

DOI: 10.13382/j.jemi.B2508101

基于改进双曲正余弦优化算法的无人机路径协同规划*

张小庆 孙民民 张莉 曾竣哲 宋一佳 李娜

(武汉轻工大学数学与计算机学院 武汉 430023)

摘要:针对双曲正余弦优化算法(SCHO)求解无人机路径协同规划问题时搜索精度差、收敛速度慢及易陷入局部最优的不足,提出基于分段引导与动态分区改进的双曲正余弦优化算法(SDSCHO)。建立了无人机飞行的三维地理模型及威胁条件,构建了综合路径长度、障碍物威胁、飞行高度和转角的路径协同规划代价模型。同时引入混沌 Circle 映射种群初始化、非线性震荡转换因子、分段引导与反向逃逸优化及动态边界分区辅助位置更新策略对 SCHO 算法进行综合改进,并利用改进算法 SDSCHO 求解无人机路径协同规划问题。在多项不同特征的基准函数上与 7 种同类算法进行寻优测试,结果证明 SDSCHO 算法在寻优精度和收敛性能上表现更优。最后,通过搭建不同障碍物的三维山地模型,将 SDSCHO 算法应用于求解无人机单路径、多路径协同规划场景,进一步证实算法处理实际优化问题上的优越性。

关键词: 双曲正余弦优化算法;混沌映射;分段引导;动态分区;路径协同规划

中图分类号: TN91 **文献标识码:** A **国家标准学科分类代码:** 510.80

UAV path collaborative planning based on improved Sinh Cosh optimization algorithm

Zhang Xiaoqing Sun Minmin Zhang Li Zeng Junzhe Song Yijia Li Na

(School of Mathematics & Computer Science, Wuhan Polytechnic University, Wuhan 430023, China)

Abstract: To address the issues of poor search accuracy, slow convergence speed and easy fallback to local optima in solving unmanned aerial vehicle (UAV) path coordination planning problems using the hyperbolic sine-cosine optimization (SCHO) algorithm, a segment-guided and dynamical partitioned improved hyperbolic sine-cosine optimization (SDSCHO) algorithm is proposed. A three-dimensional geographic model of UAV flight and threat conditions is established, and a path coordination planning cost model is constructed that integrates path length, obstacle threat, flight altitude and turning angle. And SCHO algorithm is comprehensively improved by introducing chaos Circle mapping for population initialization, nonlinear oscillation conversion factor, segment-guided and reverse escape optimization, and dynamic boundary partitioning-assisted position update strategy. The improved algorithm SDSCHO is used to solve the UAV path coordination planning problem. On multiple benchmark functions with different characteristics, the optimizing tests are carried out with seven similar algorithms. The results prove that SDSCHO performs better in optimization accuracy and convergence performance. Finally, by building a three-dimensional mountain model with different obstacles, SDSCHO is applied to solve UAV single-path and multi-path coordination planning scenarios, which can further confirm the superiority of our algorithm in handling actual optimization problems.

Keywords: sinh cosh optimizer; chaos map; segment guidance; dynamic partition; path collaborative planning

收稿日期: 2025-01-08 Received Date: 2025-01-08

* 基金项目:湖北省教育厅科技项目(B2020063)、武汉市自然科学基金探索计划(晨光计划)(2024040801020332)、武汉轻工大学研究生课程案例库建设项目(群体智能算法应用案例库)资助

0 引言

随着计算机科学、人工智能、控制理论等多学科发展和交叉融合,无人机技术得到了飞速发展。为了实现无人机的自主导航,路径规划成为关键技术之一。路径规划旨在使无人机在满足约束条件(如避障、能耗、时间等)前提下,找到起点到终点间的最优路径。路径规划通常具有复杂的搜索空间和多种限制条件,传统搜索算法显现出性能劣势。受自然界物种社会行为启发的群智能优化算法因其强大的全局搜索能力备受学者关注,成为解决路径规划的重要方法。该背景下,众多智能算法被提出,如粒子群算法(particle swarm optimization, PSO)^[1]、遗传算法(genetic algorithm, GA)^[2]、鲸鱼优化算法(whale optimization algorithm, WOA)^[3]、阿基米德优化算法(Archimedes optimization algorithm, AOA)^[4]、麻雀搜索算法(sparrow search algorithm, SSA)^[5]、正余弦优化算法(Sine Cosine algorithm, SCA)^[6]、算术优化算法(arithmetic optimization algorithm, AOA)^[7]等。此类算法可分为 4 类:(1)生物进化类算法,如遗传算法和差分进化算法,主要通过模拟生物进化中染色体交叉、变异及选择策略对个体迭代更新;(2)物理或数学类算法,如算术优化算法结合算术运算符加减乘除运算规律搜索最优解;(3)人类活动规律类算法,如头脑风暴优化算法^[8]模拟头脑思考过程;(4)群体智能算法,主要模拟动物群体的觅食及躲避天敌行为,通过个体间协作和竞争实现目标求解,如粒子群算法模拟鸟群捕食行为,鲸鱼算法模拟鲸鱼群的螺旋捕食行为。

种群个体分布、超参数及全局搜索与局部开采均衡性是影响算法搜索精度、收敛速度及陷入局部最优的关键。1)超参数改进,如文献[9]利用可变算子对 PSO 算法的惯性权重和加速因子进行调节,增强算法搜索能力;文献[10]引入分段权值降低 AOA 算法陷入局部最优概率。2)融入变异算子,如文献[11]引入柯西和反向学习对蝗虫算法最优解变异,丰富种群多样性;文献[12]利用定向重组和诱导变异对秃鹰算法个体多维信息重组,提升多样性和个体质量。3)重构种群更新机制,如文献[13]引入局部混沌搜索对解空间扩展,提高阿基德优化算法局部开发能力;文献[14]设计兼顾距离和方向的启发函数,提升无人机航迹规划质量;文献[15]采用二阶段非线性惯性权重均衡全局和局部搜索精度,在勘探阶段引入柯西扰动增强算法跳出局部最优能力。4)多智能算法融合,如文献[16]将鸡群算法的随机跟随引入麻雀算法,优化种群追踪能力;文献[17]将教与学算法引入灰狼算法优化搜索方式,提升收敛能力;文献[18]引入动态拉普拉斯算子对麻雀算法进行改进。

双曲正余弦优化算法(hyperbolic Sinh Cosh optimizer, SCHO)^[19]是模拟双曲正余弦函数变化规律的新型元启发式算法。虽然算法较传统算法具有明显优势,但在处理复杂多维优化问题上依然易出现探采比例不均衡及陷入局部最优的不足。目前仅有文献[20]设计融合算术优化算法的 SCHO 算法 MSCHO,能一定程度提升全局搜索能力。为了提高算法性能,本文设计分段引导和动态分区双曲正余弦优化算法(segment-guidance dynamic-partition hyperbolic Sinh Cosh optimizer, SDSCHO)。在种群多样性、协调搜索与开采过程转换及优化位置更新上实施以下改进:引入混沌 Circle 映射进行种群初始化,提高初始种群多样性;引入非线性震荡转换因子平衡算法的搜索与开发能力;设计分段引导与反向逃逸机制协助算法保留优质个体信息,同时结合动态边界分区辅助位置更新,降低算法陷入局部最优的概率。以此提升算法解决无人机路径协同规划的性能。

1 无人机路径规划模型

1.1 地形模型

地形模型主要包括地形特征、障碍物分布及飞行禁区等信息。地形模型需要精确描述障碍物位置、形状及高度,以便无人机飞行中安全避让。障碍物包括建筑物、树木、不良天气等。飞行禁区是指无人机禁止飞行的区域,如军事禁区、雷达辐射区等。地形模型需要明确标注禁区位置和范围,以确保无人机在飞行过程中不会闯入禁区。

1) 三维地理模型

在复杂环境下,如城市区域、山区等,无人机的飞行面临诸多挑战。三维地理建模可以为无人机提供这些复杂环境的详细地形信息,从而帮助无人机更好地适应这些环境。在山区中,无人机可以根据地形起伏调整飞行高度和速度,确保飞行的稳定性和安全性。本文采用的基准地形模型为:

$$Z(x, y) = \sin(y + a) + b \sin x + c \cos(d\sqrt{x^2 + y^2}) + e \cos y + f \sin(g\sqrt{x^2 + y^2}) \quad (1)$$

式中: x, y 为模型在水平面上的点坐标; Z 为水平面对应的高程值; a, b, c, d, e, f, g 为常数,控制三维地形图中的基准地形起伏。

三维山地环境中自然山体数学模型为:

$$Z(x, y) = \sum_{i=1}^n h_i \exp\left[-\left(\frac{x - x_i}{x_{si}}\right)^2 - \left(\frac{y - y_i}{y_{si}}\right)^2\right] \quad (2)$$

