

预警机制的鲸鱼算法及时间约束的航迹规划

闫浩泉¹, 杨柳庆^{2,3}, 张勇^{2,3}

(1. 南京航空航天大学 自动化学院, 南京 211106;

2. 南京航空航天大学 无人机研究院, 南京 211106;

3. 南京航空航天大学 中小型无人机先进技术工业和信息化部重点实验室, 南京 211106)

摘要: 针对高实时性要求的固定翼无人机航迹规划问题, 在模型中引入算法的运行时间以契合实际工程要求, 并提出基于预警机制的改进鲸鱼优化算法完成求解; 该算法通过对适应度进行排序定义个体的预警概率, 并借此控制个体更新机制的选取; 随后引入与预警概率关联的权重系数控制螺旋更新机制的收缩扩张, 同时使用莱维飞行改进随机游走机制加快收敛, 达到平衡各机制开发与探索能力的目的; 有效地改善算法收敛速度慢、精度低的缺陷; 使用基准函数测试并验证算法的有效性, 并在不同维度与距离的航迹规划对比仿真实验中量化改进算法在收敛精度与收敛速度的优越性; 仿真实验表明, 面对低维度航迹规划时, 算法精度可提高 8.0%; 面对高纬度航迹规划时, 算法收敛速度可提高 50%;

关键词: 鲸鱼优化算法; 航迹规划; 时间约束; 莱维飞行; 可变维度

Whale Optimization Algorithm Based on Early Warning Mechanism and Path Planning with Time Constraint

YAN Haoquan¹, YANG Liuqing^{2,3}, ZHANG Yong^{2,3}

(1. Research Institute of Pilotless Aircraft, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China;

2. Key Laboratory of Unmanned Aerial Vehicle Technology, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China;

3. College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: For the fixed wing UAV path planning with high real-time requirements, time cost is introduced into the optimization model to fit the actual project, and thenew improved whale optimization algorithm with early warning mechanism is proposed to complete the optimization solution. The algorithm establishes individual early warning probability by ranking fitness of the population, and uses itto control the selection of update mechanism. By introducing the weight coefficient associated with the early warning probability to control the expansion and contraction of the spiral update mechanism, and using Lévy flight to improve the random walk mechanism to accelerate the convergence, the goal of balancing the development and exploration capabilities of each mechanism is achieved, which is beneficial to alleviate the problems of whale optimization algorithm, such as slow convergence speed and low convergence accuracy. The simulation experiment uses benchmark function to prove the effectiveness of the algorithm, and the simulation of path planning in different dimensions and distances shows the superiority of the improved algorithm. The simulation results show that the algorithm accuracy can be improved by 8.0% when dealing with low dimension path planningandthe convergence speed of algorithm can be improved by 50%.

Keywords: whale optimization algorithm; path planning; time constraint; lévy flight; variable dimension

0 引言

无人机技术的高速发展使其在石油、军事、农业、气象、地理测绘等领域展现广泛的应用潜力。不同于四旋翼

无人机, 军用固定翼无人机的发展越来越重视对控制系统的实时性设计。无论是无人机蜂群技术的小型化发展趋势^[1], 还是高超声速无人机的超高速发展趋势^[2], 短距离与长续航均对航迹规划的求解提出了快速性要求。

收稿日期: 2022-11-16; 修回日期: 2022-12-20。

基金项目: 国家自然科学基金项目(52272369)。

作者简介: 闫浩泉(1998-), 男, 硕士研究生。

杨柳庆(1979-), 男, 硕士, 副研究员, 硕士生导师。

引用格式: 闫浩泉, 杨柳庆, 张勇. 预警机制的鲸鱼算法及时间约束的航迹规划[J]. 计算机测量与控制, 2023, 31(10): 140-146, 173.

无人机航迹规划是典型的多目标优化问题。旨在综合考虑环境威胁、战场威胁、飞行时间、油耗等多约束条件下, 寻找一条从起始点到目标点的最优或可行的飞行轨迹。目前的优化求解算法可大致分为两类, 传统优化算法与智能优化算法。基于图论的 Voronoi 图法^[3]、基于随机抽样的随机搜索树算法 RRT^[4]、动态规划等均属于传统优化算法。此类算法提出时间早, 理论成熟, 但在解决凸优化等日益复杂的优化问题表现乏力。新型智能优化算法包括不限于各类群智能优化算法: 蚁群算法 (ACO, ant colony optimization)^[5]、粒子群算法 (PSO, particle swarm optimization)^[6]、灰狼算法 (GWO, grey wolf optimizer)^[7]、鲸鱼优化算法 (WOA, whale optimization algorithm)^[8] 等。各类机器学习算法: Q 学习算法^[9]、深度确定性策略梯度算法 (DDPG, deep deterministic policy gradient)^[10] 等。多数群智能算法是通过观察自然界中的生物群体并分析对应的群体智能性发展而来的仿生算法。个体的数量决定并行运算的规模, 个体之间的信息交流决定并行运算的效率。算法具备收敛性前提下, 二者合力影响求解的效率。鲸鱼优化算法相比 ACO 算法与 PSO 算法, 参数少, 信息交流机制丰富, 具有、易于实现、高效、普适性等特点。即便对比于新兴的机器学习算法, 也具有无训练时间成本、应用成熟的优势。

