physics Conference Series, 2024, 2746(1): 1-6.

[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 Guo, WU Xiao, XIAO Ruqi, 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] LAITH A, DALIA Y, MOHAMED E A, et al. Aquila optimizer: A novel meta-heuristic optimization algorithm[J]. Computers & Industrial Engineering, 2021, 157: 107250.
- [4] XUE J K, BO S. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization[J]. The Journal of Supercomputing, 2022, 79(7): 7305-7336.
- [5] DUAN H B, QIAO P X. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning[J]. International Journal of Intelligent Computing and Cybernetics, 2014, 7(1): 24-37.
- [6] 李兆博,孙双蕾.基于鸽群优化改进动态窗的多无人车协同编队避障控制[J].工程科学学报,2024,46(7): 1279-1285.
- LI Zhaobo, SUN Shuanglei. Multi-vehicle formation and obstacle avoidance control based on pigeon-inspired optimization and dynamic window approach[J]. Chinese Journal of Engineering, 2024, 46(7): 1279-1285.
- [7] 秦昌礼,张华强,刘林,等.一种基于改进 PSO-PIO 算法的 AGV 路径规划方法[J].哈尔滨理工大学学报,2022,27(3): 82-89.
- QIN Changli, ZHANG Huaqiang, LIU Lin, et al. A path planning method for AGV based on improved PSO-PIO algorithm[J]. Journal of Harbin University of Science and Technology, 2022, 27(3): 82-89.
- [8] 刘昂,蒋近,徐克锋.改进蚁群和鸽群算法的机器人路径规划[J].计算机应用,2020,40(11): 3366-3372.
- LIU Ang, JIANG Jin, XU Kefeng. Robot path planning based on improved ant colony and pigeon-inspired optimization algorithm[J]. Journal of Computer Applications, 2020, 40(11): 3366-3372.
- [9] ALAZZAM H, HARIEH A, SABRI E K. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer[J]. Expert Systems with Applications, 2020, 148: 113249.
- [10] GAABOURI E I, SENHADJI M, BELKASMI M, et al. A new S-box pattern generation based on chaotic enhanced logistic map: Case of 5-bit S-box[J]. Cybersecurity, 2024, 7(1): 59.
- [11] 任庆欣,冯锋.多策略融合改进的斑马优化算法[J].计算机科学,2024,51(S2): 56-62.
- REN Qingxin, FENG Feng. Zebra optimization algorithm improved by multi-strategy fusion[J]. Computer Science, 2024, 51(S2): 56-62.
- [12] 李书辞,岳树岭.求解无人机三维航迹规划问题的改进 SO 算法[J].机械设计与研究,2024,40(6): 296-302.
- LI Shuci, YUE Shuling. Improved snake swarm optimizer for UAV 3D path planning problem[J]. Machine Design and Research, 2024, 40(6): 296-302.
- [13] 王海群,郭庆通,葛超.基于改进蛱蝶优化算法的搬运机器人路径规划[J].现代制造工程,2024(11): 87-95.
- WANG Haiqun, GUO Qingtong, GE Chao. Path planning of handling robot based on improved dung beetle optimization algorithm[J]. Modern Manufacturing Engineering, 2024(11): 87-95.
- [14] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [15] XU Y, YANG Z, LI X, et al. Dynamic opposite learning enhanced teaching-learning-based optimization [J]. Knowledge-Based Systems, 2019, 188(C): 104966.
- [16] TIZHOOSH R H. Opposition-based reinforcement learning[J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2006, 10(4): 578-585.
- [17] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(3): 46-61.
- [18] LUAN M T, KHATIR S, TRAN T M, et al. Exponential-trigonometric optimization algorithm for solving complicated engineering problems[J]. Computer Methods in Applied Mechanics and Engineering, 2024, 432 (PB): 117411.
- [19] 陈丽芳,杨火根,陈智超,等.B样条技术与遗传算法融合的全局路径规划[J].浙江大学学报(工学版),2024,58(12): 2520-2530.
- CHEN Lifang, YANG Huogen, CHEN Zhichao, et al. Global path planning by fusing B-spline technique and genetic algorithm[J]. Journal of Zhejiang University (Engineering Science), 2024, 58(12): 2520-2530.
- [20] 吴素谦,闫建国,杨斌,等.多策略改进的天鹰优化器及其在路径规划中的应用[EB/OL]. [2025-01-02] <http://kns.cnki.net/kcms/detail/51.1307.TP.20240424.1641.008.html>.
- WU Suqian, YAN Jianguo, YANG Bin, et al. Multi-strategy improved aquila optimizer and its application in path planning[EB/OL]. [2025-01-02]. <http://kns.cnki.net/kcms/detail/51.1307.TP.20240424.1641.008.html>.