

多策略改进蜣螂优化算法的无人机航迹规划^{*}梅雨琳¹ 曲良东² 饶爽¹

(1. 广西民族大学物理与电子信息学院 南宁 530006; 2. 广西民族大学人工智能学院 南宁 530006)

摘要: 针对蜣螂优化算法存在陷入局部最优、全局搜索能力不足致使无人机三维航迹规划效果不佳的问题,设计了一种多策略改进的蜣螂优化算法。通过构建三维空间模型,结合路径长度、威胁、高度和平滑度等因素,构建了综合评价函数。首先,采用混合混沌序列提升初始种群多样性;其次,在蜣螂滚球阶段引入“差分变异”算子以提升算法的局部搜索能力,并结合改进的正弦算法,通过概率切换机制进行个体更新,进一步提升算法的全局搜索性能;最后,在繁殖阶段引入了改进的螺旋搜索策略,增强算法跳出局部最优的能力。通过对6个基准函数的优化对比分析并展示粒子在搜索空间中的运动轨迹,结果表明改进后的算法在收敛速度、精确度和鲁棒性方面表现更优。将算法应用于三维无人机路径规划中,路径长度的最优值、平均值和最差值分别提升了0.41%、5.67%和18.03%,进一步验证了改进策略的有效性以及该算法在处理实际工程应用中的优越性。

关键词: 蜣螂优化算法;混合混沌映射;多策略引导机制;改进螺旋搜索策略;三维无人机路径规划

中图分类号: TP18; TN96 **文献标识码:** A **国家标准学科分类代码:** 520.1040

Multi-strategy improved dung beetle optimization algorithm
for UAV path planningMei Yulin¹ Qu Liangdong² Rao Shuang¹

(1. College of Physics & Electronic Information, Guangxi Minzu University, Nanning 530006, China; 2. School of Artificial Intelligence, Guangxi Minzu University, Nanning 530006, China)

Abstract: To address the issues in the dung beetle optimization algorithm, such as falling into local optima and insufficient global search capability, which lead to suboptimal performance in 3D UAV path planning, a multi-strategy improved dung beetle optimization algorithm was designed. A 3D spatial model was constructed, and a comprehensive evaluation function was developed by considering factors such as path length, threat, altitude, and smoothness. First, a hybrid chaotic sequence was employed to enhance the initial population diversity. Then, during the dung beetle rolling stage, a “differential mutation” operator was introduced to improve the algorithm’s local search ability. This was combined with an improved sine algorithm to update individuals via a probability switching mechanism, further enhancing the global search capability. Finally, an improved spiral search strategy was incorporated during the breeding stage to strengthen the algorithm’s ability to escape local optima. Through optimization of six benchmark functions and analysis of particle motion trajectories in the search space, the results demonstrated that the improved algorithm performed better in terms of convergence speed, accuracy, and robustness. When applied to 3D UAV path planning, the optimal, average, and worst values of path length improved by 0.41%, 5.67%, and 18.03%, respectively, further validating the effectiveness of the improvement strategies and the superiority of this algorithm in practical engineering applications.

Keywords: dung beetle optimization algorithm; hybrid chaotic mapping; multi-strategy guidance mechanism; improved spiral search strategy; 3D UAV path planning

0 引言

伴随科技快速进步,无人机(unmanned aerial vehicle,

UAV)^[1]因其在复杂和危险环境中的工作能力而受到广泛关注,应用于军事、农业和消防等领域。无人机通过无线电遥控和内部程序进行操控对目标检测^[2],相较于载人航天

器,其体积小、成本低、操作简便、适应性强,且在战场上的生存能力更强。随着无人机技术的快速发展,其在军事和民用领域的应用日益增多。中国也在积极开展无人机研究,但低空空域的开放使得飞行安全问题愈发严重,引起了全社会的关注。由于无人机数量激增且任务环境复杂,避免碰撞和提高工作效率,让无人机飞行轨迹成为可预见可控制的成为首要任务。因此,解决无人机路径规划问题^[3]具有重要意义。传统的启发式路径规划方法在三维复杂环境中的搜索速度慢、收敛精度低,无法快速准确地提供短且无碰撞的合理路径,从而推动了基于智能优化的路径规划算法的研究。

当前,三维路径规划方法已取得许多研究成果,传统路径规划方法有 A* 算法^[4]、快速探索随机树算法^[5]、人工势场法^[6]、Dijkstra 算法^[7]等,该类方法由于需要大量计算时间和资源,对于地形变化较多、障碍类型多样的战场环境而言大多难以适用。元启发式算法是仿照自然行为的群智能算法,具有计算速度快、计算逻辑简单、算法灵活性强的特点,正被越来越多地应用于解决复杂空间中的多无人机三维路径规划问题。为此,国内外许多学者对智能优化算法进行了大量研究。吴学礼等^[8]针对麻雀算法易陷入局部最优、收敛精度低等问题,对种群中的发现者加入动态权重因子同时引入高斯变异,其在算法后期加快了收敛速度,但收敛精度还有待提高;Song 等^[9]通过 PSO 算法结合连续的高次贝塞尔曲线来规划移动机器人的平滑路径,使其更符合机器人的运动学模型,但是其易陷入局部最优解的情况并没有得到改善;张鹭等^[10]通过将角度导向因子引入状态转移规则中,使蚂蚁以更大的概率朝着目标点的方向前进,提高了路径的搜索效率。引入启发式信息自适应调整因子平衡了算法的收敛性和全局搜索能力。结果表明算法鲁棒性及跳出局部最优能力得到提升,但是算法的收敛时间还有待缩短;赵娟等^[11]为了实现无人机从特定方向接近和打击目标点,提出了基于启发点引导 D* 算法扩展的航迹规划方法,同时缩短了无人机航机路径减少了飞行耗时。

在上述研究背景下,针对复杂地形环境中的三维无人机路径规划问题,以最小化的路径长度、飞行高度和转弯角度为综合成本,通过优化飞行路径来提升无人机的飞行效率和稳定性。为解决蜣螂优化算法(dung beetle optimizer, DBO)初始种群分布不均、全局搜索能力弱、易陷入局部收敛的问题,提出了一种融合多策略的蜣螂优化算法(MGDBO)。

1 环境建模

1.1 三维环境建模

无人机飞行路线和地理环境的联系密切相关,故将障碍物区域处理成天然山体形状,通过函数模拟法对其进行建模,使用地形常数以及山峰模型来构建地形地图,地形常数如式(1)所示。

$$z = \sin(y+a) + b \times \sin(x) + c \times \cos(d(y^2+x^2)) + e \times \cos(y) + f \times \sin(f(y^2+x^2)) + g \times \cos(y) \quad (1)$$