式中: (x_i, y_i) 为山峰 i 的中心坐标; h_i 为地形参数; x_{si} 和 y_{si} 分别为山峰 i 沿 x, y 轴的衰减量控制坡度; n 为山峰总数。以式(1)、(2)可建构三维山地模型如图 1 所示。

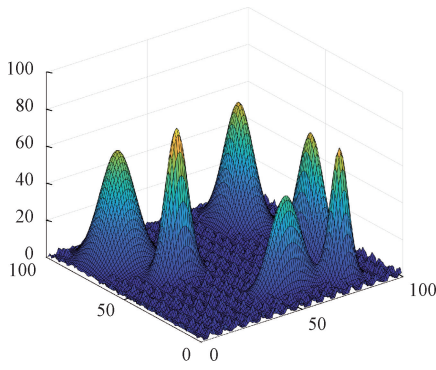


图1 三维山地模型

Fig. 1 3D mountain model

2) 威胁条件

无人机飞行的主要威胁为地面雷达和禁飞区威胁。地面雷达威胁模型用于模拟其对无人机飞行路径的潜在威胁。禁飞区模型主要作用为识别和量化无人机在飞行过程中可能遇到的禁飞区域威胁,以确保无人机能安全有效完成任务。以山地环境为例,无人机在飞行时需要避开高压电杆、恶劣天气区域等。加入威胁条件后的三维山地模型如图2所示。

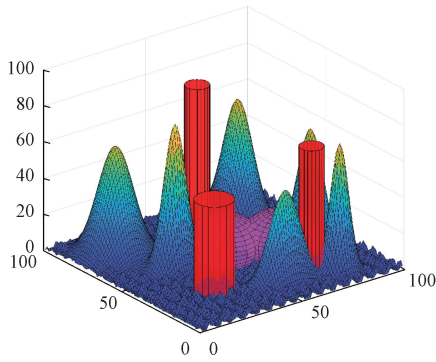


图2 加入障碍物和威胁源的三维山地模型

Fig. 2 3D mountain model with obstacles and threat sources

1.2 无人机路径协同规划代价模型

影响无人机路径规划的主要指标包括路径长度代价、障碍物威胁代价、飞行高度代价和转角代价。

路径长度代价为起点至终点间的距离,定义为:

$$J_{path} = \sum_{i=1}^N L_{path,i} \quad (3)$$

$$L_{path,i} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (4)$$

式中: N 为路径包含的路径点数; x_i, y_i 对应路径点 i 的 x, y 轴坐标; z_i 为路径点 i 的高度; $L_{path,i}$ 是第 i 段路径的长度。

无人机在复杂山地模型中执行飞行任务时,高度过

高会增加燃料成本和危险系数,高度过低会增加与地面、山脉的碰撞概率,保持稳定高度至关重要。定义高度代价模型为:

$$J_{height} = \sqrt{\frac{1}{N} \sum_{i=1}^N (z(i)^2 - z_{avg})^2} \quad (5)$$

$$z_{avg} = \frac{1}{N} \sum_{i=1}^N z(i) \quad (6)$$

式中: $z(i)$ 是路径点 i 的高度; z_{avg} 所有路径点的平均高度。

无人机飞行时需对飞行方向不断调整,其调整幅度的大小会影响无人机飞行的稳定性,其转角代价为:

$$J_{curve} = \begin{cases} \infty, & \theta_{max} < \theta_i \\ \sum_{i=1}^N (\cos\theta_{max} - \cos\theta_i), & \text{其他} \end{cases} \quad (7)$$

$$\cos\theta = \frac{\mathbf{a}_i^T \mathbf{a}_{i+1}}{\|\mathbf{a}_i\| \|\mathbf{a}_{i+1}\|} \quad (8)$$

式中: θ_i 为第 i 个转角; θ_{max} 为预设最大转角约束; \mathbf{a}_i 为整个路径 i 的矢量。

将威胁条件分为地面雷达威胁和禁飞区威胁,以半球体定义地面雷达的威胁范围。则地面雷达威胁代价模型为:

$$J_o = \begin{cases} \infty, & L_i < R_i \\ 0, & \text{其他} \end{cases} \quad (9)$$

$$L_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \quad (10)$$

式中:无人机和第 i 个雷达坐标分别表示为 (x, y, z) 、 (x_i, y_i, z_i) ; R_i 为雷达 i 的探测半径; L_i 为当前无人机与雷达坐标的间距。

利用圆柱包围体定义禁飞区威胁,其代价模型为:

$$J_o = \begin{cases} \infty, & L_i < R_i \text{ and } z \in [0, H_i] \\ 0, & \text{其他} \end{cases} \quad (11)$$

$$L_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (12)$$

式中:无人机和第 i 个禁飞区底部圆心坐标分别为 (x, y, z) 、 (x_i, y_i, z_i) ; R_i 为第 i 个禁飞区的半径; H_i 为第 i 个禁飞区的高度; L_i 为当前无人机与禁飞区底部圆心坐标的间距。

针对多无人机多路线路径协同规划场景,需针对无人机飞行过程中可能出现的相互碰撞进行规避,其代价模型为:

$$J_m = \begin{cases} \infty, & (L_{x,t} > r \text{ or } L_{y,t} > r \text{ or } L_{z,t} > r) \\ 0, & \text{其他} \end{cases} \quad (13)$$

$$\begin{cases} L_{x,t} = |x_{i,t} - x_{i+1,t}| \\ L_{y,t} = |y_{i,t} - y_{i+1,t}| \\ L_{z,t} = |z_{i,t} - z_{i+1,t}| \end{cases} \quad (14)$$

式中: (x_i, y_i, z_i) 、 $(x_{i+1}, y_{i+1}, z_{i+1})$ 对应第 i 个无人机和第 $i+1$ 个无人机的坐标; $L_{x,t}, L_{y,t}, L_{z,t}$ 分别为两架无人机在 t

时刻于 x, y, z 轴上的距离; r 为无人机的质心半径。

根据上述指标建立无人机路径规划代价模型为:

$$\min f = w_1 J_{\text{path}} + w_2 J_{\text{height}} + w_3 J_{\text{curve}} + J_o + J_m \quad (15)$$

式中: f 为总代价; $w_i (i=1, 2, 3)$ 对应代价函数权重, $w_1 + w_2 + w_3 = 1, w_i \geq 0$; $J_{\text{path}}, J_{\text{height}}$ 和 J_{curve} 分别为无人机路径长度代价、高度代价和转角代价; J_o 为威胁代价; J_m 为多无人机路径协同规划代价。

2 SCHO 算法

SCHO 算法包括探索和开发、有界搜索和切换机制 4 个部分组成。

探索和开发过程切换由转换因子 T 决定, 公式为:

$$T = \text{floor}\left(\frac{T_{\max}}{ct}\right) \quad (16)$$

式中: $\text{floor}()$ 为四舍五入函数; ct 为切换点系数, 值为 3.6。

2.1 探索阶段

探索第 1 阶段, 个体围绕原始位置邻近区域及最优解区域搜索, 逐步向最优解靠近, 位置更新方式为:

$$X_{i,j}(t+1) = \begin{cases} X_{\text{best},j} + r_1 \times W_1 \times X_{i,j}(t), r_2 > 0.5 \\ X_{\text{best},j} - r_1 \times W_1 \times X_{i,j}(t), r_2 < 0.5 \end{cases} \quad (17)$$

式中: t 为当前迭代次数; $X_{i,j}(t), X_{i,j}(t+1)$ 分别为 t 次、 $t+1$ 次个体 i 的 j 维位置; $X_{\text{best},j}$ 为最优解位置; r_1, r_2 为 $[0, 1]$ 内随机数; W_1 为 $X_{i,j}(t)$ 的权重系数, 用于控制第 1 阶段候选解远离自身位置并逐步向最优解搜索的幅度。 W_1 公式为:

$$W_1 = r_3 \times a_1 \times (\cosh r_4 + u \times \sin r_4 - 1) \quad (18)$$

式中: r_3, r_4 为 $[0, 1]$ 内随机数; u 为控制探索精度的敏感系数, 值为 0.388; a_1 为单调递减函数。 a_1 公式为:

$$a_1 = 3 \times \left(-1.3 \times \frac{t}{T_{\max}} + m\right) \quad (19)$$

式中: m 为敏感系数, 用于控制搜索精度, 且 $m=0.45$ 。

探索第 2 阶段, 种群个体受最优解影响较少, 个体将以当前位置为基础进行非定向位置搜索, 其位置更新方式为:

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + |\varepsilon \times W_2 \times X_{\text{best},j} - X_{i,j}(t)|, r_5 > 0.5 \\ X_{i,j}(t) - |\varepsilon \times W_2 \times X_{\text{best},j} - X_{i,j}(t)|, r_5 < 0.5 \end{cases} \quad (20)$$

式中: r_5 为 $[0, 1]$ 内随机数; ε 为极小正数, 为 0.003; W_2 为第 2 搜索阶段中 $X_{\text{best},j}$ 的权重系数。 W_2 公式为:

$$W_2 = r_6 \times a_2 \quad (21)$$

式中: r_6 为 $[0, 1]$ 内随机数; a_2 为单调递减函数。 a_2 公式为:

$$a_2 = 2 \times \left(-\frac{t}{T_{\max}} + n\right) \quad (22)$$

式中: n 为控制第二搜索阶段精度的敏感系数, 值为 0.5。

2.2 开发阶段

开发第 1 阶段, 个体侧重于搜索位置的邻近区域, 其位置更新方式为:

$$X_{i,j}(t+1) = \begin{cases} X_{\text{best},j} + r_7 \times W_3 \times X_{i,j}(t), r_8 > 0.5 \\ X_{\text{best},j} - r_7 \times W_3 \times X_{i,j}(t), r_8 < 0.5 \end{cases} \quad (23)$$

式中: r_7, r_8 为 $[0, 1]$ 内随机数; W_3 为第 1 开发阶段权重系数, 用于控制候选解开发搜索空间。 W_3 公式为:

$$W_3 = r_9 \times a_1 \times (\cosh r_{10} + u \times \sin r_{10}) \quad (24)$$

式中: r_9, r_{10} 为 $[0, 1]$ 内随机数, $u=0.388$ 。

开发第 2 阶段, 个体将围绕最优解进行深度开发, 且开发强度随迭代次数增加, 其位置更新方式为:

$$X_{i,j}(t+1) = X_{i,j}(t) + r_{11} \times \frac{\sinh r_{12}}{\cosh r_{12}} | W_2 \times X_{\text{best},j} - X_{i,j}(t) | \quad (25)$$

式中: r_{11}, r_{12} 为 $[0, 1]$ 内随机数; W_2 用于保持候选解多样性。

2.3 有界搜索

为了充分利用潜在搜索空间, SCHO 在后期采用一种类似动物狩猎的搜索策略, 称为有界搜索。为了实现潜在解的完全搜索与开发, 所有候选解将随机初始化解空间内。该策略的每一个起始点计算方式为:

$$BS_{k+1} = BS_k + \text{floor}\left(\frac{T_{\max} - BS_k}{\alpha}\right) \quad (26)$$

式中: k 为正整数, 起始值为 1; BS_k 代表当前有界搜索策略开始的迭代数; α 为敏感系数, $\alpha=4.6$ 。

$$BS_1 = \text{floor}\left(\frac{T_{\max}}{\beta}\right) \quad (27)$$

式中: β 用于控制有界搜索策略的起始值, 值为 1.55。

SCHO 每次执行有界搜索时, 上、下边界设置为:

$$ub_k = X_{\text{best},j} + \left(1 - \frac{t}{T_{\max}}\right) \times | X_{\text{best},j} - X_{\text{second},j} | \quad (28)$$

$$lb_k = X_{\text{best},j} - \left(1 - \frac{t}{T_{\max}}\right) \times | X_{\text{best},j} - X_{\text{second},j} | \quad (29)$$

式中: ub_k 和 lb_k 表示潜在搜索空间上界和下界; $X_{\text{second},j}$ 为次优解的 j 维位置。

2.4 转化机制

SCHO 利用转化机制实现搜索与开发阶段中 \sinh 和 \cosh 的转换。为了均衡搜索方式, 后期迭代, 转化机制集中于开发而搜索比例降低。转化机制为:

$$A = (p - q \times \eta^{\frac{\cosh \eta}{\sinh \eta}}) \times r_{13}, \eta = t/T_{\max} \quad (30)$$

式中: r_{13} 为 $[0, 1]$ 内随机数; p, q 用于均衡搜索与开发占

比,且 $p=10, q=9$ 。当 $A>1$ 时, SCHO 执行搜索; 当 $A<1$ 时, SCHO 执行开发。

3 分段引导和 SDSCHO 算法

3.1 Circle 混沌映射初始化

初始种群质量对智能优化算法的整体寻优精度和收敛速度影响很大。标准 SCHO 算法中, 种群初始化使用伪随机策略生成初始种群分布, 个体多样性较差, 容易造成算法搜索阶段寻优精度差、收敛速度慢。为此, 本文在初始化种群中引入 Circle 混沌映射。Circle 混沌映射是在单位圆上定义的混沌系统, 它通过某种非线性变换将单位圆上的点映射到新的位置, 从而产生混沌序列, 其数学公式为:

$$X_{i+1} = \text{bmod}(X_i + 0.2 - \frac{0.5}{2\pi} \sin(2\pi X_i), 1) \quad (31)$$

图 3(a) 所示为伪随机初始化, 图 3(b) 所示为 Circle 混沌映射初始化种群分布。可见, 比较伪随机初始化, Circle 混沌初始化能更均匀地将种群个体分布于解空间, 便于加速算法寻优和收敛。

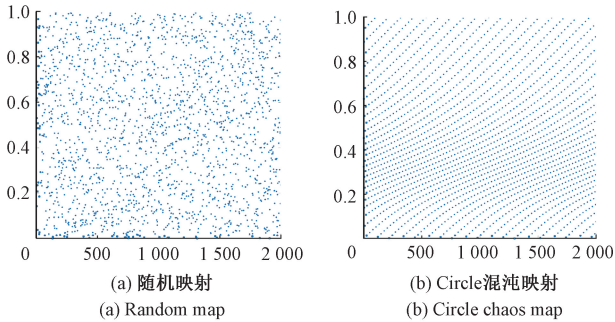


图 3 两种映射方式对比

Fig. 3 Comparison of two mapping methods

3.2 非线性震荡转换因子

转换因子 T 控制 SCHO 算法不同搜索策略的切换。 $t \leq T$ 时, 算法侧重在解空间中探索更多未知区域; $t > T$ 时, 算法根据探索区域进行重点局部开采。标准 SCHO 将 T 设置为固定常量, 两种策略被迭代次数严格区分无法进行有效信息补充, 若算法迭代前期未发现较好的个体, $t > T$ 后种群将受其影响陷入局部最优, 出现早熟现象。为此, 本文引入非线性震荡的转换因子 T_s , 公式为:

$$T_s = |T + c \times t + a \times \sin(b \times t)| \quad (32)$$

$$a = 1 + \frac{t}{2} \times r_{14} \quad (33)$$

$$c = 0.7 \times r_{14} \quad (34)$$

式中: T 为原转换因子; t 为当前迭代次数; $b = 1.5$; r_{14} 为 $[0, 1]$ 间随机数。

图 4 所示为改进前后转换因子对比趋势。相比 T , 改进后的 T_s 呈现非线性上升且随机震荡趋势。迭代后期, 算法依旧可以通过震荡低点使用探索阶段策略进行位置更新, 帮助算法探索更多解空间从而跳离局部最优。

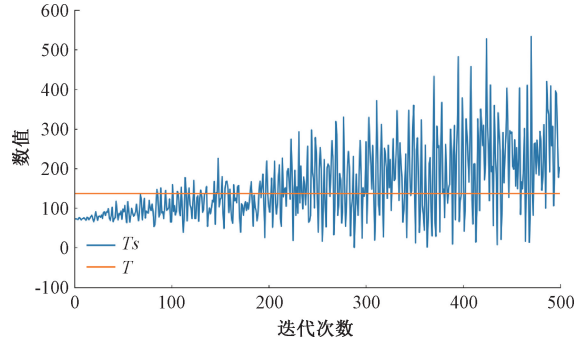


图 4 转换因子改进对比

Fig. 4 Comparison of conversion factor improvement

3.3 分段引导与反向逃逸优化策略

标准 SCHO 根据转换因子 T 实现不同搜索策略的转换。图 5 所示为 SCHO 在单峰函数 Sphere 上的收敛曲线。可见, $t \leq T$ 的迭代前期, SCHO 收敛速度极慢。此时种群个体以最优解为基准, 结合双曲正余弦函数特性进行位置更新。因此, 该阶段的位置更新会因个体信息丢失导致收敛速度过慢。 $t > T$ 时, 虽然 SCHO 在局部优化阶段收敛加快, 但每次迭代都以当前个体位置为基准, 结合当前最优解与当前个体位置间距进行位置更新。此时种群个体可能集中于次优解附近, 易导致算法陷入局部最优。

