

基于改进樽海鞘群算法的无人机高程模型航迹规划

赵南南, 吕尚扬, 吴广政, 乔鹏博, 王洪波

(西安建筑科技大学机电工程学院, 陕西 西安 710055)

摘要: 针对启发式算法在无人机不规则复杂地形和多重威胁环境下进行三维航迹规划时, 存在路径波动大和优化性能不足的问题, 提出结合高程数据的凸包策略以及一种改进的樽海鞘群算法(ISSA)。首先, 基于ASTER GDEM V3和Open Street Map数据, 构建杭州某处山区和纽约城市区域的高程模型; 其次, 结合地形高程信息, 采用凸包策略编码并通过B样条曲线构建路径; 最后, 对樽海鞘群算法在个体位置更新公式上加入自适应Alpha稳定分布策略与非线性扰动策略, 以平衡算法的全局开发能力与局部探索能力, 并引入贪婪策略和鱼类聚集装置策略, 提高算法搜索效率和精度。利用CEC2020测试函数对所提算法进行实验对比, 验证了改进算法的性能。实验结果表明, 凸包策略能有效提升算法规划能力, 且与传统算法相比, 改进后的算法能够使无人机的寻优精度更高, 代价函数更小。

关键词: 航迹规划; 凸包策略; 樽海鞘群算法; 自适应Alpha稳定分布策略; 鱼类聚集装置策略

DOI: 10.11907/rjtk.241888

扫描二维码阅读全文:



中图分类号: TP273

文献标识码: A

文章编号: 1672-7800(2026)001-0063-12

UAV Elevation Models Path Planning Based on Improved Salp Swarm Algorithm

ZHAO Nannan, LYU Shangyang, WU Guangzheng, QIAO Pengbo, WANG Hongbo

(School of Mechanical and Electrical Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: To tackle path fluctuations and suboptimal optimization in 3D UAV path planning over complex terrain and multiple threats, we propose a convex hull strategy with elevation data and an improved salp swarm algorithm (ISSA). First, construct elevation models for a mountainous area in Hangzhou and an urban area in New York based on ASTER GDEM V3 and Open Street Map data. Next, we encode the terrain elevation information using the convex hull strategy and construct paths using B-spline curves. Finally, we enhance the SSA by incorporating adaptive Alpha-stable distribution strategy and nonlinear disturbance strategies in the individual position update formula to balance global exploration and local exploitation. Additionally, we introduce greedy and fish aggregation device (FAD) strategies to improve the efficiency and accuracy searches. The performance of the improved algorithm was validated by conducting experimental comparisons using the CEC2020 test functions. The results demonstrate that the convex hull strategy significantly enhances the planning capability of the algorithm. Compared to traditional algorithms, the improved SSA achieves higher optimization accuracy and lower cost functions for UAV path planning.

Key Words: flight path planning; convex hull strategy; salp swarm algorithm; adaptive alpha-stable distribution strategy; fish aggregation device strategy

0 引言

近年来,随着无线通信传感技术、智能控制技术和机械制造业的迅速发展,无人飞行器技术也取得了显著进

步,并呈现出日益成熟的趋势。由于无人机的快速发展及其在成本效益和灵活性方面的优势,其在军事和民用领域得到了广泛应用。然而,无人机的能源有限,执行任务的时间也受到限制。因此,如何找到一条低能耗、无碰撞的路径,并确保无人机远离潜在威胁或限制区域,是无人机

收稿日期: 2024-11-26

基金项目: 国家自然科学基金青年科学基金项目(52206109)

作者简介: 赵南南(1983-),男,博士,西安建筑科技大学机电工程学院副教授,研究方向为群体智能、电机控制;吕尚扬(1999-),男,西安建筑科技大学机电工程学院硕士研究生,研究方向为群体智能、路径规划;吴广政(1999-),男,西安建筑科技大学机电工程学院硕士研究生,研究方向为群体智能、微电网优化调度;乔鹏博(2000-),男,西安建筑科技大学机电工程学院硕士研究生,研究方向为群体智能、新能源发电与并网技术;王洪波(1995-),男,西安建筑科技大学机电工程学院助理工程师,研究方向为机电一体化及自动机器人、嵌入式系统。本文通讯作者: 吕尚扬。

路径规划中的关键问题^[1]。

航迹规划方法可分为如 Dijkstra 算法^[2]、A* 算法^[3]、人工势场法^[4]和 RRT 算法^[5]等的传统经典方法,以及如粒子群算法^[6]、蚁群算法^[7]、灰狼算法^[8]和鲸鱼算法^[9]等的智能优化算法。智能算法因其具有适应性好、实现简单和鲁棒性高的特点,受到了大量学者的广泛研究。文献[10]提出球面矢量粒子群优化算法,使用球面坐标系将路径规划问题转化为多约束的优化问题,并对代价函数进行求解。文献[11]针对多场景多障碍无人机路径规划问题,提出一种改进海鸥算法,有效避免了算法陷入局部最优。文献[12]建立模拟城市模型,结合混沌映射、自适应惯性权重和指数变化学习因子,对粒子群算法进行优化,提升了粒子群算法在求解城市地形路径规划问题中的效果。文献[13]提出一种混合灰狼和差分的进化算法,有效平衡了算法的开发能力和探索能力,并在柱形障碍物地图上进行了测试,证明了该算法能够获得更平稳,更短的轨迹。文献[14]提出一种多策略融合的宽吻海豚优化器,引入鱼类聚集装置策略、随机混合变异策略与无序反向学习策略,并通过实验对比验证了该算法在无人机路径规划算法中的良好综合性能。尽管这些算法在航路规划中表现较好,但随着无人机任务复杂性和规划环境多样性的日益增加,其对算法计算处理能力的要求也在不断提高。并且,大多数文献均使用基于数学模型的基准地形对算法进行验证,考虑复杂地形不规则性的文献较少。

凸包算法在无人机领域有较多应用。文献[15]将凸包理论应用于无人机部署,旨在最小化地面终端用户与无人机之间的平均距离。文献[16]提出一种双侧凸包扩张模型并将其应用于二维路径规划,使得路径运算效率得以提升,能够获得实际最短路径。尽管凸包算法在无人机部署和平面路径规划中显示出了优越性,其在航迹规划中的潜力仍未得到充分挖掘。

樽海鞘群算法(Salp Swarm Algorithm, SSA)由 Mirjalili 等^[17]提出,该算法原理易于理解、结构简单、参数少,且具有良好的寻优能力,已成为许多学者研究的热点,并在多阈值图像分割^[18]、永磁同步电机的参数辨识^[19]和流水车间调度^[20]问题上得到应用。虽然该算法求解大部分优化问题具有优越性,但标准樽海鞘群算法在处理复杂问题时仍然存在求解精度不高、易陷入局部最优等问题。文献[21]在领导者与跟随者位置更新公式中引入黄金正弦算法,自适应权重因子等策略,平衡了算法的全局搜索和局部勘探能力,增强了算法跳出局部最优的能力。文献[22]提出一种正交反向学习策略和惯性权重策略,使改进后的算法具有更强的全局搜索能力。文献[23]采用 Sobol 序列初始化种群,并结合改进透镜成像反向学习与自主追随机制进一步改进领导者和跟随者的位置更新公式,有效提升了算法寻优效果和稳定性。虽然这些樽海鞘群变体在一定程度上提高了算法搜索性能,但其搜索速度、收敛精度

和应对复杂问题的稳定性仍是一项具有挑战性的任务。

为此,本文以基本樽海鞘群算法为基础,在算法的位置更新阶段与生成新解过程中引入自适应 Alpha 稳定分布策略、非线性扰动策略、贪婪选择策略与鱼类聚集装置策略,提升了 SSA 算法的开发和探索能力。同时,引入凸包策略并建立真实的山区和城市地形三维航迹规划模型,以进行仿真实验验证。

1 无人机路径规划模型

1.1 环境建模

数字高程模型(Digital Elevation Model, DEM)是地球科学研究中最广泛应用的基础数据之一,提供了地表的三维高程信息。山区环境模型的基准地形数据由 ASTER GDEM V3 30 m 获取,位置来源于杭州某处山区,其中 30 m 精度表示地图上每个栅格覆盖的 30 m×30 m 实际地面区域。如图 1 所示,DEM 数据通过栅格形式直观地表现地形的起伏变化。

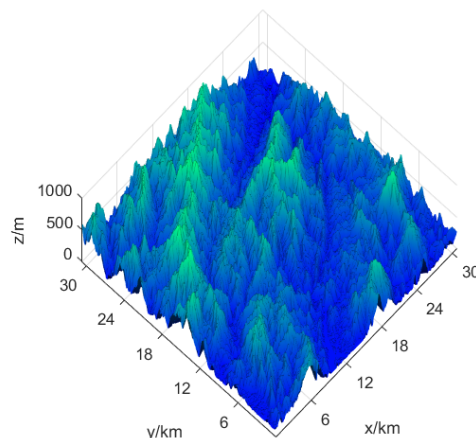


Fig. 1 Mountain elevation model

图 1 山区高程模型

城市环境模型是于 Open Street Map(OSM)数据库中提取纽约某处建筑物信息而构建,经纬度位置分别为 $(-74.012\ 84, 40.712\ 26)$ 和 $(-74.000\ 26, 40.712\ 51)$,并将在线地理位置系统 WGS84 的纬度和经度坐标转换为笛卡尔坐标系。