式中: x 、 y 为地图上的坐标点, z 对应的是该坐标的高, a 、 b 、 c 、 d 、 e 、 f 为地形常数会影响地形的形状。对地形进行建模完成之后,根据式(2)对山体部分进行三维环境建模。

$$Z_i = \sum_{i=1}^k H_i \times \exp\left(-\left(\frac{x-x_i}{x_{si}}\right)^2 - \left(\frac{y-y_i}{y_{si}}\right)^2\right) \quad (2)$$

式中: Z_i 为山峰体的高度, H_i 是地形参数用来决定生成山体的高度, (X_i, Y_i) 表示第 i 个山体对应的中心坐标位置, X_{si} 和 Y_{si} 表示第 i 个山体向 x 轴和 y 轴方向的衰减量, n 为生成山体的总个数。

鉴于无人机常在复杂的无人区环境中飞行,这些区域常伴有大量架设的高压电线杆塔。为更准确模拟山地环境中的电线杆塔特征,通过设置圆柱形障碍物对其形状进行建模,以提高无人机路径规划算法在复杂环境中的适用性和可靠性,其数学模型如式(3)所示。

$$(x-x_0)^2 + (y-y_0)^2 \leq r^2, 0 \leq z \leq h \quad (3)$$

式中: (x_0, y_0) 表示每个圆柱体障碍物的中心点,半径为 r ,高度为 h 。 (x, y) 为地形上任意点的坐标, z 为高度。

三维环境的仿真示意图如图 1 所示。

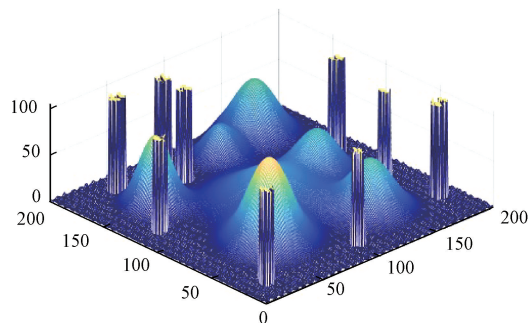


图 1 三维仿真环境示意图

Fig. 1 3D simulation environment schematic diagram

1.2 成本模型

本文基于无人机的路径长度成本、飞行高度成本与飞行转角成本构建综合目标函数,对综合目标函数的寻优过程就是寻找无人机最优路径。

1) 路径长度成本

路径长度用于衡量无人机的飞行距离,通常定义为飞行从起始点到目标点所需要的距离,用路径上相邻点之间的欧几里得距离的累加和表示,表达式如式(4)所示。

$$PathLength = \sum_{i=1}^{n-1} \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2 + (z_{i+1}-z_i)^2} \quad (4)$$

式中: (x_i, y_i, z_i) 表示给定路径上的点。

2) 飞行高度成本

飞行高度对于无人机的飞行过程至关重要,较低的高度可以提升对地面细节的检测精度但同时增加碰撞风险,较高的高度则会节省能耗但同时检测精度也会降低。飞行

高度成本表示为飞行路径上每个点与路径平均高度的偏差累加和,表达式如式(5)所示。

$$Height = \sum_{i=1}^n |z_i - \bar{z}| \quad (5)$$

式中: n 为路径上点数目, z_i 是路径上第 i 个点的高度, \bar{z} 表示路径的平均高度。

3) 飞行转角成本

无人机的转角成本表示为路径上相邻两个线段之间的夹角余弦与理想直线路径的夹角之差的累计和,具体如式(6)所示。

$$Curve = \sum_{i=1}^{n-1} (\cos(\phi) - C_i) \quad (6)$$

式中: C_i 表示路径上相邻线段之间夹角的余弦值, ϕ 为理想直线路径与水平面的夹角。

4) 总成本函数

通过考虑上述各项与路径相关的成本约束,可定义总成本函数如式(7)所示。

$$Fitness = w_1 \times PathLength + w_2 \times Height + w_3 \times Curve \quad (7)$$

式中: $PathLength$ 、 $Height$ 和 $Curve$ 分别表示飞行的路径长度成本、高度成本和转交成本。 w_i 为权重系数,将总成本函数作为无人机路径规划的适应度函数。

5) 航迹平滑

急剧的变化和转弯通常对飞行控制系统的影响很大,为了提高飞行的安全性和可控性需要对飞行路径进行平滑处理,故采用三次样条曲线法进行处理。三次样条函数的形式如式(8)所示。

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad (8)$$

2 多策略改进蜣螂优化算法

2.1 引入 Bernoulli-Tent 混合混沌映射的种群初始化

初始化策略的选择决定了初始种群在解空间中的分布。在原始 DBO 算法^[12]中,种群使用随机生成方法进行初始化,尽管这种方法简单易行,但是随机生成的分布具有不确定性,会导致算法的性能受到影响。为了提高蜣螂种群的多样性,避免传统随机初始化而导致的种群质量参差

不齐,算法在初始化阶段引入混沌映射方法来优化蜣螂个体的初始位置生成。大多数应用都只局限于引入简单的混沌序列,但是由于不同的混沌序列存在不同的特性,Bernoulli 映射可以提供强大的全局搜索能力,但仍然缺乏足够的精细调整能力,而 Tent 映射则可以提供更为精细的局部探索能力,弥补了 Bernoulli 的不足。因此,在 MGDBO 初始化阶段引入融合 Bernoulli 序列和 Tent 序列的复合混沌序列 Bernoulli-Tent。首先,使用均匀分布在区间 $[0,1]$ 内随机生成初始点 x_0 ,然后引入混沌映射切换规则,即通过生成随机数 $r \in [0,1]$,以参数 a 作为阈值控制两种映射的切换,以确保搜索的平衡性。其映射如式(9)所示。

$$x_{n+1} = \begin{cases} \frac{x_n}{1-\lambda}, & 0 \leq x_n < 1-\lambda, r > a \\ \frac{x_n - (1-\lambda)}{\lambda}, & 1-\lambda \leq x_n \leq 1, r > a \\ 2x_n, & 0 \leq x_n < 0.5, r \leq a \\ 2(1-x_n), & 0.5 \leq x_n \leq 1, r \leq a \end{cases} \quad (9)$$

该混沌序列融合了 Bernoulli 序列的快速混沌分布特性和 Tent 序列较为平滑的迭代动态行为。在混合模型中,Bernoulli 映射和 Tent 映射的分界点均设为 0.5,即将 Bernoulli 映射中的 λ 设为 0.5,因为它们都属于典型的双区间映射,使用统一的分界点有助于保证两种映射在区间 $[0,1]$ 内的一致性和连续性。参数 a 取值于 $[0,1]$,其大小决定了 Bernoulli 映射与 Tent 映射的使用频率。常规经验表明,当 a 取 0.5 时,二者的搜索强度和局部搜索能力最为均衡。每次初始化随机生成一个随机数 r ,控制两种映射(Bernoulli 映射和 Tent 映射)之间的切换,提高种群的多样性。图 2 为 10 000 次混沌值分布统计,可以看出相较于单一的 Bernoulli 和 Tent 序列,Bernoulli-Tent 序列产生的随机数分布更加均匀。