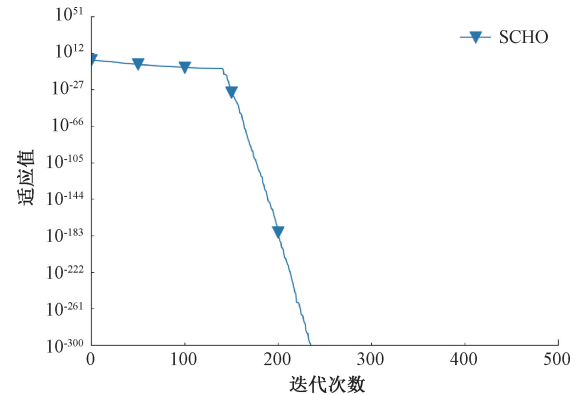


图 5 单峰函数收敛曲线

Fig. 5 Convergence curve of unimodal function

为了提高算法在不同阶段的搜索性能, 本文针对不同阶段的位置更新改进策略如下:

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + F \times r_{15} \times X_{best,j}(t), & t \leq T_s \\ X_{best,j}(t) + F \times r_{16} \times ((\frac{ub-lb}{2}) - X_{worst,j}(t)), & t > T_s \end{cases} \quad (35)$$

式中: t 为当前迭代次数; $X_{best,j}$ 为最优解; $X_{worst,j}$ 为最差解; lb 、 ub 对应空间下界与上界; r_{15} 、 r_{16} 为 $[0, 1]$ 间随机数; F 用于控制个体的移动方向。 F 公式为:

$$F = \begin{cases} +1, & p \leq 0.5 \\ -1, & p > 0.5 \end{cases} \quad (36)$$

式中: p 为 $[0, 1]$ 间随机数。

根据 SCHO 的转换因子 T_s 将算法迭代过程进行分段引导。根据式(35)可知, $t \leq T_s$ 时, 种群个体使用当前位置和最优个体引导进行位置更新, 可有效保留部分有效个体信息。 $t > T_s$ 时, 使用最优个体结合最差个体在解空间中的相反位置进行位置更新, 通过反向逃逸策略帮助算法跳出局部最优解。

3.4 动态边界分区辅助位置更新

在 SCHO 算法中, 算法不能动态调整收敛行为以便对次优解外的其他区域进行有效探索, 这种限制提高了算法陷入局部最优的概率。 为了提升算法跳出局部最优的能力, 引入动态边界分区辅助位置更新。 算法每轮迭代后, 将个体分为优等个体和次等个体两组, 针对不同组个体使用针对性辅助位置更新, 并利用贪婪策略选择更优个体。 在优等个体组中, 由于组内个体适应值相对较低, 为避免局部最优, 使用随机反向学习策略帮助算法搜索更广泛范围内的解空间, 有效增强算法跳出局部最优的能力。 随机反向学习公式为:

$$X_{new}(t) = lb + ub - r_{17} \times X_i(t) \quad (37)$$

式中: r_{17} 为 $[0, 1]$ 间随机数; lb 、 ub 为空间下界与上界。

针对次等个体组, 由于组内个体质量普遍较差, 个体信息价值较低, 使用动态分区二次初始化。 根据当前个体位置, 反向计算出新的动态临时解空间边界, 针对当前种群进行二次初始化, 此时位置更新为:

$$X_{new}(t) = r_{18} \times (inub - inlb) + inlb \quad (38)$$

$$inub = \left| \frac{ub}{2} - X_i(t) \right| \quad (39)$$

$$inlb = \left| \frac{lb}{2} - X_i(t) \right| \quad (40)$$

式中: r_{18} 为 $[0, 1]$ 间随机数; $inub$ 、 $inlb$ 为临时上界和下界。

4 基于 SDSCHO 算法的路径规划流程

利用 SDSCHO 处理路径规划问题, 需根据实际情况设置种群个体维度、搜索空间及适应度函数方可规划出最优路径。 根据三维山地模型, 经多次实验, 设置起点与终点之间节点数为 2, 结合三维空间中每个路径点具有 x 、 y 和 z 轴, 最终设置种群个体维度为 6, 对应路径点坐标为 (x_1, y_1, z_1) 、 (x_2, y_2, z_2) 。 通过算法每轮迭代后产生的解即为两个路径点。 为生成平滑路径, 本文使用三次

B 样条曲线法, 将两个路径点拟合生成该种群个体所代表的最终路径曲线, 并利用式(15)计算路径适应度值, 作为评价该种群个体拟合路径的质量评价参数。 经过算法迭代, 即可得到最优种群个体及其拟合路径。 图 6 所示为 SDSCHO 算法流程。

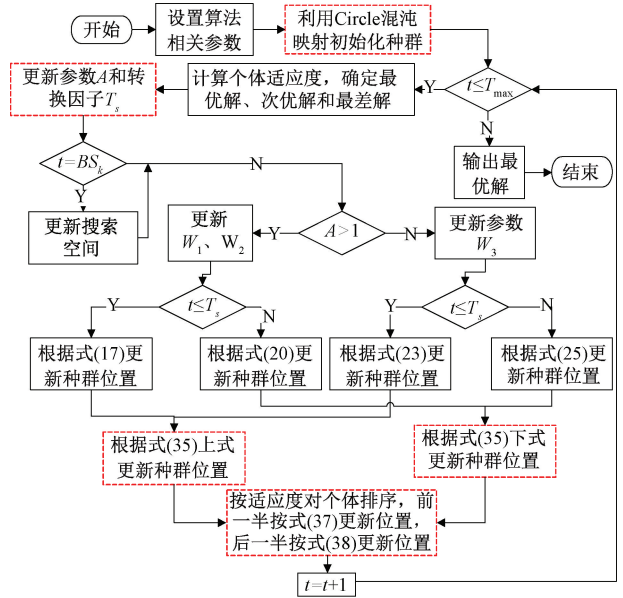


图 6 SDSCHO 算法流程

Fig. 6 SDSCHO algorithm process

5 实验分析

5.1 基准函数测试

基于 PyCharm2023. 1. 14 平台进行仿真实验。 设定种群规模 $NP = 20$, 迭代次数 $T_{max} = 500$ 。 从 CEC2005 基准库中选取 10 个不同特征的函数评估算法搜索性能, 函数说明如表 1 所示。 函数 $F_1 \sim F_5$ 为单模态, 函数 $F_6 \sim F_{10}$ 为复杂非线性多模态。 为了验证 SDSCHO 的性能优势, 将改进算法 SDSCHO 与 SCHO 算法^[19] 及其改进算法 MSCHO^[20]、PSO 算法^[1]、WOA 算法^[3]、AOA 算法^[4] 及 SAOA 算法^[10] 和 MAOA^[13] 进行纵横向对比。 SCHO 及其改进算法中 $c_1 = 3.6$, AOA 及其改进算法中, $c_1 = 2, c_2 = 6, c_3 = 1, c_4 = 1, u = 0.9, l = 0.1$ 。 WOA 中 $a = 2$ 。

1) 寻优性能分析

为确保测试公平性, 算法在相同条件下独立运行 30 次, 取指标均值对比分析, 寻优结果如表 2 所示。 可知, 在全部测试函数中, SDSCHO 都能找到相对最优的极值点。 针对单模态测试函数 $F_4 \sim F_5$, SDSCHO 相比标准 SCHO 和 MSCHO 都大幅提高了搜索精度。 在多模态测试函数 F_6 、 F_{10} 中, 由于局部最优值较多, 寻优难度更大, 只有 SDSCHO 找到理论最优值。 结果证明 SDSCHO 在寻优精度和跳出局部最优能力上具有明显优越性。

表1 基准测试函数
Table 1 Benchmark test function

测试函数	函数名称	维度	搜索范围	最优值
F_1	Sphere Function	30	$[-100, 100]$	0
F_2	Schwefel's Problem 1.2	30	$[-100, 100]$	0
F_3	Schwefel's Problem 2.21	30	$[-100, 100]$	0
F_4	Step Function	30	$[-100, 100]$	0
F_5	Quartic Function i. e. Noise	30	$[-1.28, 1.28]$	0
F_6	Generalized Schwefel's 2.26	30	$[-500, 500]$	-12 569.5
F_7	Generalized Penalized Function 1	30	$[-50, 50]$	0
F_8	Shekel's Foxholes	2	$[-65.536, 65.536]$	1
F_9	Hartman's Family	6	$[0, 1]$	-3.321 995
F_{10}	Shekel's Family	4	$[0, 10]$	-10.153 199