在提取的地理信息中,建筑物通过一系列点坐标及其标签构成的空间多边形进行表示。常规的轴对称包围盒(Axis-aligned Bounding Box, AABB)不能对由点坐标构成的建筑物进行完全覆盖,故本文对建筑物信息使用主成分分析法(Principal Component Analysis, PCA)分析其坐标数据,找到数据的主要变化方向,使计算得到的最小有向包围盒(Oriented Bounding Box, OBB)能够对齐建筑物的主要方向,从而获得更符合实际的最小矩形边框。模型最终效果如图 2 所示。

对生成的最小有向包围盒使用向量叉乘判别法,计算测试点与多边形的叉乘和点积,从而判断该点是否位于包

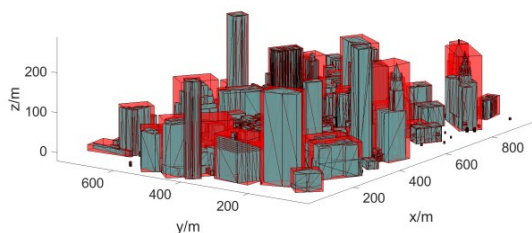


Fig. 2 Urban model

图 2 城市模型

围盒的内部或边界上,并生成高程地形。城市建筑高程转换如图 3 所示。

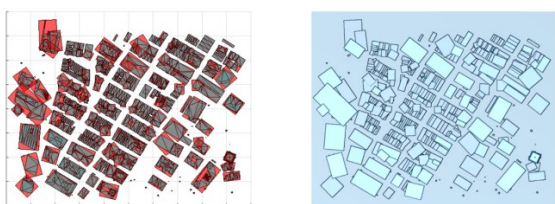


Fig. 3 Urban building elevation conversion

图 3 城市建筑高程转换

1.2 航迹编码与表示

在三维空间直角坐标系 $O-XYZ$ 中(见图 4),圆柱形障碍物模拟了不可飞行区域。在路径规划过程中,无人机飞行路径需绕开障碍物区域。规划路径从起点 S 出发,经过若干中间航迹点,最终达到目标点 E 。其中, P_1 到 P_n 为无人机规划中间航迹点, n 为规划航迹点个数。

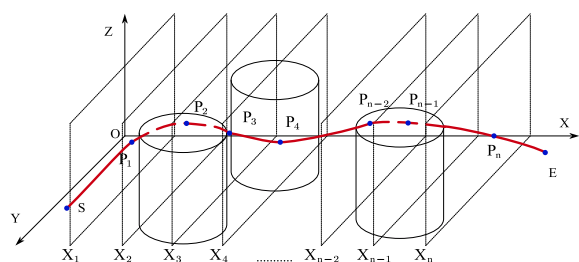


Fig. 4 Flight path planning diagram

图 4 航迹规划示意图

首先生成平面路径的二维坐标,中间航迹点的平面坐标从 (x_1, y_1) 开始,到 (x_n, y_n) 结束。每个个体的决策变量表示一条路径,生成一组 $n \times 2$ 的矩阵 $(s_{x1}, s_{y1}, \dots, s_{xn}, s_{yn})$,其中 (s_{xi}, s_{yi}) 生成于 0 到 1 之间的随机数, (x_s, y_s) 表示起点平面坐标, (x_E, y_E) 表示终点平面坐标,如式(1)所示。

$$\begin{aligned} x_i &= x_s + s_{xi} \cdot (x_E - x_s) \\ y_i &= y_s + s_{yi} \cdot (y_E - y_s) \end{aligned} \quad (1)$$

传统的编码方式难以在复杂真实地形的航迹规划上取得好的效果,故对于规划路径的高度信息,结合山区环境中山体起伏较大的特点,使用凸包算法的方式生成凸包顶点。凸包算法用于找到一个给定点集的凸多边形,该多

边形包含了所有给定的点,如图 5 所示。

通过将凸包顶点与起点和终点连接生成路径规划轨迹,可以有效降低碰撞。同时,由于规划点不会采集到山谷中的位置,也在一定程度上提升了运行框架应对复杂真实地形的运行效率。无人机高度规划如图 6 所示。

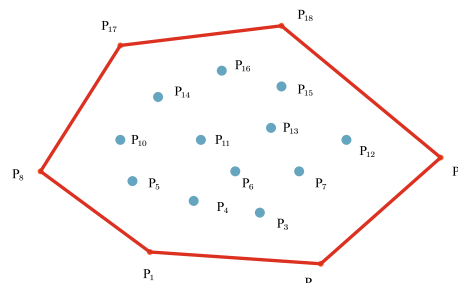


Fig. 5 Convex hull strategy

图 5 凸包策略

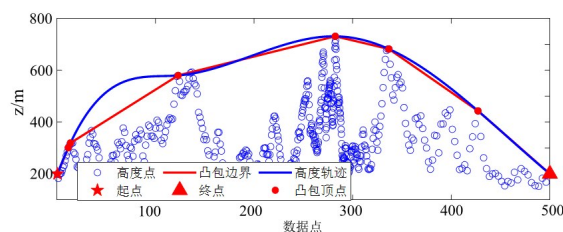


Fig. 6 Unmanned aerial vehicle altitude planning

图 6 无人机高度规划

数据点的高度由 xy 平面规划路径对应的高程信息所得。高度信息计算公式如式(2)所示。

$$Z_i = \max_{p \in P} (Z_p) + h_{safe} \quad (2)$$

其中, P 代表了平面规划路径生成点附近的基准地形高度值, h_{safe} 表示无人机与山体间飞行的安全距离,该参数可用于控制山体与无人机间的高度差。

1.3 路径平滑

在完成平面路径规划后,需要对生成的路径点进行平滑插值。同样,在高度点生成后,需要结合 x 轴或 y 轴进行插值。因此,为了优化无人机的飞行路径轨迹,使用 B 样条曲线方法对路径点进行平滑处理,将其转换为平滑的轨迹点集合。B 样条曲线具有简单灵活的表征能力,是一种被广泛使用的路径平滑方法,其具体实现方法如式(3)一式(5)所示。

$$\begin{cases} \overrightarrow{x(t)} = \sum_{i=0}^n \overrightarrow{x_i} \cdot B_{i,j}(t) \\ \overrightarrow{y(t)} = \sum_{i=0}^n \overrightarrow{y_i} \cdot B_{i,j}(t) \\ \overrightarrow{z(t)} = \sum_{i=0}^n \overrightarrow{z_i} \cdot B_{i,j}(t) \end{cases} \quad (3)$$

其中, $B_{i,j}(t)$ 为基函数, (x_i, y_i, z_i) 为控制点,具体基函数的递归更新公式如式(4)、式(5)所示。

$$B_{i,0}(u) = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$B_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} B_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} B_{i,k-1}(u) \quad (5)$$

1.4 飞行约束条件

在规划无人机路径时,为了提高飞行稳定性和安全性,并减少失稳风险,需将路径长度和转角视作一个严格的约束条件。如果在路径规划过程中发现某些路径段不符合这些约束条件,需对不符合的参数进行调整,直至所有路径段均满足约束要求。

1.4.1 路径长度约束

无人机在飞行过程中需要确保其路径长度不超过燃料所允许的最大范围。路径长度约束如式(6)所示。

$$\sum_{i=1}^{n-1} L_i \leq L_{\max} \quad (6)$$

其中,路径长度 L_i 的计算公式如式(7)所示。

$$L_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad (7)$$

式(6)中, $L_{\max}$ 为最大航程, n 为路径点个数。如果路径长度超出最大航程约束,则需重新生成路径点,以确保路径长度在允许范围内。

1.4.2 飞行转角约束

为了确保无人机在三维空间中飞行路径稳定,并防止因方向突然变化而失控,需要在保持飞行稳定的同时,严格约束水平偏航角和竖直俯仰角,这些约束对飞行路径优化至关重要。水平偏航角不应超过最大值 $\phi_{\max}$,竖直俯仰角不应超过 $\theta_{\max}$ 。对于规划路径点中不满足约束的部分,采用随机调整操作,调整其平面位置或高度,直到满足约束条件为止^[24]。水平偏航角与竖直俯仰角可通过式(8)和式(9)进行计算。

$$\cos^{-1} \left(\frac{L_{i,i-1} \cdot L_{i+1,i}}{\|L_{i,i-1}\| \|L_{i+1,i}\|} \right) \leq \phi_{\max} \quad (8)$$

$$\tan^{-1} \left(\frac{\|z_{i+1} - z_i\|}{\sqrt{\|(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2\|}} \right) \leq \theta_{\max} \quad (9)$$

其中, $L_{i,i-1}$ 表示第 $i-1$ 个路径点到第 i 个路径点的欧式距离, $(x_{i-1}, y_{i-1}, z_{i-1})$ 和 (x_i, y_i, z_i) 为上一个路径规划点与下一个路径规划点的坐标。

1.5 航迹代价函数

航迹代价用于评估每次迭代过程中路径的质量,从而择优选择并获取最优解。在复杂环境中,不同目标可能存在冲突,代价函数通过加权求和的方法进行平衡,综合考虑如路径长度、飞行高度变化、路径平滑度和避障威胁程度,从而使得最终选择的路径在综合指标上达到最优。路径总代价函数 P_i 被定义为如式(10)所示。

$$C(P_i) = \sum_{k=1}^3 u_k C_k(P_i) + C_4(P_i) \quad (10)$$

$$0 \leq \mu_k \leq 1, \sum_{k=1}^3 \mu_k = 1$$

其中, C 为总的成本代价, C_1 为航迹长度代价, C_2 为飞行高度变化代价, C_3 为飞行平滑度代价, C_4 为威胁成本代价, μ_k 为各代价的系数。