引入 Bernoulli-Tent 混合混沌映射的种群初始化策略,其作用机理体现在通过动态切换全局和局部搜索的特性,改善种群的分布结构,提升种群多样性。Bernoulli 映射以快速跳跃特性增强了全局探索能力,Tent 映射则利用平滑渐进的特性提高了算法的局部搜索精度,二者的交替切换实现了算法的搜索平衡。

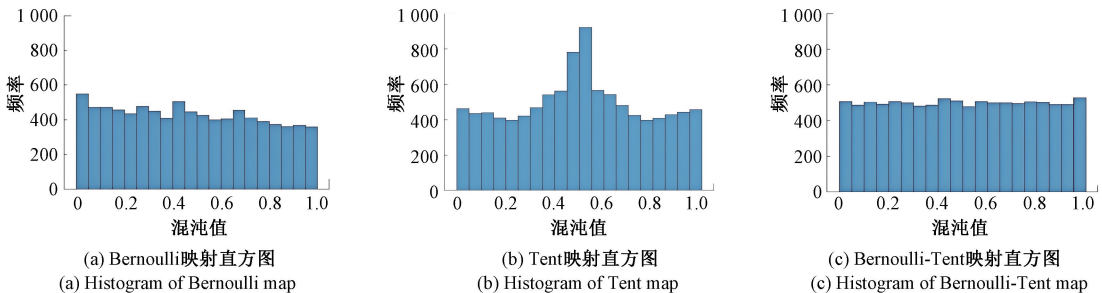


图 2 Bernoulli、Tent 和 Bernoulli-Tent 混沌值分布统计图

Fig. 2 Bernoulli, Tent, and Bernoulli-Tent chaotic value distribution charts

2.2 引入多策略引导机制

潘劲成等^[13]在 2023 年提出了一种的新的改进正弦机制,该机制利用改进的正弦函数进行迭代寻优,通过引入自适应可变惯性权重系数 w_i ,增强算法的全局搜索性能。实验证明,该改进能有效提高原正弦算法的全局搜索能力及收敛精度,其更新公式如式(10)所示。

$$x_i(t+1) = \omega_i x_i(t) + r_1 \times \sin r_2 \times (r_3 \times p_i(t) - x_i(t)) \quad (10)$$

在此基础上,为了简化计算进一步对正弦函数进行优化,具体表现为用线性函数 r_2 代替正弦函数 $\sin r_2$ 。相比于传统的正弦函数计算涉及到复杂的函数运算和周期性角度变化,这在大规模、多维度的优化问题中会增加额外的计算负担,而线性函数不仅可以简化计算过程降低计算复杂度,同时还能保留原有算法中正弦函数所带来的随机性和多样性。经过改进的正弦算法的更新公式如式(11)所示。

$$x_i(t+1) = \omega_i x_i(t) + r_1 \times r_2 \times (r_3 \times p_i(t) - x_i(t)) \quad (11)$$

此外,蛭螂算法在滚球阶段主要依赖于当前解与最差解的差值 Δx 来更新个体位置,这种策略虽然有助于引导算法远离质量较差的解,但也容易导致算法在复杂搜索空间中局部搜索的能力降低。为了弥补这一点,受差分进化算法思想的启发,在蛭螂算法的滚球阶段引入“差分变异”算子,从而增加额外的扰动,使得算法能够扩大搜索空间。差分变异策略的核心思想是随机选取两个个体,并利用它们之间的位置差值来生成新的解,从而增加算法的随机性,帮助算法跳出局部最优。在差分进化算法中,随机个体之间的差值表示如式(12)所示。

$$\Delta x_1 = X_{r_1}(t) - X_{r_2}(t) \quad (12)$$

式中: $X_{r_1}(t)$ 与 $X_{r_2}(t)$ 分别是当前种群中随机选取的两个不同个体的位置。

基于这一思想,将在蛭螂算法的滚球阶段增加差分变异项。在原有的更新公式基础上,改进后的滚球蛭螂的位置更新公式如式(13)所示。

$$\begin{cases} x_i(t+1) = x_i(t) + m \times k \times x_i(t-1) + n \times \Delta x + u \times \Delta x_1 \\ \Delta x = x_i(t) - x^w \\ \Delta x_1 = x_{r_1}(t) - x_{r_2}(t) \end{cases} \quad (13)$$

式中: $u \in (0,1)$ 为一个权重系数,用于调节 Δx_1 对于位置更新的影响。

为了平衡蛭螂算法在探索和开发阶段表现,将引入“差分变异”算子的滚球蛭螂更新公式与经线性函数改进后的正弦算法相结合。在滚球阶段,对蛭螂个体的位置进行更新,改进后的更新公式如式(14)所示。

$$\begin{cases} x_i(t+1) = x_i(t) + m \times k \times x_i(t-1) + n \times \Delta x + u \times \Delta x_1, \varepsilon < P \\ \omega_i x_i(t) + r_1 \times r_2 \times (r_3 \times p_i(t) - x_i(t)), \varepsilon \geq P \end{cases} \quad (14)$$

式中: ε 取 $0 \sim 1$ 的随机数,用于控制蛭螂在位置更新时对策略的选择。 $P \in (0.5, 1]$ 为一个阈值,定义了两种不同策略的概率边界。 P 较小时,系统更偏向全局搜索, P 越大时,系统越偏向局部搜索。经多次实验验证, P 取 0.7 时效果最佳,此时算法的局部搜索与全局搜索的能力达到平衡。

引入线性函数替代正弦函数和差分变异算子的引导机制,从平衡全局与局部搜索能力的角度出发提升算法性能。其改进机理体现在,使用线性函数替代正弦函数,通过简化运算降低计算复杂度,同时保留正弦函数的随机性和多样性,可以显著提升算法的执行效率;差分变异算子的引入通过随机个体位置差值生成新解,增加了种群的扰动性和随机性,有效扩展了搜索空间,增强了算法跳出局部最优解的能力;结合两种策略后的混合更新公式利用概率参数 P 灵活调节全局搜索与局部开发的权衡,确保优化过程中的动态适应性,从而在提高算法收敛速度的同时,使算法保持较高的寻优精度和鲁棒性。

2.3 改进的螺旋搜索策略

在蛭螂算法中,如果种群在产卵区域内直接繁殖雏球,会导致算法快速收敛,种群多样性降低,容易陷入局部最优。蛭螂在构建孵化点时,通常通过反复调整位置和方向,类似于螺旋上升运动逐渐靠近最优产卵点,这个过程类似鲸鱼优化算法(whale optimization algorithm, WOA)^[14]中头鲸通过螺旋搜索策略围捕猎物的行为。WOA 的螺旋搜索不仅加快了收敛速度,还保持了种群多样性,其数学模型如式(15)所示。

$$x(t+1) = |x_{best}(t) - x(t)| \times e^{cl} \times \cos(2\pi l) + x_{best}(t) \quad (15)$$