表2 寻优性能结果
Table 2 Optimization performance results

测试函数	统计指标	SDSCHO	SCHO	MSCHO	SAOA	MAOA	AOA	WOA	PSO
F_1	最优值	0.0	0.0	0.0	2.39×10^{-297}	4.34×10^{-142}	5.86×10^{-23}	3.23×10^{-20}	1 056.32
	平均值	0.0	0.0	0.0	4.07×10^{-261}	1.30×10^{-141}	3.84×10^{-21}	2.07×10^{-17}	2 445.94
	标准差	0.0	0.0	0.0	2.90×10^{-200}	3.76×10^{-285}	2.50×10^{-50}	3.39×10^{-34}	328 216.59
F_2	最优值	0.0	0.0	0.0	1.86×10^{-286}	4.55×10^{-108}	1.07×10^{-17}	2.118 75	5 608.856
	平均值	0.0	0.0	0.0	4.91×10^{-235}	1.51×10^{-108}	0.023 039	3.707 52	10 484.76
	标准差	0.0	0.0	0.0	1.69×10^{-212}	4.59×10^{-216}	0.001 061	3.995 79	2 4621 209.4
F_3	最优值	0.0	0.0	0.0	8.62×10^{-135}	4.79×10^{-65}	2.27×10^{-12}	0.000 82	23.625 4
	平均值	0.0	0.0	0.0	5.75×10^{-98}	1.58×10^{-65}	5.01×10^{-11}	0.002 01	27.134 2
	标准差	0.0	0.0	0.0	6.56×10^{-195}	5.01×10^{-130}	2.45×10^{-23}	4.05×10^{-6}	8.567 2
F_4	最优值	0.000 56	3.031 42	0.512 3	6.863 65	2.894 2	5.740 7	1.434 5	2 703.747
	平均值	0.000 63	3.574 05	0.786 44	6.979 56	3.222 26	6.461 77	1.610 16	3 206.60
	标准差	4.768×10^{-6}	0.396 52	0.034 35	0.003 81	0.054 15	0.277 66	0.044 92	935 140.54
F_5	最优值	5.95×10^{-6}	0.000 65	0.000 25	0.000 12	0.000 35	0.001 19	0.000 13	5.384 35
	平均值	4.01×10^{-6}	0.000 69	0.000 35	0.000 41	0.000 38	0.002 23	0.001 53	8.099 92
	标准差	1.44×10^{-9}	3.51×10^{-8}	3.15×10^{-8}	4.64×10^{-8}	1.38×10^{-8}	7.31×10^{-7}	1.56×10^{-6}	4.13009
F_6	最优值	-12 569.50	-6 008.82	-4 593.70	-1 945.41	-5 003.10	-4 940.11	-6 744.43	-3 781.08
	平均值	-12 474.71	-5 084.20	-4 430.70	-1 528.84	-4 827.70	-4 560.79	-6 140.71	-2 922.87
	标准差	16 407.91	275 749.73	1 048 944.87	134 260.94	43 604.38	1 142 607.57	222 964.65	355 660.57
F_7	最优值	0.000 24	0.492 78	0.004 62	1.348 28	0.312 57	1.869 17	0.060 01	33.178 9
	平均值	0.000 45	0.702 87	0.005 58	1.436 65	0.352 69	1.975 28	0.114 47	26 330.17
	标准差	9.35×10^{-7}	0.121 86	3.75×10^{-8}	0.025 65	0.002 83	0.229 40	0.002 93	802 198 664.89
F_8	最优值	0.998 01	12.682 11	12.670 5	2.982 10	0.995 21	0.994 30	1.982 12	6.992 03
	平均值	4.914 42	6.550 20	7.441 03	6.336 62	0.996 02	0.995 01	3.330 19	5.551 52
	标准差	17.760 0	26.005 4	20.858 9	10.665 4	3.41×10^{-25}	2.14×10^{-10}	0.850 01	25.339 9
F_9	最优值	-3.322 11	-3.136 72	-3.314 75	-2.111 44	-3.322 02	-2.526 23	-3.015 54	-3.309 77
	平均值	-3.322 01	-3.198 69	-3.371 95	-2.369 76	-3.343 56	-2.906 15	-2.052 42	-3.187 61
	标准差	1.01×10^{-12}	0.007 58	0.004 03	0.058 41	0.012 31	0.076 11	0.523 81	0.018 65
F_{10}	最优值	-10.1531	-5.105 00	-5.059 99	-3.768 89	-10.101 1	-4.575 76	-0.899 46	-2.630 41
	平均值	-10.153 1	-3.455 71	-9.402 30	-3.618 97	-10.011 3	-2.247 70	-0.278 94	-1.703 55
	标准差	4.66×10^{-11}	1.279 31	0.005 09	0.905 492	0.000 18	1.059 35	0.15743	2.461 77

2) 收敛性分析

以收敛曲线更直观展现算法的寻优精度和收敛速度,结果如图7所示。可见,对于单模态函数 F_1 、 F_4 、 F_5 ,相比SCHO迭代前期收敛速度较慢的不足,SDSCHO的曲线在同一阶段下降速度更快,说明在迭代前期,动态边

界分区辅助位置更新策略对算法寻优能力有明显提升。由于标准SCHO跳出局部最优能力差,MSCHO虽然优化了算法收敛速度,但其跳出局部最优能力并没有明显提升。相比之下,SDSCHO在保持收敛速度更快的前提下,搜索精度和跳出局部最优能力也大幅提高。在更考验算

法搜索精度的多模态函数 F_6 、 F_9 、 F_{10} 中,SDSCHO 都表现出更快的搜索速度且找到更优的极值点。尤其在函数 F_6 、 F_{10} 中,在标准算法和其他改进算法都陷入局部最优

值的情况下,SDCHO 能准确的找到全局理论最优值,表现出明显的寻优性能优势。

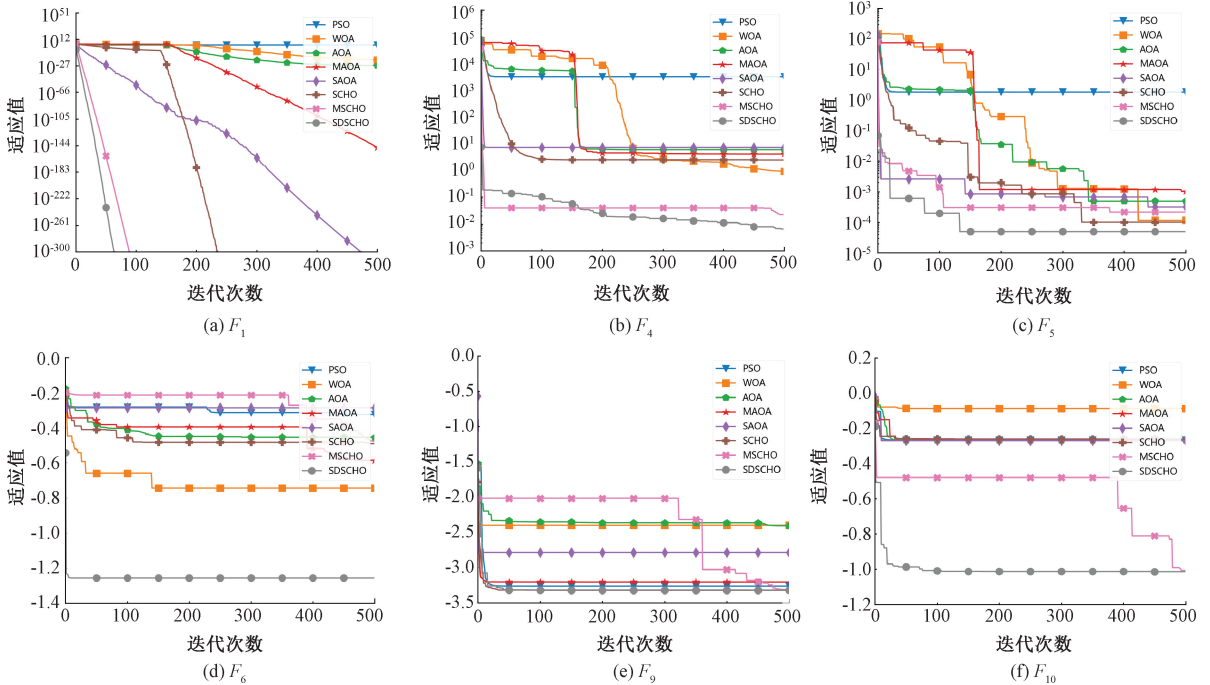


图 7 算法收敛性分析

Fig. 7 Algorithm convergence analysis

3) 稳定性分析

图 8 所示为 SDSCHO 的稳定性结果,针对 10 个测试函数进行 30 次实验得到最优解的分布情况,以箱型图表示。可见,SDSCHO 的异常值极少,除 F_6 以外的所有函数中,SDSCHO 都表现出更好的稳定性。在 F_6 中,SDSCHO 虽然存在异常值点,箱体同比略长,但其得到的最优值及其范围都远优于对比算法。综上,SDSCHO 在绝大多数测试函数中没有大幅度波动,其箱体最小性能也最稳定,具有更好的鲁棒性。