1.5.1 航迹长度代价

无人机的路径长度在很大程度上决定了飞行效率,是影响飞行时间和能耗的关键因素之一。较短的路径不仅意味着无人机能够更快速地到达目标位置,还可以显著减少电池消耗和其他资源使用。航迹长度代价如式(11)所示。

$$C_1(P_i) = \sum_{i=1}^{n-1} L_i \quad (11)$$

其中, L_i 表示相邻两点的欧式距离。

1.5.2 飞行高度变化代价

由于爬升下降过程会使无人机产生更多能耗,且飞行高度变化会增加无人机的总能耗,因此应尽量保持飞行高度平稳。无人机飞行高度变化代价计算公式如式(12)所示。

$$C_2(P_i) = \sqrt{\frac{1}{n-2} \sum_{i=2}^{n-1} \left(z_i - \frac{1}{n-2} \sum_{i=2}^{n-1} z_i \right)^2} \quad (12)$$

其中, z_i 表示当前点的高度。

1.5.3 飞行平滑度代价

受障碍物或地形影响,无人机在飞行过程中的路径不可避免地会进行一些路径调整。通过优化路径平滑度,不仅可以提高避障效果,还能减少急转弯带来的能耗和时间损失,提升飞行稳定性和任务执行效率。其无人机飞行平滑度代价可表示为如式(13)所示。

$$C_3(P_i) = \sum_{i=1}^{n-1} \phi_i + \sum_{i=1}^{n-1} \theta_i \quad (13)$$

其中, ϕ_i 和 θ_i 分别为水平偏航角和竖直俯仰角。

1.5.4 威胁代价

在无人机路径规划过程中,可能会遇到来自地面和空中的多重威胁,包括障碍物、敌方威胁区域和自然环境因素等。为了有效地规避这些障碍物,需要在规划路径时综合考虑地面和空中的威胁代价。特别是,当无人机的路径点高度低于附近高程值,容易受到地面威胁的影响。这些威胁代价通常与路径点的高度偏差与距离威胁源的距离相关,如式(14)所示。

$$\begin{aligned} C_{v1}(P_i) &= \sum_{j=1}^m \left(\frac{1}{n} \sum_{i=1}^n \max(1 - \frac{d_j}{R_j}, 0) \right) \times 10^7 \\ C_{v2}(P_i) &= \sum_{k=1}^s \left(\frac{1}{n} \sum_{i=1}^n \max(1 - \frac{d_k}{R_k}, 0) \right) \times 10^7 \\ C_{v3}(P_i) &= \sum_{i=2}^{n-1} \sum_{p=1}^4 \max(Z_p - Z_i, 0) \times 10^7 \end{aligned} \quad (14)$$

$$d_j = \sqrt{(x_i - X_j)^2 + (y_i - Y_j)^2}$$

$$d_k = \sqrt{(x_i - X_k)^2 + (y_i - Y_k)^2 + (z_i - Z_k)^2}$$

$$C_4(P_i) = C_{v1}(P_i) + C_{v2}(P_i) + C_{v3}(P_i)$$

其中, C_{v1} 表示由 m 个地面威胁所造成的总威胁代价, C_{v2} 代表了由 s 个空中威胁所造成的总代价, C_{v3} 为路径点低于其附近高程点 Z_p 的总代价。第 j 个地面威胁半径 R_j 的圆心为 (X_j, Y_j) , 第 k 个空中威胁半径 R_k 的球心为 (X_k, Y_k, Z_k) 。若无人机到障碍物圆心的距离 d_j 小于地面威胁半径 R_j 时, 则施加相应的地面威胁代价; 同理, 若无人机 S 距离球心的距离 d_k 小于球体半径 R_k , 则施加相应的空中威胁。

2 改进樽海鞘群算法

在无人机航迹规划中, 优化算法的性能对无人机航迹规划的有效性和安全性破为重要。为此, 本文使用了多种策略对传统樽海鞘群算法加以改进。

2.1 基本樽海鞘群算法

樽海鞘群算法的灵感来源于樽海鞘的聚集行为, 模拟了其觅食过程和运动。与大多数启发式算法相同, 樽海鞘在搜索空间范围内随机初始化种群, 其公式如式(15)所示。

$$x_j^i = rand(N, D) \times (ub(j) - lb(j)) + lb(j) \quad (15)$$

其中, N 为种群数量, D 为搜索空间维度。

领导者寻找食物源位置并带领追随者跟随移动。领导者的位置更新公式如式(16)所示。

$$x_j^i = \begin{cases} F_j + c_1(c_2(ub_j - lb_j) + lb_j) & c_3 \geq 0.5 \\ F_j - c_1(c_2(ub_j - lb_j) + lb_j) & c_3 < 0.5 \end{cases} \quad (16)$$

其中, x_j^i 表示第 i 只樽海鞘在第 j 维中的位置, F_j 表示食物源在第 j 维中的位置, ub_j 和 lb_j 为第 j 维的上下界, c_1 为平衡勘探与开发的系数, c_2, c_3 为 $[0, 1]$ 区间均匀分布的随机数, c_1 的动态更新公式如式(17)所示。

$$c_1 = 2e^{-\left(\frac{4t}{T}\right)^2} \quad (17)$$

其中, t 表示当前迭代次数, T 表示最大迭代次数。

跟随者的更新公式由牛顿运动定律简化而成, 根据式(18)可知, 追随者的新位置可以表示为第 j 维上第 $i-1$ 个跟随者的位置和第 i 个位置的平均值。

$$x_j^i = \frac{1}{2}(x_j^{i-1} + x_j^i), i > 2 \quad (18)$$

2.2 优化策略

2.2.1 自适应 Alpha 稳定分布策略

由式(16)可知, 樽海鞘群的领航者由于受限于食物源位置而缺乏勘探能力, 并且其寻优过程依赖于随机数, 这可能导致算法在搜索过程中错失最优解, 影响寻优效果。因此, 引入自适应 Alpha 稳定分布策略以增强算法搜索能力, 降低算法搜索停滞可能性。稳定分布 (Levy Stable Distribution, LSD) 是一类概率分布, 也称分形分布^[25]。稳定分布有两大特例, 当稳定性指数 $\alpha = 2$ 时, 稳定分布为正态分布, 其步长分布也被称为高斯飞行。它的一个重要特性

是有一个有限的二阶矩(方差), 这意味着它的波动是有限且可预测的。当 $\alpha = 1$ 时, 对应的分布通常称为柯西分布。综上所述, 并结合图 7(a) 和 (b) 可以看出, α 越小, 分布的尾部越厚, 出现极端值的概率也越大。

随机数生成过程如下: 首先, 生成两个独立的随机变量 V 和 W , V 为区间 $[-\pi/2, \pi/2]$ 上均匀分布的随机变量, W 为均值为 1 的指数分布变量; 其次, 根据 V 和 W 生成服从 $S(\alpha, \beta, 1, 0)$ 的随机变量 X , 如式(19)所示。

$$X = \begin{cases} S_{\alpha, \beta} \times \frac{\sin(\alpha(V + B_{\alpha, \beta}))}{(\cos(V))^{\frac{1}{\alpha}}} \times \left(\frac{\cos(V - \alpha(V + B_{\alpha, \beta}))}{W} \right)^{\frac{1-\alpha}{\alpha}} & \alpha \neq 1 \\ \frac{2}{\pi} \left[\left(\frac{\pi}{2} + \beta V \right) \tan(V) - \beta \log \left(\frac{W \cos(V)}{\frac{\pi}{2} + \beta} \right) \right] & \alpha = 1 \end{cases} \quad (19)$$

其中, α 为稳定指数, 控制尾部厚度, $0 < \alpha < 2$; β 为偏斜参数, 控制分布的偏斜性。 $S_{\alpha, \beta}$ 与 $B_{\alpha, \beta}$ 的定义如式(20)所示。

$$S_{\alpha, \beta} = \left[1 + \beta^2 \tan^2 \left(\frac{\pi \alpha}{2} \right) \right]^{\frac{1}{2\alpha}} \quad (20)$$

$$B_{\alpha, \beta} = \frac{\arctan \left(\beta \tan \left(\frac{\pi \alpha}{2} \right) \right)}{\alpha} \quad (21)$$

最后, 根据随机变量 X 生成服从 $S(\alpha, \beta, \sigma, \mu)$ 的随机变量 Y , 如式(22)所示。

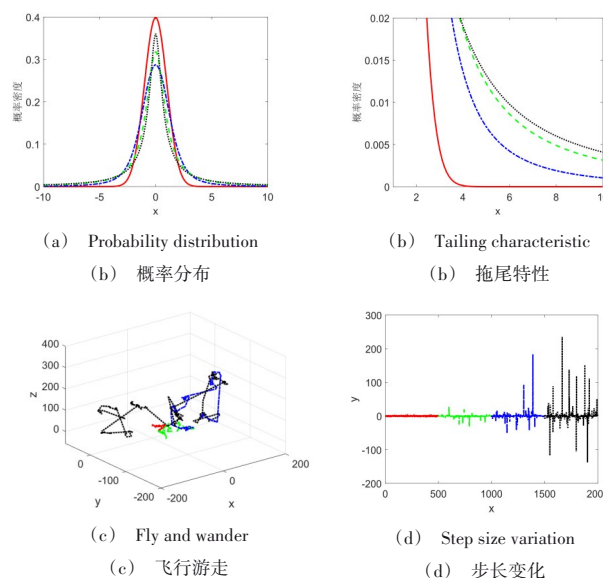


Fig. 7 Probability distributions and jump comparison of different α values

图 7 不同 α 值的概率分布和跳变比较

$$Y = \begin{cases} \sigma X + \mu & \alpha \neq 1 \\ \sigma X + \frac{2}{\pi} \beta \sigma \log \sigma + \mu & \alpha = 1 \end{cases} \quad (22)$$