式中: c 为常数,用于定义对数螺线的形状; l 为 $[-1, 1]$ 上的随机数。

针对这一情况,可以利用螺旋搜索策略对蛭螂算法繁殖阶段的繁育球位置更新公式进行改进。改进后的繁育球位置更新公式如式(16)所示。

$$\begin{aligned} B_i(t+1) = & x^* + e^{cl} \times \cos(2\pi l) \times (B_i(t) - Lb^*) + \\ & e^{cl} \times \cos(2\pi l) \times b_2 \times (B_i(t) - Ub^*) \end{aligned} \quad (16)$$

由于 e^{cl} 指数函数计算复杂度高,为了来简化计算可以用 $(cl)^2 + 1$ 来代替。首先,指数函数涉及到高昂的计算成本,相比之下 $(cl)^2 + 1$ 是一个简单的二次函数,其计算复杂度远低于指数函数;其次,虽然指数函数与二次函数的增长速率不完全相同,但二次函数仍然保持了指数函数非线性增长的特性,二次函数的增长速度随着 l 的增大而加快,类似于指数增长的趋势,这种非线性的增长方式在一定程度上可以模拟出螺旋上升过程中对距离的调整。

同理,更新公式中的 $\cos(2\pi l)$ 为三角函数,引入了周期性震荡增加了计算复杂度。为了简化计算,可以将其替换为取值范围在 $[-1, 1]$ 的随机数 o 。这样既保证了值域同样属于 $[-1, 1]$,也保持了必要的随机性和多样性。故

此时,繁育球的位置更新公式改进如式(17)所示。

$$B_i(t+1) = x^* + ((cl)^2 + 1) \times o \times (B_i(t) - Lb^*) + ((cl)^2 + 1) \times o \times b_2 \times (B_i(t) - Ub^*) \tag{17}$$

改进的螺旋搜索策略通过模拟蜣螂在繁殖阶段的螺旋上升运动行为,优化繁育球位置更新机制,可以显著提升算法的搜索性能和计算效率。首先,该策略有效避免了直接在产卵区域生成繁育球所导致的快速收敛问题,避免陷入局部最优;其次,通过引入螺旋路径搜索,提升了算法的全局搜索能力,进一步提高算法的寻优性能。此外,使用二次函数替代指数函数模拟螺旋增长趋势,用随机数代替三角函数表达周期性震荡,简化了计算复杂度。

繁育球的搜索路径的示意图如图 3 所示。

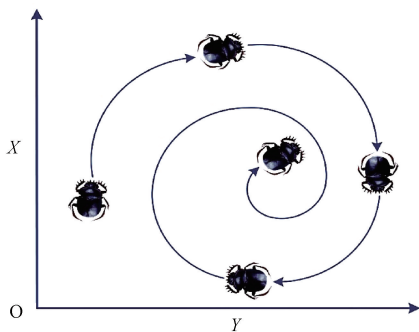


图 3 蜣螂搜索路径示意图

Fig. 3 Dung beetle search path schematic diagram

2.4 MGDBO 的算法步骤

步骤 1)初始化算法参数,包括种群的规模 N 、最大迭代次数 T 、滚球蜣螂比例、觅食蜣螂比例,搜索间的上下界 Lb 、 Ub 和目标函数的维度 Dim 等。

步骤 2)使用 Bernoulli-Tent 混合混沌映射初始化种群,并记录最佳位置 $x_{best}(t)$ 。

步骤 3)按照比例选取一部分蜣螂作为滚球蜣螂,按照式(14)更新位置。

步骤 4)重新计算蜣螂种群的个体适应度值及全局最优解 $x_{best}(t)$,同时更新边界,去除超过边界的个体。

步骤 5)根据式(17)更新蜣螂繁育阶段的繁育球位置,并逐步完成蜣螂的觅食和偷窃行为位置更新。

步骤 6)判断达到最多迭代次数 T 。若是,转步骤 7);否则,转步骤 2)。

步骤 7)输出最优蜣螂的位置及全局最优解。

3 仿真实验与结果分析

3.1 标准测试实验

仿真实验的操作系统均为 macOS 10.13.6,CPU 为 1.8 GHz Intel Core i5,内存为 8 GB(1 600 MHz),编程软件为 MATLAB R2018A。为了测试 MGDBO 算法的性能,共选用了 6 个经典标准测试函数进行仿真实验,其中, $F_1 \sim F_3$ 为单峰测试函数, $F_4 \sim F_6$ 为多峰测试函数,测试函数信息如表 1 所示。

表 1 基准测试函数

Table 1 Benchmark test functions

测试函数	维度	范围	最优值
$F_1 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	10	$[-100,100]$	0
$F_2 = \max\{ x_i , 1 \leq i \leq n\}$	10	$[-100,100]$	0
$F_3 = \sum_{i=1}^n (x_i + 0.5)^2$	10	$[-100,100]$	0
$F_4 = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	10	$[-500,500]$	$-418.982\ 9 \times Dim$
$F_5 = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos 2\pi x_i) + 20 + e$	10	$[-32,32]$	0
$F_6 = \frac{\pi}{n} \{10\sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n-1} u(x_i, 10, 100, 4),$ $y_i = 1 + \frac{1}{4}(x_i + 1), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	10	$[-50,50]$	0

所有算法设置最大迭代次数 500,种群大小 50,将传统的蜣螂优化算法(DBO)、遗传算法(genetic algorithm, GA)、白鲸优化算法(beluga whale optimization,

BWO)^[15]、正余弦优化算法(sine cosine algorithm, SCA)^[16]、基于高斯量子改进的粒子群算法(Gaussian quantum particle swarm optimization, GQPSO)、自适应狐

狸优化算法 (adaptive fox optimization algorithm, AFOX) 和改进的蜣螂优化算法 (MGDBO) 独立运行 30 次, 对实验结果进行统计, 对比结果如表 2 和图 4 所示。选择这些算法进行比较是基于其各自的性能特点。GA 为经典算法具有强大的全局搜索能力, 但易于早熟收敛, DBO 通过模拟蜣螂行为改善了种群平衡, 降低局部最优的风险; BWO 的

螺旋搜索路径在局部优化中表现良好, 但种群多样性不足, 而 DBO 的动态策略提升了全局搜索能力; SCA 路径生成简单, 但平衡性较差且计算复杂度较高, DBO 在全局与局部搜索的动态调整方面表现更优且效率更高。同时, GQPSO 和 AFOX 等新算法有助于分析不同搜索策略对收敛速度和精度的影响, 从而全面验证 MGDBO 的性能。