4) 消融实验

为验证改进策略的有效性,将 Circle 混沌映射初始化改进算法 SDSCHO1、非线性震荡转换因子改进算法 SDSCHO2、分段引导与反向逃逸优化改进算法 SDSCHO3、动态边界分区辅助位置更新改进算法 SDSCHO4 与标准 SCHO 开展消融实验,图 9 所示为消融实验收敛曲线。可见,4 项改进策略对比原算法均有不同程度性能提升。对于 F_1 、 F_4 和 F_5 ,SDSCHO1 的初始适应度值更低,表明 Circle 混沌初始化可以在迭代初期更均匀搜索解空间,找到适应度更优的个体。 F_1 、 F_5 、 F_6 、 F_9 和 F_{10} 中 SDSCHO2 前期收敛速度明显更快,后期搜索精度相比原算法更高,表明非线性震荡转换因子可有效帮助算法提高搜索性能。无论是单模态函数还是多模态函数,SDSCHO3 和 SDSCHO4 都表现更快的收敛速度,尤其

在局部极值更多、寻优难度更大的 F_6 、 F_9 、 F_{10} 上,分段引导与反向逃逸和动态边界分区辅助位置更新改进的两种算法均能找到更优极值点。可见,改进策略能不同程度提高算法搜索性能。

5.2 无人机路径协同规划实验

设置整体规划空间为 100 km×100 km×100 km 的三维空间,无人机起点坐标为(0,0,10),终点坐标为(100,100,70)。山地模型中山峰信息如表 3 所示,雷达威胁和禁飞区威胁信息如表 4 和 5 所示。代价模型中权重设置为 $w_1=0.5$ 、 $w_2=0.2$ 、 $w_3=0.3$ 。无人机质心半径 $r=2$,路径点数量为 3,最大转角 $\theta_{max}=90^\circ$ 。实验平台为 MATLAB R2020b。由路径点连接生成的航迹通过 3 次 B 样条进行路径平滑处理。

表 3 山峰信息

Table 3 Mountain information

编号	中心坐标	x 轴坡度	y 轴坡度	高度
1	(15, 80)	5	5	70
2	(20, 57)	5	10	50
3	(40, 85)	10	5	65
4	(55, 30)	5	7	80
5	(80, 20)	10	10	60
6	(80, 80)	10	7	65

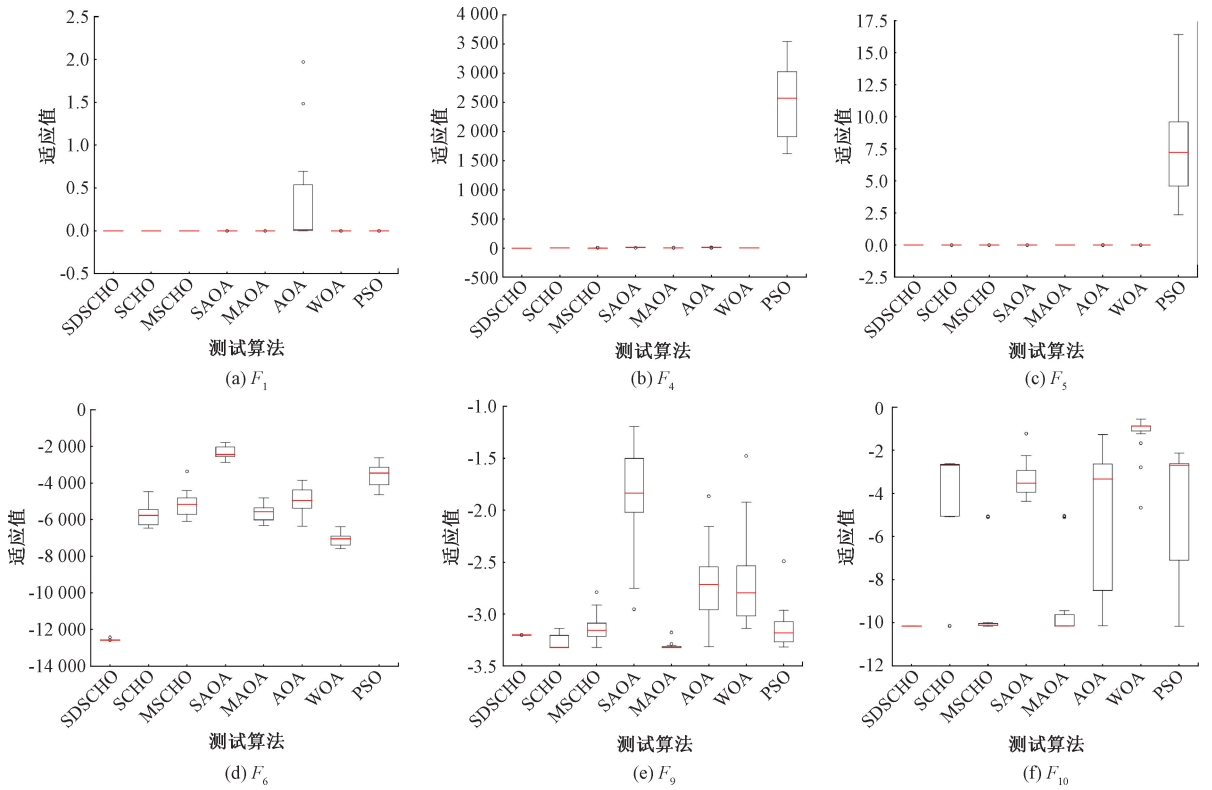


图8 算法稳定性分析

Fig. 8 Algorithm stability analysis

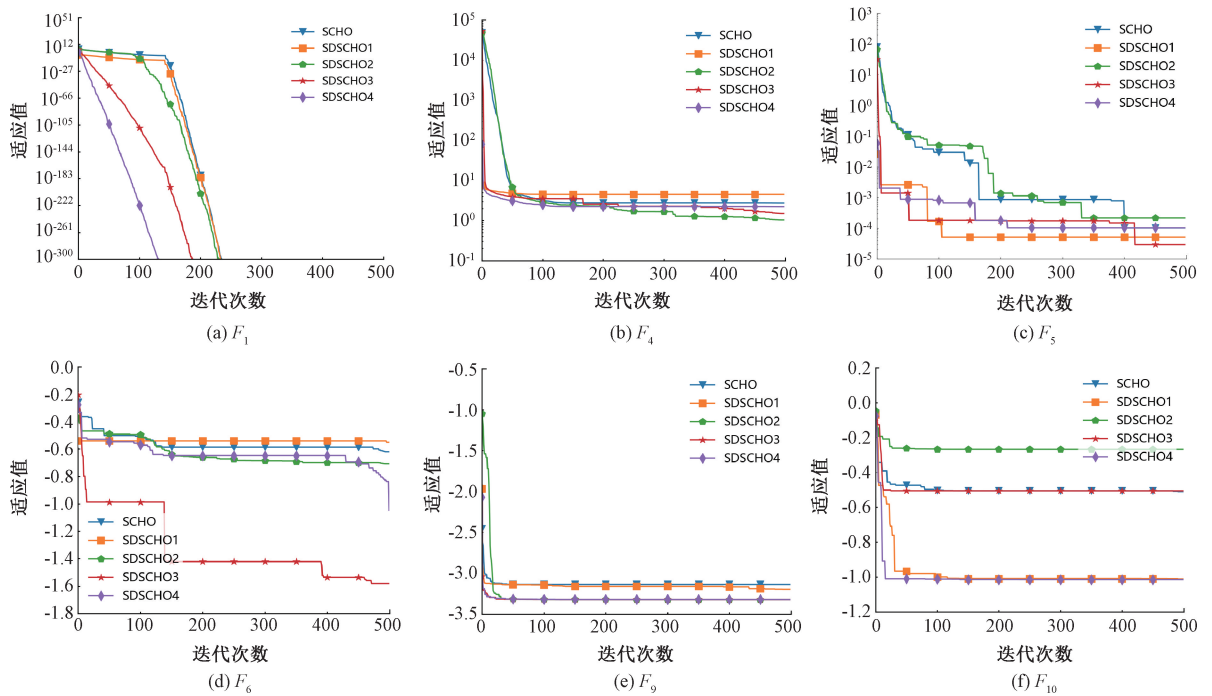


图9 消融实验收敛曲线

Fig. 9 Convergence curve of ablation experiment

表 4 雷达威胁信息

Table 4 Radar threat information

编号	中心坐标	威胁半径
1	(50,50)	15
2	(70,40)	15

表 5 禁飞区威胁信息

Table 5 No-fly zone threat information

编号	中心坐标	威胁半径	威胁高度
1	(30,30)	8	50
2	(60,80)	5	80
3	(70,20)	5	70

1) 单路径规划分析

将 SDSCHO 与标准 SCHO 和 MSCHO 在三维山地模型进行单路径规划分析,设置种群规模为 30,最大迭代次数为 50,算法执行 30 次,路径规划立体侧视及俯视图结果如图 10 所示,适应度收敛曲线如图 11 所示。迭代初期,SDSCHO 相比原算法在初始代价方面有较大提升,证明改进算法在种群初始化上具有更好性能;迭代后期,相比 MSCHO,SDSCHO 依然可以找到代价更低的路径,表现出更好的跳出局部最优能力。综上,相比于 SCHO 和 MSCHO,SDSCHO 的收敛速度更快,所规划的路径总体代价更低,路径更平滑。