其中, σ 为尺度参数, 控制分布的尺度或扩展程度; μ 为位置参数, 控制分布的中心位置。本文不考虑偏斜参数 β 和位置参数 μ , 故将其设置为 0, 而尺度参数 σ 设置为 1。

通过调节 α 值, 并由图 7(c) 和 (d) 可以看出, 当 α 较小时, 跳跃频率增加, 步长也相应增大, 使得其有更大的可能性跳出局部最优。改进领导者更新公式如式 (23) 所示。

$$x_j^1 = \begin{cases} F_j + c_1 \times LSD(\alpha_1) \times (c_2 \times (ub_j - lb_j) + lb_j) & 0 < t < T/2 \\ F_j - c_1 \times LSD(\alpha_2) \times (c_2 \times (ub_j - lb_j) + lb_j) & T/2 < t < T \end{cases} \quad (23)$$

其中, 斜率递减的 α_1 和斜率递增的 α_2 可以表示为如式 (24) 所示。

$$\begin{cases} \alpha_1 = 1.8 \times \left(1 - \sin\left(\frac{\pi \times t}{6 \times T}\right) \right) + C \\ \alpha_2 = 1.8 \times \left(\sin\left(\frac{\pi \times t}{6 \times T}\right) + 0.5 \right) + C \end{cases} \quad (24)$$

其中, C 为 $[-0.2, 0.2]$ 区间上生成的随机数。稳定指数 α_1 和 α_2 随迭代次数的变化曲线如图 8 所示。

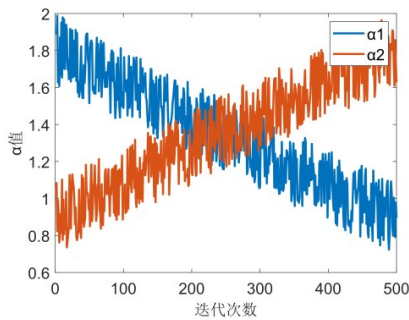


Fig. 8 α value fluctuation curve

图8 α 值波动曲线

为了增强勘探能力, 针对樽海鞘群算法在迭代过程中不同的寻优策略, 在算法前半段迭代周期内, 通过递减 α_1 值, 使其更偏向于长跳跃, 这有助于全局搜索和勘探新的解空间。而在后半段迭代周期, 逐步增加 α_2 值, 使其趋向于 2, 从而缩短飞行步长, 促进局部搜索和开发, 集中优化已经发现的潜在解。这种设计能够在算法初期更有效地勘探广阔的解空间, 并在后期对发现的优良区域进行充分开发, 从而提升整体优化性能。

2.2.2 非线性扰动策略

合适的非线性因子能够有效平衡算法全局搜索与局部开发过程, 提升算法在目标优化函数上的收敛效率和寻优精度。在迭代过程中, 跟随者的位置更新受其前后位置共同决定。因此, 当领导者的勘探能力减弱时, 追随者的作用也会因此显著减弱。为增强追随者跳出局部最优的能力, 引入了非线性扰动以改善这一问题。该非线性扰动

在前期同样进行大幅度跳跃, 而在算法后期, 参数逐渐收敛以进行局部搜索。通过与自适应 Alpha 稳定分布策略相配合, 达到更好的开发与探索能力平衡。跟随者位置更新公式和非线性因子 δ 分别如式 (24) 和式 (25) 所示。

$$\begin{aligned} x_j^i &= c_4 \times (x_j^{i-1} + x_j^i) / 2 + \delta \times (F - x_j^{i1}) + u_1 \times (x_j^{i2} - x_j^{i3}) \quad (25) \\ \begin{cases} \delta = (1 - 3 \times \text{rand} \times u_2 \times (1 - \log(1 + t/T)))^2 & \text{rand} \leq 0.5 \\ \delta = 3 \times \text{rand} \times u_2 \times (1 - \log(1 + t/T))^2 & \text{rand} > 0.5 \end{cases} \quad (26) \end{aligned}$$

其中, c_4 表示 $[0, 1]$ 区间均匀分布的随机数, u_1 和 u_2 是 $[-1, 1]$ 区间呈均匀分布的随机数, x_j^m 为搜索空间中的随机樽海鞘。

2.2.3 贪婪选择策略

樽海鞘群算法根据式 (23) 和式 (25) 更新策略迭代更新后生成的新种群, 仅挑选适应度最优值的种群作为食物来源, 这未充分利用其他种群在迭代更新中的潜力, 不利于完全发挥领航者与跟随者的其他种群在全局和局部搜索中的能力。

为了改善这一情况, 根据微分进化算法 (DE) 中的贪婪选择策略, 在生成候选解后, 对每个种群, 比较其当前解和候选解的适应度值。如果候选解的适应度优于当前解, 则将候选种群替换当前种群, 具体表达如式 (27) 所示。

$$x_i(l+1) = \begin{cases} x_i(t+1), f(x_i(t+1)) < f(x_i(t)) \\ x_i(t), f(x_i(t+1)) \geq f(x_i(t)) \end{cases} \quad (27)$$

2.2.4 鱼类聚集装置策略

随着迭代次数的增加, 由于各阶段采用不同的寻优策略, 而这些策略均一定程度上围绕当前最优食物源位置进行寻优, 使得樽海鞘群个体逐渐趋于一致, 很容易因算法自身局限性陷入局部最优解。受文献 [26] 的启发, 引入了鱼类聚集装置 (FADs) 策略, 该策略不受食物源影响, 因而能够高效地探索空间。当随机数小于 FADs 时, 算法基于范围约束进行变异, 根据每个个体位置的约束进行小范围随机移动, 以提高搜索效率。当随机数大于 FADs 时, 算法根据个体之间的相对位置差异进行探索, 以防止陷入局部最优。在每次迭代结束前, 算法会利用 FADs 策略引入新的候选种群, 以提升整体探索和寻优能力。该策略的数学过程描述如式 (28) 所示。

$$x_j^i = \begin{cases} x_j^i + CF \times (lb_j + c_5 \times (ub_j - lb_j)) \times U_j^i & c_6 \leq FADs \\ x_j^i + (c_7 + FADs \times (1 - c_6)) \times (x_j^{i4} - x_j^{i5}) & c_6 > FADs \end{cases} \quad (28)$$

其中, $FADs$ 表示优化过程中所影响的效率, 取值为 0.2。 U 为二进制的向量, c_5, c_6, c_7 为 $[0, 1]$ 区间均匀分布的随机数, x_j^{i4}, x_j^{i5} 表示空间中其他种群里第 j 维的位置, CF 为控制移动步长的自适应参数, 其公式如式 (29) 所示。

$$CF = (1 - t/T)^{2 \times //T} \quad (29)$$

2.3 算法实现步骤与流程

改进樽海鞘群算法(Improved Salp Swarm Algorithm, ISSA)实现步骤如下:

Step 1: 参数初始化,设置种群规模 N 、搜索空间维度 d 、当前迭代次数 t 、最大迭代次数 T 和变量取值范围。

Step 2: 初始化所有种群位置,计算种群适应度值,并对适应度值进行排序,选择最优种群作为食物位置 F 。

Step 3: 判断算法是否达到最大迭代次数,如果满足则保存并输出最优食物源 F 的位置与适应度值,算法结束;否则,继续进行下一步。

Step 4: 根据式(23)计算稳定指数 α_1 与 α_2 ,并根据式(19)~式(21)计算所生成的一个 N 行 d 列的矩阵,通过式(25)、式(28)更新系数 δ 、 CF 的值,同时生成 $[-1, 1]$ 区间的随机数 u_1 和 u_2 。