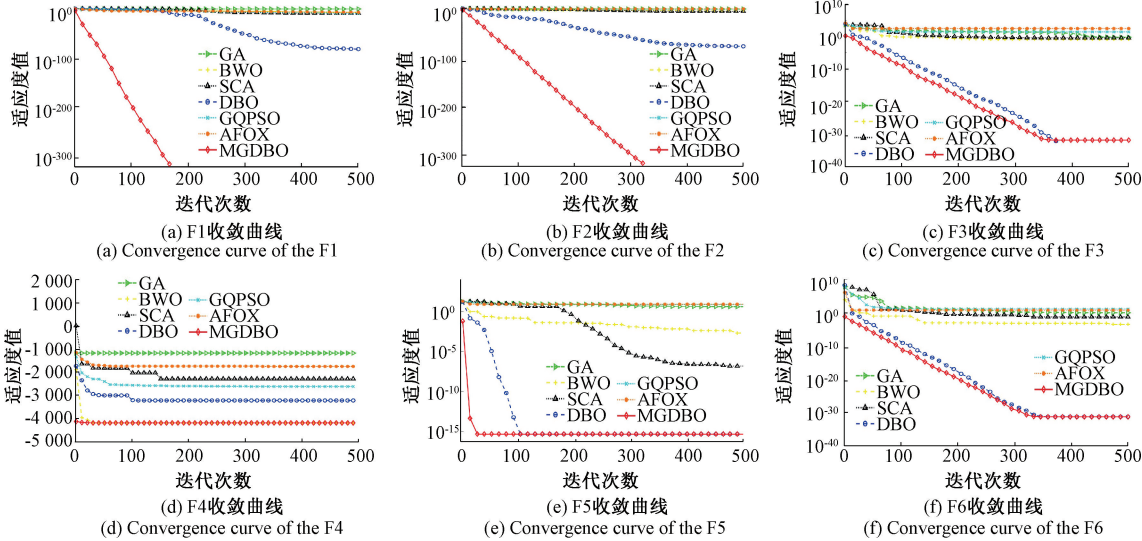


图 4 不同算法测试函数收敛

Fig. 4 Convergence plots of test functions for different algorithms

表 2 7 种算法寻优值

Table 2 Optimal values of the 7 algorithms

函数	指标	GA	BWO	SCA	GQPSO	AFOX	DBO	MGDBO
F1	平均值	2.79×10^3	1.99×10^{-3}	2.24×10^{-4}	9.83×10^1	9.99×10^2	5.06×10^{-93}	0.00
	标准差	1.36×10^3	1.28×10^{-3}	6.46×10^{-4}	2.01×10^2	5.92×10^2	2.77×10^{-92}	0.00
	最优值	5.65×10^2	4.23×10^{-4}	7.95×10^{-11}	9.08×10^{-2}	2.04×10^2	8.40×10^{-153}	0.00
	平均值	1.16×10^1	2.22×10^{-3}	2.19×10^{-4}	1.11×10^1	2.04×10^1	2.32×10^{-53}	0.00
F2	标准差	5.14	9.92×10^{-4}	6.82×10^{-4}	4.00	7.66	1.26×10^{-52}	0.00
	最优值	5.14	5.14	1.97×10^{-7}	1.97×10^{-7}	9.66	1.15×10^{-84}	0.00
	平均值	2.34	1.06×10^{-1}	3.63×10^{-1}	5.59×10^1	5.76×10^2	2.49×10^{-32}	4.01×10^{-33}
F3	标准差	4.05	6.70×10^{-2}	1.54×10^{-1}	8.87×10^1	3.19×10^2	8.01×10^{-32}	6.97×10^{-33}
	最优值	1.37×10^{-1}	1.21×10^{-3}	4.78×10^{-2}	1.19×10^{-1}	1.55×10^2	0.00	0.00
	平均值	-1.33×10^3	-4.18×10^3	-2.21×10^3	-2.43×10^3	-2.26×10^3	-3.49×10^3	-4.18×10^3
F4	标准差	2.96×10^2	2.23	1.51×10^2	2.42×10^2	-2.87×10^2	3.33×10^2	2.36×10^{-12}
	最优值	-2.04×10^3	-4.18×10^3	-2.56×10^3	-2.95×10^3	-2.84×10^3	-4.18×10^3	-4.18×10^3
	平均值	8.62	2.55×10^{-3}	8.04×10^{-8}	9.34	8.80	8.88×10^{-16}	8.88×10^{-16}
F5	标准差	5.05	1.27×10^{-3}	3.65×10^{-7}	4.92	1.80	0.00	0.00
	最优值	7.87×10^{-1}	3.08×10^{-4}	4.39×10^{-12}	2.62	4.68	8.88×10^{-16}	8.88×10^{-16}
	平均值	7.08×10^{-1}	3.86×10^{-4}	8.20×10^{-2}	7.31	3.04×10^3	6.45×10^{-18}	4.83×10^{-32}
F6	标准差	1.07	4.16×10^{-4}	3.16×10^{-2}	4.83	1.42×10^4	3.52×10^{-17}	1.05×10^{-33}
	最优值	3.96×10^{-4}	3.26×10^{-5}	4.20×10^{-2}	1.26	1.87	4.71×10^{-32}	4.71×10^{-32}

结合表 2 中结果和图 4 来看, 在 6 个经典测试函数实验中, MGDBO 算法相比 GA、BWO、SCA、GQPSO、AFOX 以及原 DBO 算法, 在单峰与多峰函数中的最优值、平均

值均表现最佳, 且标准差更小, 展现出更好的鲁棒性。在理论最优值不为零的 F4 函数中, MGDBO 算法展现出更高的收敛精度; 对于多峰函数 F5, GA、BWO、SCA 算法均

在不同程度上陷入局部最优,导致寻优精度降低, MGDBO 表现优异展现出其优秀的全局探索能力。从图 4 七种算法的收敛曲线图, MGDBO 算法在迭代初期就展现出较强的全局探索能力;随着迭代的进行, MGDBO 算法更是比其他算法有更高的收敛速度与收敛精度,相较于原 DBO 算法有了明显的提升,证明了其改进策略的有效性。

3.2 Wilcoxon 秩和检验

为了进一步验证改进算法的有效性,采用 Wilcoxon 秩

和检验来测试 MGDBO 的寻优性能与 GA、BWO、SCA、GQPSO、AFOX 及原 DBO 算法是否存在显著差异实验结果如表 3 所示。Wilcoxon 秩和检验是一种非参数统计检验方法,当 $p < 0.05$ 时说明两种算法之间存在显著性差异,否则差异不明显。

由表 3 结果可知,除测试函数 F3 中与原 DBO 算法对比的 $p > 0.05$ 外,其余 p 值均 < 0.05 ,表明了 MGDBO 与 5 种对比算法在 6 个基准测试函数上几乎都表现出了显著性差异,进一步证明了改进算法的有效性。