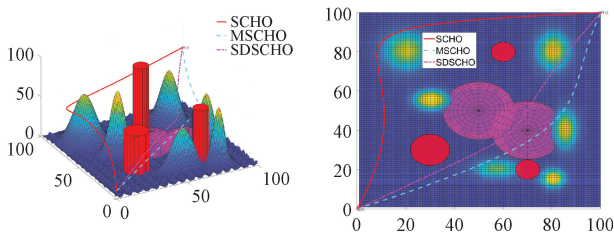


图 10 单路径规划结果

Fig. 10 Single-path planning results

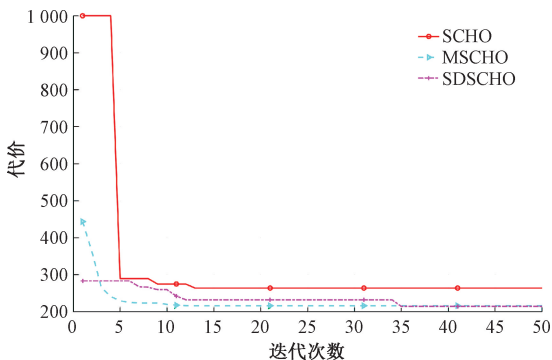


图 11 单路径规划收敛曲线

Fig. 11 Convergence curve of single-path planning

表 6 为单路径代价统计结果。最优代价方面,

SDSCHO 相比 SCHO 和 MSCHO 分别降低 7.2%、6.7%。平均代价方面,SDSCHO 相比 SCHO 和 MSCHO 分别降低 6.7%、6.65%。最差代价方面,SDSCHO 相比 SCHO 和 MSCHO 也分别降低 9.1%、8%。综上可知,SHSCHO 在路径代价最低的同时,也能保持更低的平均路径代价和路径代价方差,证明改进算法具有更好更稳定的路径规划性能。

表 6 单路径规划结果

Table 6 Single-path planning results

算法	最优代价	平均代价	最差代价	方差
SDSCHO	210.9	214.11	219.4	3.479 9
SCHO	227.4	229.6	241.6	9.553 3
MSCHO	226.1	229.11	238.6	6.792 9

2) 多路径协同规划分析

相比单路径规划,多路径协同规划需同步考虑无人机飞行中的碰撞问题,因此需要在路径代价模型中添加多无人机协同路径规划代价,如式(13)、(14)所示。为验证改进算法,将 SDSCHO 与 SCHO、MSCHO、SAOA、MAOA、AOA、WOA 和 PSO 在三维山地模型中进行比较,设置 3 架无人机在 3 个不同起点飞抵不同终点,起点 1(0,0,100)-终点 1(100,100,70),起点 2(0,0,50)-终点 2(90,100,30),起点 3(0,40,60)-终点 3(90,15,30)。选取 SDSCHO、SCHO 和 MSCHO 3 种算法勾画路径协同规划代价的收敛曲线如图 12 所示,算法的多路径协同规划结果如图 13 所示。表 7 为算法执行 30 次的统计结果。

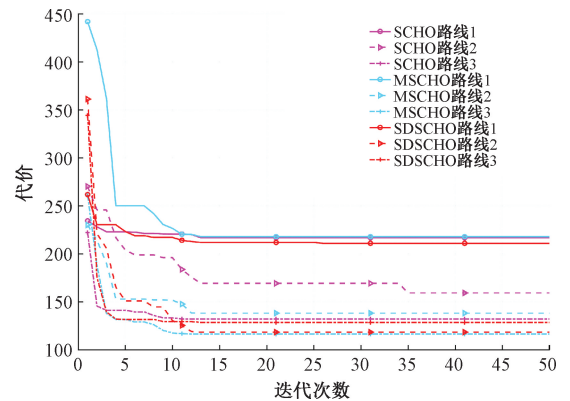


图 12 多路径协同规划代价收敛曲线

Fig. 12 Cost convergence curve of multi-path collaborative planning

由表 7 可知,相比其他算法,SDSCHO 规划路径代价均为最低,3 条协同规划路径平均代价上,SDSCHO 比 PSO、WOA、AOA、MAOA、SAOA、SCHO 和 MSCHO 分别可以降低 13.34%、10.36%、13.88%、11.85%、9.28%、9.34%、8.40%。在对比同类算法 MSHCO 和 SHCO 的收敛曲线(图 12),SDSCHO 的平均路径代价也是最低。同