Step 5: 根据式(23)更新领导者位置,根据式(25)更新追随者位置,并依据式(27)进行贪婪选择。

Step 6: 依据式(28)引入新的候选种群,生成新的 N 个樽海鞘,并将适应度最优位置作为食物 F 。

Step 7: 判断算法若未达到最大迭代次数 T ,则返回Step 4;否则,记录全局最优位置及其适应度值,并结束算法。

改进樽海鞘群算法流程如图9所示。

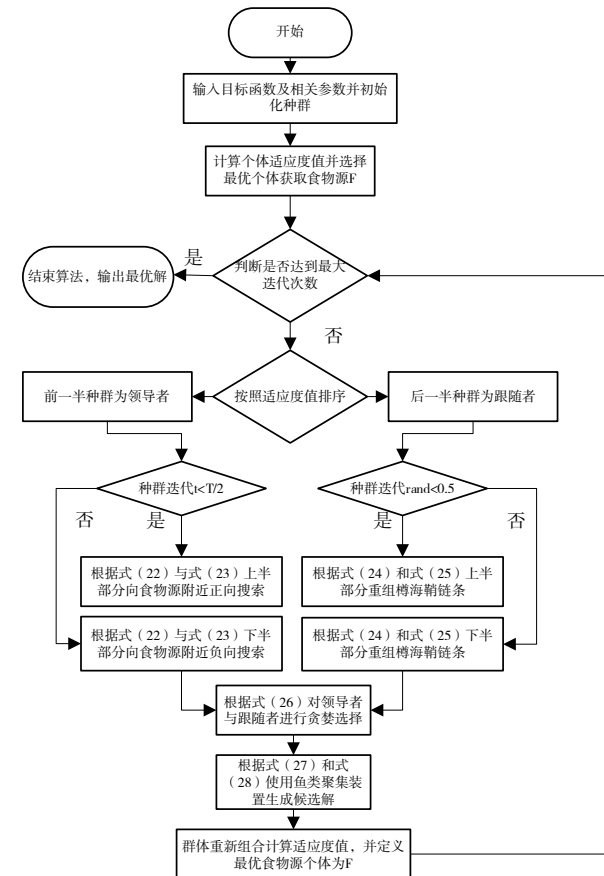


Fig. 9 Improved salp swarm algorithm (ISSA) optimization process

图9 改进樽海鞘群算法寻优流程

3 实验仿真及验证

本文实验在 Windows 10 x64 操作系统下运行,处理器为 Intel(R) Core(TM) i5-8250U,主频 1.60 GHz,8 GB 内存,软件平台使用 MATLAB2023b。首先,采用 CEC2020 基准测试函数,对变种樽海鞘群算法和其他优化算法进行实验比较,以验证改进算法的有效性;其次,建立城市与山区的基准地形并引入空中与地面的综合威胁,通过与其他算法比较,验证改进算法在航迹规划实验中的竞争力。

3.1 算法性能测试

本文使用了完整的 CEC2020 测试函数以验证改进算法的有效性。由于在测试套件中加入了偏移和旋转操作,故其最优解不再是 0,如表 1 所示。

Table 1 Test functions

表 1 测试函数

函数类型	函数 标号	函数名称	理论最 优值
单峰函数	F1	Shifted and Rotated Bent Cigar Function	100
	F2	Shifted and Rotated Schwefel's Function	1 100
基本函数	F3	Shifted and Rotated Lunacek bi-Rastrigin Function	700
	F4	Expanded Rosenbrock's plus Griewangk's Function	1 900
	F5	Hybrid Function 1 (N = 3)	1 700
混合函数	F6	Hybrid Function 2 (N = 4)	1 600
	F7	Hybrid Function 3 (N = 5)	2 100
	F8	Composition Function 1 (N = 3)	2 200
复合函数	F9	Composition Function 2 (N = 4)	2 400
	F10	Composition Function 3 (N = 5)	2 500

本文使用哈里斯鹰优化算法(HHO)^[27]、海洋捕食者算法(MPA)^[26]、基本樽海鞘群算法(SSA)^[17]、疯狂自适应樽海鞘群算法(CASSA)^[28]和差异演化寄生樽海鞘群算法(PDESSA)^[29]进行实验对比,以上算法的参数设置与原文保持一致。算法种群数量设置为 30,迭代次数为 1 000 次,维数为 20。同时,记录 6 种算法所得解的平均值(Ave)、标准差(Std)与总的最优值(Best)。测试结果如表 2 所示。

可以看出,单峰函数 $f1$ 具有唯一的全局最优解,因而适合评估算法开发能力。ISSA 算法相比于其他算法取得了更优效果。基本函数和混合函数 $f2$ 到 $f7$ 包含多个局部最优解,并且随着维度的增加,这些局部最优解的数量呈现出指数级增长,适合评估算法探索能力。ISSA 算法除 $f2$ 和 $f6$ 的标准差外,均表现出了很好的性能。复合函数 $f8$ 到 $f10$ 旨在测试算法避免陷入局部最优解的能力,同时兼顾探索和开发能力,ISSA 算法也优于其他 5 种算法。总体来看,改进策略整体表现出很好的寻优性能和鲁棒性,验证了其在应对复杂优化问题时的有效性。

图 10 展示出了 6 种算法在测试函数上的适应度曲线。可以看出,除 $f4$ 和 $f10$ 外,算法的综合寻优精度均优于其他算法。同时,从收敛速度分析,算法整体收敛速度均快于其余 5 种算法,充分验证了改进算法寻优性能的有效性。

Table 2 Comparison of test functions

表2 测试函数比较

函数	指标	HHO	MPA	SSA	CASSA	PDESSA	ISSA
f1	平均值	5.33E+06	4.57E+03	3.85E+03	4.11E+03	1.27E+10	3.05E+03
	标准差	2.09E+06	3.65E+03	3.93E+03	3.75E+03	3.84E+09	1.90E+03
	最优值	2.81E+06	2.71E+02	1.05E+02	2.10E+02	7.60E+09	1.03E+02
f2	平均值	3.30E+03	2.20E+03	3.41E+03	3.50E+03	5.27E+03	2.10E+03
	标准差	5.59E+02	1.94E+02	3.65E+02	5.96E+02	6.09E+02	2.89E+02
	最优值	2.40E+03	1.78E+03	2.94E+03	2.73E+03	3.92E+03	1.72E+03
f3	平均值	9.01E+02	7.51E+02	8.10E+02	8.15E+02	9.92E+02	7.50E+02
	标准差	3.14E+01	1.14E+01	3.99E+01	4.42E+01	4.33E+01	8.46E+00
	最优值	8.49E+02	7.38E+02	7.53E+02	7.60E+02	9.37E+02	7.38E+02
f4	平均值	1.92E+03	1.90E+03	1.91E+03	1.91E+03	6.57E+04	1.90E+03
	标准差	7.89E+00	6.88E-01	2.54E+00	1.92E+00	8.03E+04	5.43E-01
	最优值	1.91E+03	1.90E+03	1.90E+03	1.90E+03	5.01E+03	1.90E+03
f5	平均值	5.87E+05	3.97E+03	1.96E+05	1.35E+05	2.58E+06	2.05E+03
	标准差	2.17E+05	1.02E+02	1.22E+05	1.03E+05	1.47E+06	7.69E+01
	最优值	8.15E+04	3.89E+03	3.87E+04	4.10E+04	7.27E+05	1.85E+03
f6	平均值	2.37E+03	1.83E+03	2.12E+03	2.21E+03	2.98E+03	1.68E+03
	标准差	2.81E+02	6.17E+01	2.50E+02	2.52E+02	2.71E+02	6.50E+01
	最优值	1.99E+03	1.75E+03	1.76E+03	1.88E+03	2.52E+03	1.61E+03
f7	平均值	2.14E+05	4.25E+03	1.29E+05	1.00E+05	1.70E+06	2.24E+03
	标准差	2.36E+05	8.73E+01	1.57E+05	1.04E+05	7.24E+05	7.99E+01
	最优值	1.43E+04	4.15E+03	1.28E+04	9.72E+03	7.36E+05	2.11E+03
f8	平均值	3.51E+03	2.40E+03	2.67E+03	2.56E+03	3.99E+03	2.30E+03
	标准差	1.60E+03	7.85E-01	7.97E+02	8.24E+02	4.83E+02	5.38E-01
	最优值	2.31E+03	2.40E+03	2.30E+03	2.30E+03	3.43E+03	2.30E+03
f9	平均值	3.18E+03	2.87E+03	2.87E+03	2.87E+03	3.19E+03	2.83E+03
	标准差	1.60E+02	1.86E+01	1.78E+01	2.40E+01	8.65E+01	1.16E+01
	最优值	2.91E+03	2.85E+03	2.84E+03	2.84E+03	3.04E+03	2.82E+03
f10	平均值	2.98E+03	2.93E+03	2.94E+03	2.94E+03	4.10E+03	2.92E+03
	标准差	2.25E+01	3.09E+01	3.30E+01	3.05E+01	4.49E+02	1.19E+01
	最优值	2.94E+03	2.91E+03	2.91E+03	2.91E+03	3.44E+03	2.91E+03