表 3 Wilcoxon 秩和检验
Table 3 Wilcoxon rank-sum test

函数	GA	BWO	SCA	GQPSO	AFOX	DBO
F1	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
F2	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
F3	2.33×10^{-11}	2.33×10^{-11}	2.33×10^{-11}	1.99×10^{-11}	1.99×10^{-11}	1.24×10^{-1}
F3	1.41×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	6.47×10^{-11}
F5	2.08×10^{-11}	1.21×10^{-11}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	3.37×10^{-1}
F6	2.08×10^{-11}	2.08×10^{-11}	2.08×10^{-11}	2.35×10^{-11}	2.35×10^{-11}	1.58×10^{-2}

3.3 搜索代理历史轨迹

尽管上述实验结果验证了 MGDBO 算法的高性能,但仍需监控其搜索代理在优化期间的行为,以观察它们如何在搜索空间中移动。为此,设置了 50 个搜索代理,最大迭代次数为 100 次,结果如图 5 所示。观察到其在 F1、F3 及

F5 测试函数上的搜索轨迹, MGDBO 算法在搜索空间内有希望的区域进行搜索,全局最优值周围的采样点的分布密集,这表明了该算法在探索的同时还利用了搜索空间中最有希望的区域。

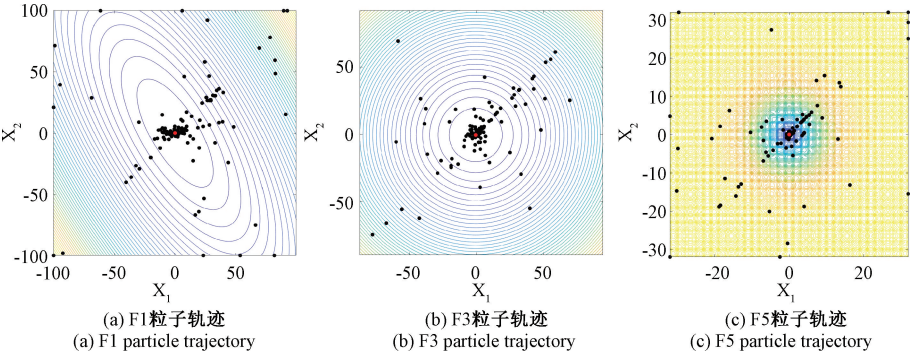


图 5 搜索代理的历史搜索记录图
Fig. 5 Search history plot of the search agent

4 三维路径规划仿真测试

4.1 多种算法对比实验

为验证 MGDBO 算法无人机路径规划规划的可行性及优越性,对此进行仿真实验,并与樽海鞘群算法 (salp swarm algorithm, SSA)、BWO 及原 DBO 算法路径规划结果进行比较,对比算法的参数如表 4 所示。各算法种群数均设为 70、迭代次数为 50,同时进行实验 30 次,获得每个算法计算的最优路径长度的对比数据如表 5 所示,最佳的数据结果用加粗显示。

表 4 各算法参数设置

算法	参数设置
SSA	$k = 0.2, R_2 = 0.8, SD = 0.1, \beta = 2, ST = 0.5$
BWO	$\beta = 1.5$
DBO	$k = 0.2, a = 1$ 或 $a = -1, b = 0.3, S = 0.5, P = 0.7, c = 1$
MGDBO	$k = 0.2, a = 1$ 或 $a = -1, b = 0.3, S = 0.5, P = 0.7, c = 1, \omega_t = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \frac{t}{T_{\max}}, \omega_{\max} = 0.9, \omega_{\min} = 0.782$

表 5 各个算法的运行结果对比

Table 5 Comparison of the results from different algorithms

算法	最优值	平均值	最差值	标准差
SSA	284.57	311.73	377.6	29.79
BWO	305.33	378.76	435.64	55.63
DBO	284.58	313.56	391.01	34.38
MGDBO	283.41	291.18	320.49	9.21

从表 5 可以看出, MGDBO 在最优适应度、平均适应度和标准差上均优于其他 3 种对比算法。与原 DBO 相比, MGDBO 在寻优能力以及平衡探索和开发方面表现更佳。BWO 与 MGDBO 差距较大, 在寻优能力和鲁棒性上均逊色于 MGDBO。虽然 SSA 与 MGDBO 的寻优能力接近, 但 MGDBO 的稳定性更强。总体而言, MGDBO 算法展现了较强的寻优能力和较高的算法稳定性。

图 6 为多个算法在三维地形图上的飞行轨迹图以及俯视角, 共设置山体障碍物 7 个和 5 个柱形区域障碍模拟无人机飞行环境, 图 7 则显示了各算法的成本收敛曲线。

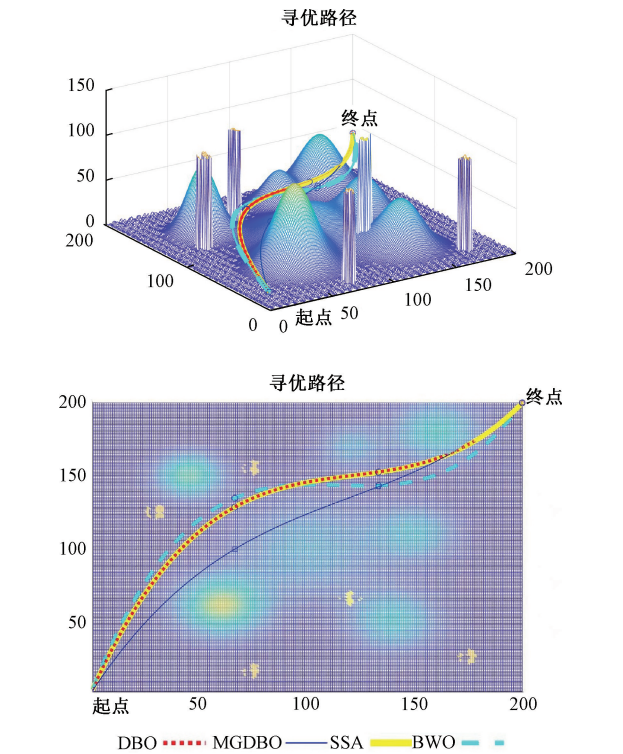


图 6 各算法在相同地形中的飞行路径

Fig. 6 Flight paths of different algorithms in the same terrain

结合图 6 和图 7 可知, MGDBO 算法在相同飞行环境中以最快速度找到最优路径, 展现出极高的收敛性能及平衡探索和开发的能力。BWO 算法的路径呈现很明显 S 型, 产生不必要弯曲, 说明其易陷入局部最优。DBO 和

SSA 的收敛能力和速度仅次于 MGDBO, 但在前期陷入局部最优, 直到 35 次迭代后才继续收敛, 导致收敛效果不理想。综上, 改进的 MGDBO 算法在无人机路径规划上效率更高, 寻优能力更强, 有较好的实用价值。