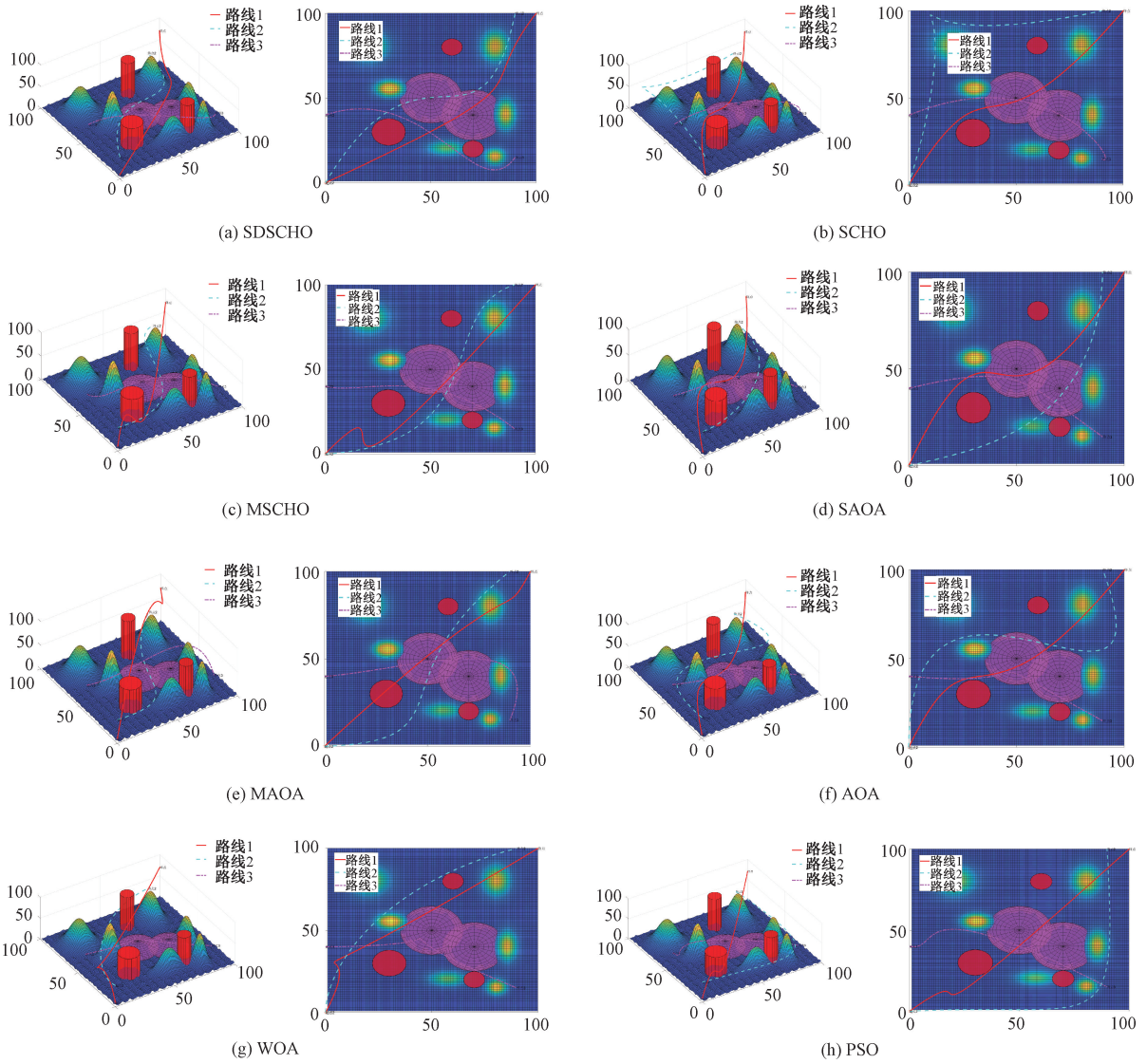


图 13 多路径协同规划结果

Fig. 13 Multi-path collaborative planning results

表 7 多路径协同规划结果

Table 7 Multi-path collaborative planning results

算法	路径 1 代价	路径 2 代价	路径 3 代价	平均路径代价
SDSCHO	212.2	114.9	116.1	147.7
SCHO	213.4	137.2	133.8	161.5
MSCHO	226.2	136.6	117.7	160.1
SAOA	217.4	131.9	134.9	161.4
MAOA	222.5	127.2	146.1	165.2
AOA	234.9	144.7	125	168.2
WOA	213.7	137.5	137.9	163
PSO	217.2	145.5	139.5	167.4

时,从图 13 可见,SDSHCO 规划的 3 条路线相比对比算法更加平滑,能够安全避让所有障碍物和威胁区域,不存在绕行路径,证明 SDSHCO 在多路径协同规划问题求解上同样具有显著的性能优势。

6 结 论

本文提出了一种基于分段引导和动态分区改进的双曲正弦优化算法,以此解决双曲正弦优化算法在无人机路径协同规划中搜索精度与跳离局部最优能力不足的问题。通过引入混沌 Circle 映射种群初始化、非线性震荡转换因子、分段引导与反向逃逸优化策略及动态边界分区辅助位置更新策略,降低算法陷入局部最优的概率。将改进算法应用于求解无人机单路径、多路径协同规划场景,证实改进后算法在处理无人机路径协同规划上具有更强的寻优能力。由于实战环境更加复杂,构建符合实战环境的动态威胁源环境并开展在线实时路径规划将是后续的研究重点。

参考文献

- [1] BLINDEIM S, JOHASEN A. Particle swarm optimization for dynamic risk-aware path following for autonomous ships[J]. Ifac Papersonline, 2022, 55(31): 70-77.
- [2] BAI P, LIU N, DONG ZH L, et al. Genetic algorithm-based meta cellular automata model for evacuation from complex buildings [J]. Experimental Technology and Management, 2024, 41(8): 236-243.
- [3] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [4] HASHIM F, HUSSAIN K, HOUSSEIN E, et al. Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems [J]. Applied Intelligence, 2021, 51(3): 1531-1551.
- [5] XU J K, SHE N. A novel swarm intelligence optimization approach: Sparrow search algorithm [J]. Systems Science & Control Engineering, 2020, 8(1): 22-34.
- [6] MIRJALILI S. SCA: A sine cosine algorithm for solving optimization problem [J]. Knowledge-Based Systems, 2016, 96(15): 120-133.
- [7] ABUALIGAH L, DIABAT A, MIRJALILI S, et al. The arithmetic optimization algorithm [J]. Computer Methods in Applied Mechanics and Engineering, 2021, 37(6): 113-125.
- [8] GUO Y, YANG H, CHEN M, et al. Grid-based dynamic robust multi-objective brain storm optimization algorithm [J]. Soft Computing, 2020, 24(10): 7395-7415.
- [9] 郝琨, 邓晁硕, 赵璐, 等. 基于区域搜索粒子群算法的机器人路径规划 [J]. 电子测量与仪器学报, 2022, 36(12): 126-135.
- HAO K, DENG CH SH, ZHAO L, et al. Robot path planning based on region search particle swarm optimization [J]. Journal of Electronic Measurement and Instrumentation, 2022, 36(12): 126-135.
- [10] 罗仕杭, 何庆. 融合 Sin 混沌和分段权值的阿基米德优化算法 [J]. 计算机工程与应用, 2022, 58(14): 63-72.
- LUO SH H, HE Q. Archimedes optimization algorithm combining Sin chaos and segmented weights [J]. Computer Engineering and Applications, 2022, 58(14): 63-72.
- [11] 何庆, 林杰, 徐航. 混合柯西变异和均匀分布的蝗虫优化算法 [J]. 控制与决策, 2021, 36(7): 1558-1568.
- HE Q, LIN J, XU H. Hybrid Cauchy mutation and uniform distribution of grasshopper optimization algorithm [J]. Control & Decision, 2021, 36(7): 1558-1568.
- [12] 郭云川, 张长胜, 段青娜, 等. 融合多策略的改进秃鹰搜索算法 [J]. 控制与决策, 2024, 39(1): 69-77.
- GUO Y CH, ZHANG CH SH, DUAN Q N, et al. Improved bald eagle search algorithm fused with multiple strategies [J]. Control & Decision, 2024, 39(1): 69-77.
- [13] 罗仕杭, 何庆. 多策略协同改进的阿基米德优化算法及其应用 [J]. 计算机应用研究, 2022, 39(5): 1386-1394.
- LUO SH H, HE Q. Improved Archimedes optimization algorithm by multi-strategy collaborative and its application [J]. Application Research of Computers, 2022, 39(5): 1386-1394.
- [14] 冉宁, 杨宏飞, 张家明, 等. 基于改进蚁群算法的无人机三维航迹规划 [J]. 电子测量技术, 2023, 46(20): 41-49.
- RAN N, YANG H F, ZHANG J M, et al. 3D path planning of UAV based on improved ant colony algorithm [J]. Electronic Measurement Technology, 2023, 46(20): 41-49.
- [15] 班云飞, 张达敏, 左锋琴, 等. 精英反向学习及柯西扰动引导的瞪羚优化算法 [J]. 国外电子测量技术, 2024, 43(7): 1-13.
- BAN Y F, ZHANG D M, ZUO F Q, et al. Elite reverse learning and Cauchy perturbation-guided gazelle optimization algorithm [J]. Foreign Electronic Measurement Technology, 2024, 43(7): 1-13.
- [16] 付华, 刘昊. 多策略融合的改进麻雀搜索算法及其应用 [J]. 控制与决策, 2022, 37(1): 87-96.
- FU H, LIU H. Improved sparrow search algorithm with multi-strategy integration and its application [J]. Control & Decision, 2022, 37(1): 87-96.
- [17] 李全耀, 沈艳霞. 一种基于教与学的混合灰狼优化算法 [J]. 控制与决策, 2022, 37(12): 3190-3196.
- LI Q Y, SHEN Y X. A hybrid gray wolf optimization algorithm based on the teaching-learning optimization [J]. Control & Decision, 2022, 37(12): 3190-3196.
- [18] 王雨虹, 王志中, 付华, 等. 多策略改进麻雀算法与 BiLSTM 的变压器故障诊断研究 [J]. 仪器仪表学报, 2022, 43(3): 87-97.
- WANG Y H, WANG ZH ZH, FU H, et al. Research on transformer fault diagnosis based on improved multi-strategy sparrow algorithm and BiLSTM [J]. Chinese Journal of Scientific Instrument, 2022, 43(3): 87-97.
- [19] BAI J, LI Y, ZHENG M P, et al. A sinh cosh optimizer [J].

Knowledge-based Systems, 2023, 282(20): 1-29.

- [20] IBRAHIM R, ALY M, MOEMEN Y, et al. Boosting sinh cosh optimizer and arithmetic optimization algorithm for improved prediction of biological activities for indoloquinoline derivatives [J]. Chemosphere, 2024, 359(7): 1-12.

作者简介



张小庆, 2013 年于武汉理工大学获得博士学位, 现为武汉轻工大学副教授, 主要研究方向为群智能优化算法及应用、机器学习等。

E-mail: 51449902@qq.com

Zhang Xiaoqing received his Ph. D.

degree from Wuhan University of Technology in 2013. Now he is an associate professor at Wuhan Polytechnic University. His main research interests include swarm intelligence optimization algorithm and application, machine Learning, etc.



孙民民(通信作者), 2023 年于山东科技大学获得学士学位, 现为武汉轻工大学电子信息专业研究生, 主要研究方向为智能优化算法、路径规划。

E-mail: jiangbei-north@qq.com

Sun Minmin (Corresponding author) received his B. Sc. degree from Shandong University of Science and Technology in 2023. Now he is a M. Sc. candidate in Wuhan Polytechnic University. His main research interests include intelligent optimization algorithms and path planning.