3.2 算法时间复杂度分析

在基本樽海鞘群算法中,假设其总的迭代次数为 T ,种群规模为 N ,个体维度为 d ,设置初始化参数时间为 t_0 ,每一维上生成随机数的时间为 t_1 ,求适应度值的时间为 $f(n)$,对适应度值进行排序并找到当前最优位置的时间为 t_2 ,则初始化阶段的时间复杂度如式(30)所示。

$$T_1 = O(t_0 + N(f(n) + t_1 d) + t_2) = O(n + f(n)) \quad (30)$$

在领导者位置更新阶段,领导者数目为 $N/2$, c_1 的计算时间为 t_3 , c_2 和 c_3 与 t_1 生成的时间相同。由于两种情况的计算时间完全相同,故每一维更新时间为 t_4 ,则此阶段时间复杂度如式(31)所示。

$$T_2 = O(t_3 + (N/2)d(2t_1 + t_4)) = O(n) \quad (31)$$

对于跟随者,设其每一维更新时间为 t_5 ,其时间复杂度如式(32)所示。

$$T_3 = O((N/2)dt_5) = O(n) \quad (32)$$

在边界处理和食物源更新阶段,设其边界比较时间为 t_6 ,食物源的适应度计算和比较时间设置为 t_7 ,则 T_4 的计算公式如式(33)所示。

$$T_4 = O(N(dt_6 + t_7) + f(n)) = O(n + f(n)) \quad (33)$$

综上所述,其总时间复杂度如式(34)所示。

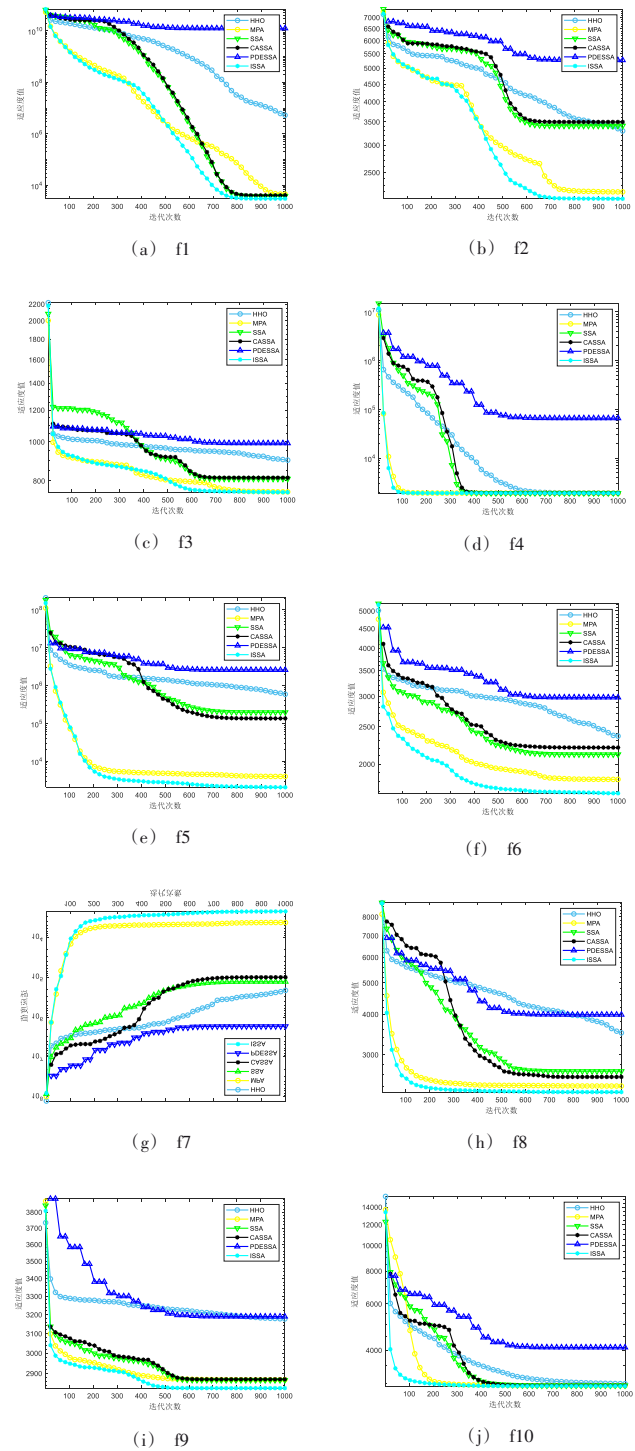


Fig. 10 Fitness curves

图10 适应度曲线

$$T(n) = T_1 + T(T_2 + T_3 + T_4) = O(n + f(n)) \quad (34)$$

在改进算法中,在初始化阶段和选定食物源位置阶段与基本樽海鞘群算法相同,如式(35)所示。

$$T_1 = T_1' \quad (35)$$

在领导者和跟随者阶段,假设其自适应稳定分布生成的步长时间为 t_8 , u_1 和 u_2 生成的随机数时间为 t_1 , δ 生成的时间假设为 t_9 ,贪婪选择时间复杂度为 $f(n)$,对比每个适应

度值的时间假设为 t_{10} , 则其总的时间复杂度如式 (36) 所示。

$$T_2' + T_3' = O(t_3 + (N/2)d(2t_1 + t_4 + 2t_8 + (N/2)d(t_5 + 2t_1 + t_9) + f(n) + Nt_{10})) = O(n + f(n)) \quad (36)$$

边界处理与食物源更新阶段的复杂度不变, 如式 (37) 所示。

$$T_4 = T_4' \quad (37)$$

鱼类聚集装置策略 c_4 、 c_5 和 c_6 生成随机数的时间为 t_1 , 自适应参数 CF 值的实现时间为 t_{10} , 设该策略每一维的运行时间为 t_{11} , 故其时间复杂度如式 (38) 所示。

$$T_5' = O((Nd(3t_1 + t_{11}) + t_{10})) = O(n) \quad (38)$$

因此, 总的运行复杂度如式 (39) 所示。

$$T_n = T_1 + T(T_2' + T_3' + T_4' + T_5') = O(n + f(n)) \quad (39)$$

可以看出, 改进樽海鞘群算法总的时间复杂度不变, 并未增加执行复杂度。

3.3 实验测试参数

实验统一采用种群数量 30, 最大迭代次数 300 次。为了降低实验的随机性, 总计实验 30 次, 并求统计结果的平均值、标准差、最优值和平均迭代曲线图, 并对实验结果进行分析。实验中使用的建模参数如表 3 所示。规划点总数 n 设计为地图最大距离的一半, 而 h_{safe} 值经过测试规定为地图最大设置高度的 1/10。

Table 3 Modeling parameters

表 3 建模参数

参数	数值	参数	数值
$\phi_{\max}$	45	$\theta_{\max}$	45
u_1	0.5	u_2	0.3
u_3	0.2		
山区地形		城市地形	
n	500	n	500/1 000
$L_{\max}$	3 000	$L_{\max}$	2 000/4 000
$h_{\max}$	1 000	$h_{\max}$	300/150
h_{safe}	100	h_{safe}	30/15

3.4 山区地形测试

山区地形实验测试部分采用了海拔高度相差 1 km 的山地地形, 起点坐标设置为 (30 m, 30 m, 200 m), 终点坐标设置为 (30 km, 30 km, 200 m)。在无人机路径规划过程中, 为了有效避开地面和空中的综合威胁, 设置了 10 个地面综合威胁, 5 个球体为空中综合威胁, 其中有 3 个球体在路径中间设置障碍, 两个球体在终点处制造障碍。

图 11 展示了各算法在实验中生成的最优适应度轨迹, 即最优路径。

图 12 和图 13 则分别展示了其俯视图与终点处路径的细节轨迹。可以看出, 所有算法的最优解均成功找到了可行路径到达终点。SSA、CASSA 与 PDESSA 算法的规划路径近似, 通过抬升飞行高度避开障碍物到达终点, 而 HHO、MPA 与 ISSA 算法则通过绕行找到一条合适的路径, 最终所有算法都经过两个球体的夹缝处到达终点。

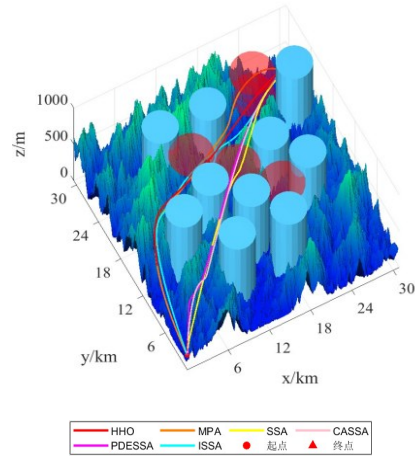


Fig. 11 Comparison of UAV flight path planning in mountain environment

图 11 山区环境无人机航迹规划对比

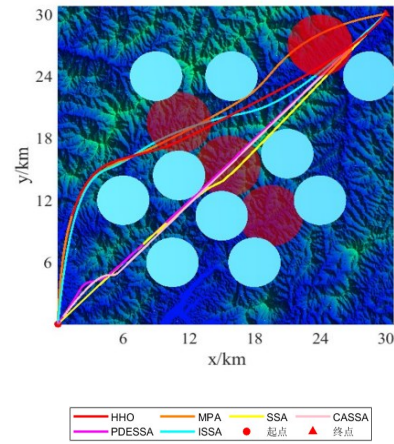


Fig. 12 Flight path planning top view comparison in mountain environment

图 12 山区环境航迹规划对比俯视图

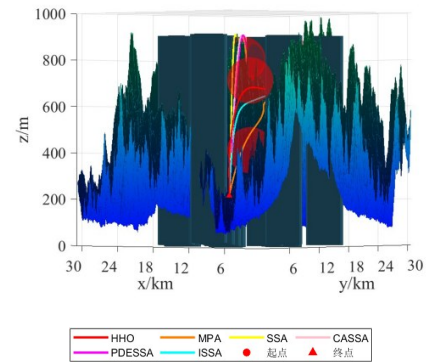


Fig. 13 Details of flight path planning at the end

图 13 终点处航迹规划细节

4 种算法的平均结果适应度迭代曲线如图 14 所示, 可以观察到 ISSA 算法的迭代平均值最小。由于种群初始化的随机性, 6 种算法在迭代前期生成的路径点与障碍物发生相交或低于高程值导致适应度值为 ∞ 。然而, 尽管 ISSA