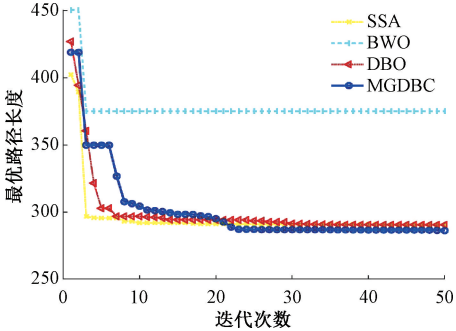


图 7 各个算法的成本收敛曲线图

Fig. 7 Cost convergence curves of different algorithms

4.2 不同场景下的路径规划实验

为验证 MGDBO 算法在不同场景下的适应性及规划能力, 仿照文献[17]共设置了 4 种不同的飞行环境, 可以更全面的验证算法的性能。图 8~11 展示了 MGDBO 算法在不同场景中的飞行路径图。

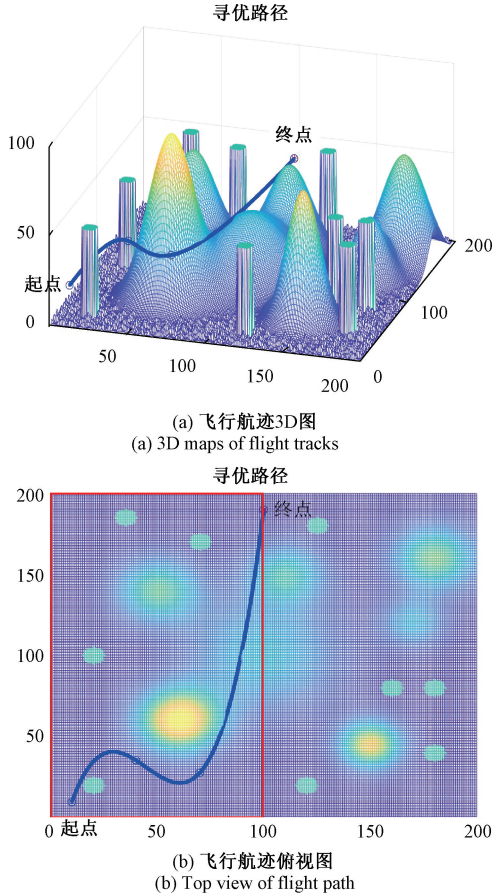


图 8 场景 1 中无人机飞行路径

Fig. 8 UAV flight path in scenario 1

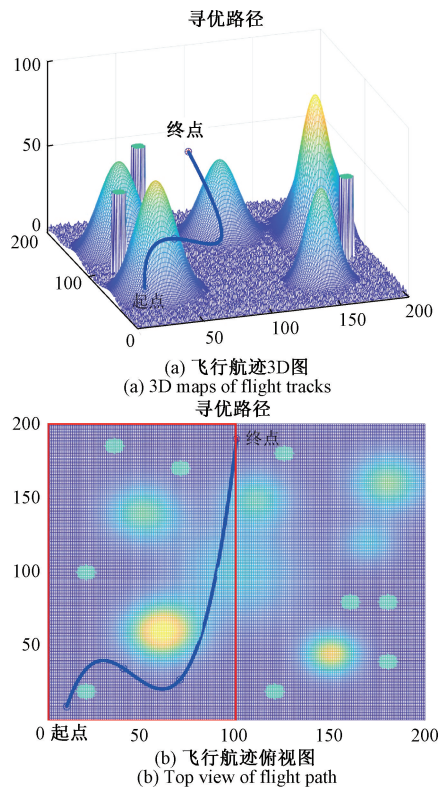


图 9 场景 2 中无人机飞行路径
Fig. 9 UAV flight path in scenario 2

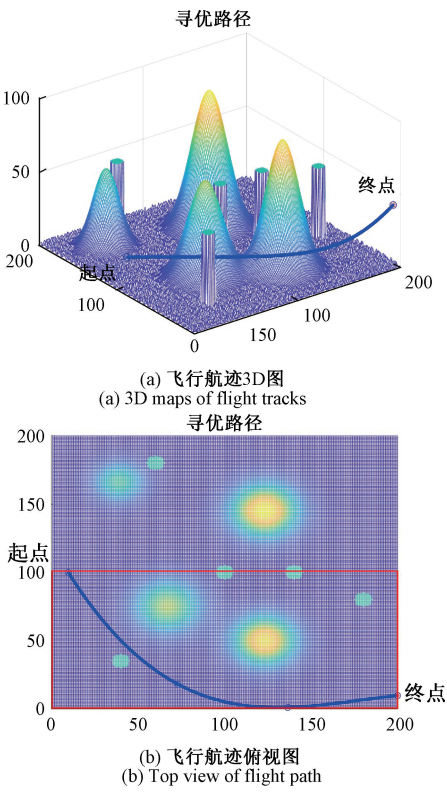


图 10 场景 3 中无人机飞行路径
Fig. 10 UAV flight path in scenario 3

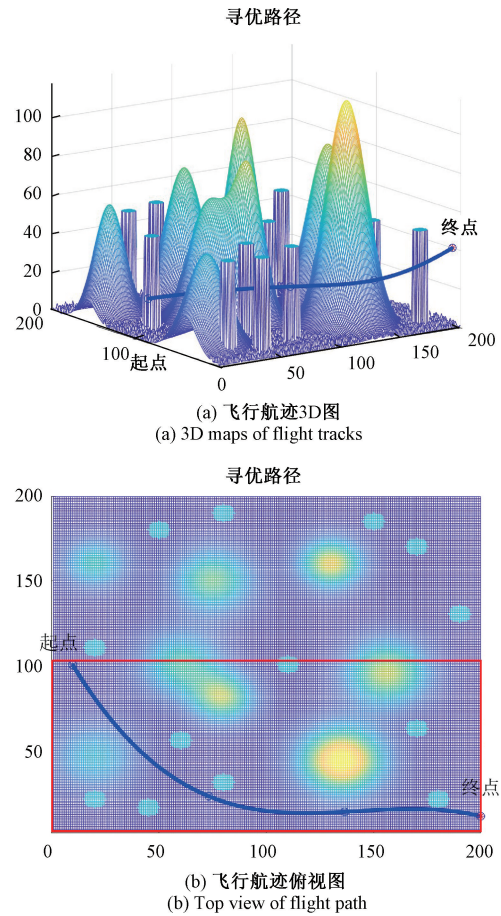


图 11 场景 4 中无人机飞行路径
Fig. 11 UAV flight path in scenario 4

表 6 展示了其在不同的山体数量、障碍物数量、飞行区域下无人机的飞行路径长度。无人机的飞行区域大小一致,但场景 1 和场景 2 的飞行区域为地图的左半部分,场景 3 和场景 4 的飞行区域为地图的下半部分。