算法在前期跳出障碍物碰撞的情况不是最快,但随着迭代次数的增加,ISSA 算法不断跳出局部最优,最终超越其他算法,这一现象证明了改进算法的有效性。

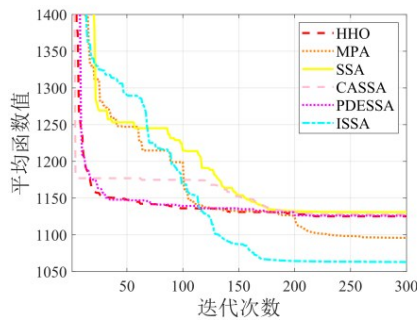


Fig. 14 Fitness iteration curve in mountain environment

图 14 山区环境适应度迭代曲线

表 4 列出了各算法在 30 次实验中的平均值、最小适应度值、航程代价、高度变化代价与平滑代价。结果表明,相较于其他算法,ISSA 算法在平均值和最优适应度方面表现更优。数据表明,ISSA 算法相比于 HHO、MPA、SSA、CASSA 和 PDESSA 在平均值上分别下降了 5.53%、3.00%、6.00%、5.73% 和 5.59%, 而最优值分别下降了 8.04%、1.05%、10.19%、9.13%、9.21%。各算法的真实航程需要乘以地形数据分辨率,分别为 60.67 km、57.58 km、63.53 km、63.05 km、62.59 km 和 57.42 km。

Table 4 Mountain model data

表 4 山区模型数据

算法	平均值	最优值	航程代价	高度变化代价	平滑代价
HHO	1 125.28	1 099.50	2 022.17	130.96	245.62
MPA	1 096.00	1 021.84	1 919.31	108.55	148.11
SSA	1 130.95	1 125.83	2 117.66	126.23	145.66
CASSA	1 127.75	1 112.74	2 101.71	101.69	156.90
PDESSA	1 126.07	1 113.74	2 086.28	121.08	171.41
ISSA	1 063.09	1 011.11	1 914.00	99.36	121.50

根据图 15 和表 4 数据,在没有障碍物的情况下,从 6 种算法生成的路径及其各项指标对应的代价可以看出,ISSA 算法相较于其他算法在路径长度上更短、高度变化更小且在平滑代价上更优,路径整体质量优于其他算法。

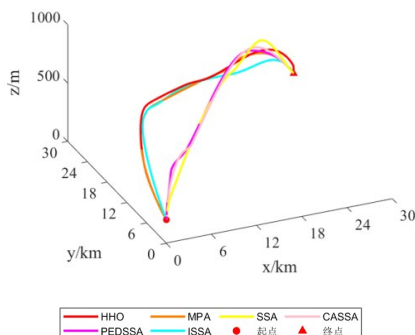


Fig. 15 UAV flight path comparison

图 15 无人机飞行路径对比

3.5 城市地形测试

为了测试运行框架与改进算法相结合的路径规划的有效性,在未添加禁飞区和障碍物的两种不同规模密集城市地形上进行测试,如图 16(a)和(c)所示。起点设置为 (1 m, 1 m, 1 m), 终点设置为 (1 000 m, 800 m, 1 m) 与 (2 000 m, 1 400 m, 1 m)。同时,为防止路径波动较大,提升路径整体质量,将凸包点选择为高度大于 50 m 的形式。可以看出,高程模型一中 MPA、SSA、CASSA 与 PDESSA 算法规划出的最优路径相似,而 HHO 算法则从另一侧平滑地到达终点,而高程模型二中所有算法的路径相似但路径波动情况不同。

图 16(b)和(d)为 6 种算法的适应度函数的迭代曲线。可以看出,ISSA 算法较快跳出局部最优,并整体上更快地获得了最优解。

由表 5 可以看出,高程模型一中 ISSA 算法的平均值相比于 HHO、MPA、SSA、CASSA 和 PDESSA 算法分别下降了 4.40%、0.60%、0.48%、2.69% 和 3.08%。而最优值与传统 SSA 算法基本相同,这在一定程度上表明传统 SSA 算法能够在没有障碍物环境下的高程地形上获得很好的搜索精度,而 ISSA 算法中的改进策略由于其跳出局部最优较强的特性,因而能够在整体寻优效果中获得更好的表现。其中,传统 SSA 算法在航程代价中表现最好,而 ISSA 算法在高度代价与平滑代价中表现最优,且 ISSA 算法在收敛速度方面显著快于其他算法。高程模型二中 ISSA 算法的平均值下降了 5.29%、1.17%、2.19%、2.96% 和 1.89%,且其余指标整体优于其他算法,随着地形复杂度的提升,算法也展现出了更强的适应性。

图 17 展示了 6 种算法在回归原始建筑模型二的终点处航迹规划路径细节。可以看出,所有算法生成的路径都有效地避开了障碍物进而到达终点,同时可以观察到,所提出的凸包策略能够有效地应用于城市复杂环境。综上所述,改进樽海鞘群算法在山区和城市地形测试中均表现优异。优化策略增强了算法跳出局部最优和局部搜索的能力,使得算法能够更好地应对不同环境变化所带来的挑战,并展现出更强的平衡性与适应性。

4 结语

本文为解决无人机在复杂地形的航迹规划问题,构建了基于真实地形的山区环境模型与城市环境模型,在基于高程信息的编码方式上,结合环境的高度信息引入凸包策略,有效提升了无人机在面对复杂地形中的效果与效率。并且,对樽海鞘群算法引入了自适应 Alpha 稳定分布策略、非线性扰动策略、贪婪选择策略与鱼类聚集装置策略,有效提高了算法开发能力和探索能力。仿真与实验结果表明,引入凸包策略的航迹编码方式使得所有算法均能找到一条合理且高度波动不大的路径,改进后的算法综合性能

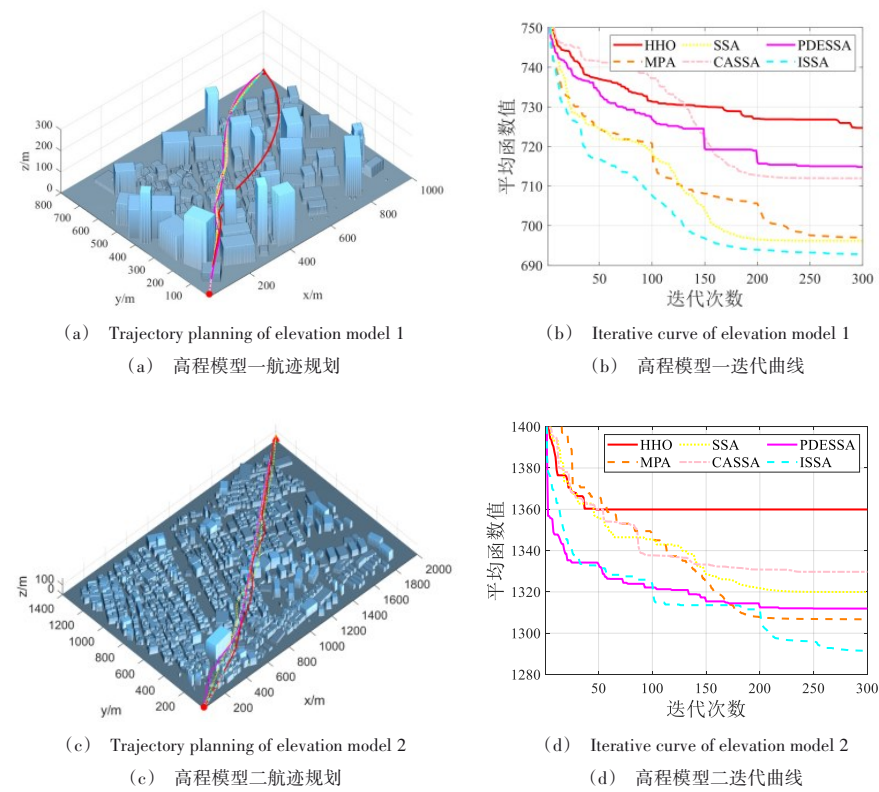


Fig. 16 Trajectory planning and fitness iteration curve
图 16 航迹规划及适应度迭代曲线

Table 5 Urban model data
表 5 城市模型数据

算法	高程模型一					高程模型二				
	平均值	最优值	航程代价	高度变化代价	平滑代价	平均值	最优值	航程代价	高度变化代价	平滑代价
HHO	724.69	704.38	1 360.67	20.42	89.60	1 359.91	1 322.38	2 583.84	15.72	128.75
MPA	696.99	681.37	1 327.07	16.98	63.70	1 306.73	1 302.03	2 558.63	14.27	92.19
SSA	696.16	681.01	1 325.43	18.63	63.54	1 319.97	1 304.39	2 563.59	12.19	94.67
CASSA	711.96	682.90	1 329.03	18.07	64.84	1 329.72	1 297.14	2 543.79	18.31	98.77
PDESSA	714.83	684.23	1 330.13	18.87	67.52	1 311.93	1 301.74	2 546.28	22.91	108.64
ISSA	692.80	680.99	1 327.01	16.60	62.54	1 291.53	1 289.36	2 535.47	11.73	90.53

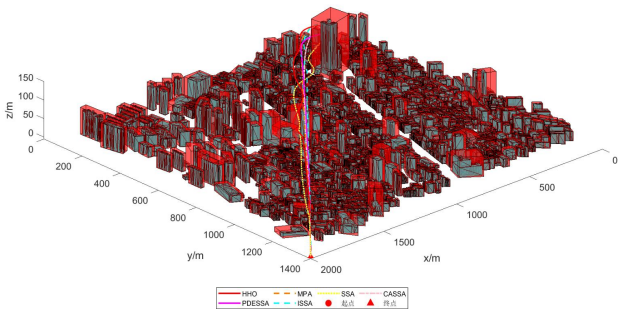


Fig. 17 Urban environment flight path planning details at the end point
图 17 终点处城市环境航迹规划细节

表现更好,能够规划出一条相对更优的路径。后续研究将扩展到多无人机系统,并尝试将其应用于例如灾区救援的特定场景中。

参考文献:

[1] GUO Q M,ZHANG P,WANG Y Y,et al. Review of intelligent algorithms for autonomous path planning of UAVs[J]. Aero Weaponry, 2023, 30(3): 29-40.
郭启敏,张鹏,王应洋,等. 无人机自主航迹规划智能算法综述[J]. 航空兵器,2023,30(3):29-40.

[2] ALSHAMMREI S,BOUBAKER S,KOLSI L. Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance[J]. Comput Mater Contin,2022,72:5939-5954.

[3] CHEN J W,BAO C C,ZHAO Z H. Research on route planning of logistics UAV based on improved A* algorithm[J]. Computer Engineering and Applications, 2021, 57(8): 231-27.
陈继伟,包长春,赵子恒. 基于改进A*算法的物流无人机航迹规划研究[J]. 软件导刊,2023,22(11):123-128.

[4] WANG Q L,WU F G,ZHENG C C,et al. UAV path planning based on optimized artificial potential field method[J]. Systems Engineering and Electronics, 2023, 45(5): 1461-1468.
王庆禄,吴冯国,郑成辰,等. 基于优化人工势场法的无人机航迹规划

- [J]. 系统工程与电子技术, 2023, 45(5): 1461-1468.
- [5] XIN P, WANG X, LIU X, et al. Improved bidirectional RRT* algorithm for robot path planning[J]. Sensors, 2023, 23(2): 1041.
- [6] SHIN J J, BANG H. UAV path planning under dynamic threats using an improved PSO algorithm[J]. International Journal of Aerospace Engineering, 2020(1): 8820284.
- [7] WAN X F, HU W, FANG W Y, et al. Research on path planning of robot based on improved ant colony algorithm[J]. Computer Engineering and Applications, 2014, 50(18): 63-66.
万晓凤, 胡伟, 方武义, 等. 基于改进蚁群算法的机器人路径规划研究[J]. 计算机工程与应用, 2014, 50(18): 63-66.
- [8] LIU C A, WANG X P, LIU C Y, et al. Three-dimensional route planning for unmanned aerial vehicle based on improved grey wolf optimizer algorithm[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2017, 45(10): 38-42.
柳长安, 王晓鹏, 刘春阳, 等. 基于改进灰狼优化算法的无人机三维航迹规划[J]. 华中科技大学学报(自然科学版), 2017, 45(10): 38-42.
- [9] WU K, TAN S C. Path planning of UAVs based on improved whale optimization algorithm[J]. Acta Aeronautica et Astronautica Sinica, 2020, 41(S2): 107-114.
吴坤, 谭劲昌. 基于改进鲸鱼优化算法的无人机航路规划[J]. 航空学报, 2020, 41(S2): 107-114.
- [10] PHUNG M D, HA Q P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization[J]. Applied Soft Computing, 2021, 107: 107376.
- [11] HOU P J, LIU J, NI F, et al. Multi-scene and multi-obstacle UAV 3D path planning based on improved seagull optimization algorithm[J]. Computer engineering and design[J]. Software Guide, 2024, 23(5): 44-51.
侯平静, 刘姜, 倪枫, 等. 基于改进海鸥优化算法的多场景多障碍无人机三维路径规划[J]. 软件导刊, 2024, 23(5): 44-51.
- [12] HUANG J, LI Y F, WANG S C, et al. 3D path planning of UAVs in urban environments based on improved PSO algorithm[J]. Electronics Optics & Control, 2024, 31(2): 41-45, 76.
黄晋, 李云飞, 王圣淳, 等. 基于改进PSO算法的无人机城域三维路径规划[J]. 光电与控制, 2024, 31(2): 41-45, 76.
- [13] YU X, JIANG N, WANG X, et al. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning[J]. Expert Systems with Applications, 2023, 215: 119327.
- [14] HU G, HUANG F, SEYYEDABBASI A, et al. Enhanced multi-strategy bottlenose dolphin optimizer for UAVs path planning[J]. Applied Mathematical Modelling, 2024, 130: 243-271.
- [15] WEI Y, WEI C B. Convex-hull based deployment of UAV algorithm in unmanned aerial vehicle rescue system in disaster area[J]. Journal of Projectiles, Rockets, Missiles and Guidance, 2022, 42(6): 26-31.
魏妍, 魏长宝. 灾区无人机救援系统中基于凸包理论的无人机部署算法[J]. 弹箭与制导学报, 2022, 42(6): 26-31.
- [16] LI G X, LYU C, PENG R C, et al. A rapid path planning algorithm based on the bilateral convex-hull expanding model[J]. Geomatics and Information Science of Wuhan University, 2021, 46(1): 58-64.
李改肖, 吕程, 彭认灿, 等. 一种利用双侧凸包扩张模型的路径快速规划算法[J]. 武汉大学学报(信息科学版), 2021, 46(1): 58-64.
- [17] MIRJALILI S, GANDOMI A H, et al. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114: 163-191.
- [18] XING Z K, JIA H M, SONG W L. Levy flight trajectory-based salp swarm algorithm for multilevel thresholding image segmentation[J]. Acta Automatica Sinica, 2021, 47(2): 363-377.
邢致恺, 贾鹤鸣, 宋文龙. 基于莱维飞行樽海鞘群优化算法的多阈值图像分割[J]. 自动化学报, 2021, 47(2): 363-377.
- [19] ZHANG Z, ZHANG S J, RAO S H, et al. Multi-parameter identification of permanent magnet synchronous motor based on improved salp swarm algorithm[J]. Electric Machines and Control, 2022, 26(8): 139-146.
张铸, 张仕杰, 饶盛华, 等. 改进樽海鞘群算法的永磁同步电机多参数辨识[J]. 电机与控制学报, 2022, 26(8): 139-146.
- [20] ZHAO W C, GUO P, WANG H B, et al. Improved slap swarm algorithm for scheduling of flexible job shop[J]. CAAI Transactions on Intelligent Systems, 2022, 17(2): 376-386.
赵文超, 郭鹏, 王海波, 等. 改进樽海鞘群算法求解柔性作业车间调度问题[J]. 智能系统学报, 2022, 17(2): 376-386.
- [21] ZHOU X, ZOU H. Adaptive salp swarm algorithm with golden sine algorithm and hybrid mutation[J]. Computer Engineering and Applications, 2021, 57(12): 75-85.
周新, 邹海. 融合黄金正弦混合变异的自适应樽海鞘群算法[J]. 计算机工程与应用, 2021, 57(12): 75-85.
- [22] WANG Z S, DING H W, WANG J, et al. Salp swarm algorithm based on orthogonal refracted opposition-based learning[J]. Journal of Harbin Institute of Technology, 2022, 54(11): 122-136.
王宗山, 丁洪伟, 王杰, 等. 基于正交设计的折射反向学习樽海鞘群算法[J]. 哈尔滨工业大学学报, 2022, 54(11): 122-136.
- [23] LIANG C L, CHEN Z H. Improved salp swarm algorithm based on hybrid strategy[J]. Control and Decision, 2024, 39(8): 2541-2550.
梁成龙, 陈志环. 基于混合策略改进的樽海鞘群算法[J]. 控制与决策, 2024, 39(8): 2541-2550.
- [24] XU X, XIE C, LUO Z, et al. A multi-objective evolutionary algorithm based on dimension exploration and discrepancy evolution for UAV path planning problem[J]. Information Sciences, 2024, 657: 119977.
- [25] LYU T Z, LI Z. Novel particle swarm optimization algorithm based on fractional calculus and Alpha-stable distribution[J]. Computer Science, 2014, 41(7): 246-249, 289.
吕太之, 李卓. 基于动态分数阶和Alpha稳定分布的粒子群优化算法[J]. 计算机科学, 2014, 41(7): 246-249, 289.
- [26] FARAMARZI A, HEIDARINEJAD M, MIRJALILI S, et al. Marine predators algorithm: a nature-inspired metaheuristic[J]. Expert Systems with Applications, 2020, 152: 113377.
- [27] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris hawks optimization: algorithm and applications[J]. Future Generation Computer Systems, 2019, 97: 849-872.
- [28] ZHANG D M, CHEN Z Y, XIN Z Y, et al. Salp swarm algorithm based on craziness and adaptive[J]. Control and Decision, 2020, 35(9): 2112-2120.
张达敏, 陈忠云, 辛梓芸, 等. 基于疯狂自适应的樽海鞘群算法[J]. 控制与决策, 2020, 35(9): 2112-2120.
- [29] LIU J S, YUAN M M, LI Y. Robot path planning based on improved salp swarm algorithm[J]. Journal of Computer Research and Development, 2022, 59(6): 1297-1314.
刘景森, 袁蒙蒙, 李煜. 基于改进的樽海鞘群算法求解机器人路径规划问题[J]. 计算机研究与发展, 2022, 59(6): 1297-1314.

(责任编辑: 孙娟)