表 6 各场景障碍物与飞行区域设置				
Table 6 Obstacle and flight area settings for each scenario				
场景	山体数量	障碍物数量	飞行区域	飞行距离
场景 1	7	9	100×200 (左半区域)	332.51
场景 2	5	3	100×200 (左半区域)	334.45
场景 3	4	5	200×100 (下半区域)	263.42
场景 4	8	11	200×100 (下半区域)	261.74

图 8 为多高压杆塔场景,包含 7 座山体和 9 个障碍物,MGDBO 成功避开障碍,规划出平滑且较短的路径;图 9 中山体分散,杆塔数量较少,共有 5 座山体和 3 个障碍物,

MGDBO 同样有效避障并生成平滑路径;图 10 中地形错落有致,包含 4 座山体和 5 个障碍物,最接近无人机飞行的真实环境,MGDBO 能够顺利绕过障碍完成路径规划;图 11 展示了最复杂的地形,有 8 座山体和 11 个障碍物,MGDBO 依然能够快速找到平滑且短的路径。这些结果表明 MGDBO 在具有不同障碍物和飞行区域的复杂场景下都具备出色的全局路径规划能力。

5 结 论

针对传统螭螂优化算法在多障碍复杂环境下进行无人机路径规划时的局限性,如易陷入局部最优、全局搜索能力不足等问题,提出了改进的 MGDBO 算法。该算法引入 Bernoulli-Tent 混合混沌映射、多策略引导机制和改进的螺旋搜索策略。通过与原始 DBO 算法以及其他 5 个智能优化算法在测试函数上的对比,验证了 MGDBO 算法在寻优精度和稳定性上都具有了进一步的提升,又在不同场景下的无人机三维航迹规划实验中与另外 3 个智能优化算法进行对比,证实了改进算法能够更稳定地生成更安全避障且总代价更低的飞行路径。

参考文献

- [1] 李奇洋,涂海燕,叶鸿达.考虑地面行人安全的无人机低风险路径规划策略[J].电子测量技术,2023,46(20):58-64.
LI Q Y, TU H Y, YE H D. Low-risk path planning strategy to consider ground pedestrian for unmanned aerial vehicle[J]. Journal of Electronic Measurement and Instrumentation, 2023, 46(20): 58-64.
- [2] 苏雨蕾,黄丹丹,刘智,等.无人机对地小目标检测方法研究[J].电子测量与仪器学报,2024,38(9):144-154.
SU Y L, HUANG D D, LIU ZH, et al. Research on detection methods of small targets on ground by UVA[J]. Journal of Electronic Measurement and Instrumentation, 2024, 38(9): 144-154.
- [3] 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.
- [4] 王洪斌,尹鹏衡,郑维,等.基于改进的 A* 算法与动态窗口法的移动机器人路径规划[J].机器人,2020,42(3):346-353.
WANG H B, YIN P H, ZHENG W, et al. Mobile robot path planning based on improved A* algorithm and dynamic window method[J]. Robot, 2020, 42(3): 346-353.
- [5] 余敏,罗建军,王明明,等.一种改进 RRT* 结合四次样条的协调路径规划方法[J].力学学报,2020,52(4):1024-1034.
YU M, LUO J J, WANG M M, et al. Coordinated path planning by integrating improved RRT* and quartic spline[J]. Chinese Journal of Theoretical and Applied Mechanics, 2020, 52(4): 1024-1034.
- [6] 陈明强,冯树娟,李奇峰.基于改进人工势场的物流无人机三维航迹规划[J].无线电工程,2023,53(10):2352-2359.
CHEN M Q, FENG SH J, LI Q F. Three-dimensional trajectory planning of logistics UAV based on improved artificial potential field [J]. Radio Engineering, 2023, 53(10): 2352-2359.
- [7] 姜辰凯,李智,盘书宝,等.基于改进 Dijkstra 算法的 AGVs 无碰撞路径规划[J].计算机科学,2020,47(8):272-277.
JIANG CH K, LI ZH, PAN SH B, et al. Collision-free path planning of AGVs based on improved dijkstra algorithm [J]. Computer Science, 2020, 47 (8): 272-277.
- [8] 吴学礼,王超,赵俊棋,等.改进麻雀算法的无人机三维路径规划[J].科学技术与工程,2024,24(15):6534-6542.
WU X L, WANG CH, ZHAO J Q, et al. Improved sparrow algorithm for UAV 3D path planning [J]. Science Technology and Engineering, 2024, 24 (15): 6534-6542.
- [9] SONG B Y, WANG Z D, ZOU L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve[J]. Electronics Optics&Control, 2021, 100:106960.
- [10] 张鹭,毛海亮,卞鹏,等.自适应蚁群算法的无人机三维航迹规划[J].电光与控制,2024,31(5):24-29.
ZHANG AO, MAO H L, BIAN P, et al. 3D UVA flight path planning with adaptive ant colony optimization[J]. Electronics Optics&Control, 2024, 31(5): 24-29.
- [11] 赵娟.启发点引导 D* 算法扩展的无人机航迹规划策略[J].机械设计与制造,2020(2):153-157.
ZHAO J. Unmanned air vehicle route planning strategy based on D* algorithm extension guided by inspiration point [J]. Machine Design and Manufacturing, 2020(2): 153-157.
- [12] XUE J K, SHEN B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization[J]. The Journal of Supercomputing, 2022, 79 (7): 7305-7336.
- [13] 潘劲成,李少波,周鹏,等.改进正弦算法引导的螭螂优

化算法[J]. 计算机工程与应用,2023,59(22):92-110.

PAN J CH, LI SH B, ZHOU P, et al. Improved sinealgorithm-guided dung beetle optimization algorithm [J]. Computer Engineering and Applications,2023,59(22):92-110.

[14] MIRJALILI S, LEWIS A. The whale optimizationalgorithm[J]. Advances in engineering software, 2016, 95: 51-67.

[15] ZHONG CH T, LI G, MENG Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm [J]. Knowledge-Based Systems, 2022, 251: 109215.

[16] MIRJALILI S. SCA: A sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016,96: 120-133.

[17] 刘文强,李涛. 基于改进蜣螂优化算法的无人机航迹规

划[J]. 电子测量技术,2024,47(15):64-72.

LIU W Q, LI T. UAV trajectory planning based on enhanced dung beetle optimization algorithm [J]. Journal of Electronic Measurement and Instrumentation, 2024,47(15):64-72.

作者简介

曲良东(通信作者),副教授,博士,硕士生导师,主要研究方向为智能计算及应用、数据挖掘与知识发现、控制理论与应用。

E-mail:quliangdong@126.com

梅雨琳,硕士研究生,主要研究方向为算法设计与分析、智能计算及应用。

E-mail:243827076@qq.com

饶爽,硕士研究生,主要研究方向为算法设计与分析、智能计算及应用。

E-mail:1064882740@qq.com