WOA 算法^[11]的生物模拟对象是捕食过程中的座头鲸群。已有众多学者通过对算法的自身改进以及算法的融合改进完成性能优化。算法的初始化方式以及随机机制是改进的常选对象, 张永、王坚浩分别采用不同的混沌映射改善初始化方式、Zhang、liu 等通过变异算子^[12]以及莱维飞行扰动改进随机机制^[13], 提高算法的探索能力。通过对自身种群进行分组还可以实现结构层面的改进。文献 [14] 将种群分成搜索组与猎物组, 利用搜索组的最优个体引导猎物组。Kaveh 提出的与碰撞优化算法混合的改进方法具有相似的机制^[15]。刘小龙^[16]将种群划分为多个相互独立的子群, 使用横向和纵向两种对比方式影响个体。

融合改进则是融合其他群智能优化算法原理进行结构、随机机制优化的方法。Korashy 将 GWO 的领导层结构引入 WOA 中, 实现结构分级的优化^[17], Strumberger 等人则将人工蜂群算法的搜索结构代替原有搜索机制^[18], 提高种群丰富度。

针对 WOA 算法高维优化问题收敛速度慢, 多极值问题易于早熟、陷入局部极值的问题, 且为满足航迹规划的实时性要求, 本文提出基于预警机制的鲸鱼优化算法 (AP-WOA, adaptive probabilistic whale optimization algorithm) 用于优化求解。同时将算法解算时间计入航迹代价的优化模型, 提高实时性的优先级。APWOA 算法引入与个体适应度相关的自适应概率控制 3 种更新机制的选取, 对适应度低的个体发出预警, 提高其进行随机更新的概率。从整体层面提高算法个体的差异性。同时自适应概率还参与

WOA 特有的螺旋式更新机制的自适应权重定义, 控制收缩范围实现机制层面的探索与开发的平衡。为验证算法的收敛能力, 使用优化算法的基准测试函数作为工具, 与多种算法进行对比分析。针对航迹规划问题, 在联合坐标等分法基础上, 通过控制步长实现变维度的航迹序列生成, 完成不同规划维度方面的仿真对比。

1 场景与鲸鱼优化算法

1.1 规划场景设计

1.1.1 地形威胁

根据飞行环境和任务要求建立适当的规划空间是解决路径规划的前提。数字高程地图 (DEM, digital elevation map) 为指定投影平面上的空间水平间隔的高程值矩阵, 本质是关于高程平面坐标 (x, y) 的连续函数, 实际使用时作为一个有限的离散集。采用文献 [19] 提及的地形模型函数采集 DEM 所需高程数据, 具体为:

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

其中: (x, y) 为水平面的点坐标, z 为坐标 (x, y) 相对应的高度。 a, b, c, d, e, f, g 为相关系数, 改变相关系数以获得不同地形数据。

第 k 山峰对无人机造成的山峰威胁表示为:

$$T_{\text{terrain}} = \begin{cases} R_i(h) + d_{\min} - d, & d < (R_i(h) + d_{\min}) \\ \text{and} & h < H(k) \\ 0, & d > (R_i(h) + d_{\min}) \\ \text{or} & h > H(k) \end{cases} \quad (2)$$

其中: d 为无人机至山峰 k 对称轴的距离; d_{\min} 为地形最小作用距离。 $H(k)$ 为山峰 k 的高度 R_i 为山峰的最大延伸半径; $R_i(h)$ 的计算公式为:

$$R_i(h) = (H(k) - h) / \tan\theta \quad (3)$$

其中: θ 为该山峰地形的坡度。

式 (2)、式 (3) 中各变量间的关系如图 1 所示。

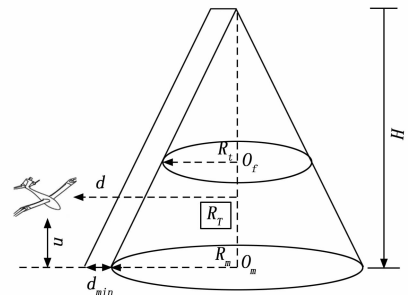


图 1 地形威胁示意图

1.1.2 雷达威胁

雷达通过发射电磁波搜寻并分析空间内物体, 获知物体的方向、高度和速度信息, 进而达到侦查目的。雷达的威胁程度与其信号强度成正比, 后者满足信噪比公式:

$$S/N = \frac{P_r G_r G_t P \lambda^2 \sigma}{(4\pi)^3 K_s L_m B_n T_s d_R^4} \quad (4)$$

S/N 表示雷达接收器的信噪比, P_t 为雷达发射器功率, G_r 为接收天线增益, G_t 为发射天线增益, σ 表示被探测目标的有效雷达散射面积。 λ 为工作波长, K_s 表示波兹曼常数, L_m 为损耗因子, B_n 为滤波器带宽, T_s 为工作时绝对温度, 最后的 d_R 表示无人机与雷达的距离, 若仅考虑 d_R 为变量, 其余参数视为常数, 则信噪比公式可简化为:

$$T_{\text{radar}} = S/N = \frac{K}{d_R^4} \quad (5)$$

不难看出, 雷达威胁关于距离的四次方成反比。

1.2 基本鲸鱼优化算法

WOA 算法将种群个体的位置信息作为特征信息进行交互。澳大利亚研究学者 Seyedali Mirjalili 将座头鲸群捕食过程划分为 3 个阶段: 包围猎物阶段、气泡网攻击阶段、以及搜寻猎物阶段。并提出三类更新机制: 收缩环绕包围机制、螺旋更新机制、以及随机游走机制。采用何种机制由系数 a 及对应的 $|A|$ 控制。系数生成方式为:

$$\begin{aligned} A &= 2 \cdot a \cdot r - a \\ C &= 2 \cdot r \\ a &= 2(1 - t/\text{Max_iter}) \end{aligned} \quad (6)$$

系数 A 、 C 由线性递减的系数 a 与限定范围为 $[0,1]$ 的随机数 r 构成。初始文献 [11] 限制 $|A| \leq 2$ 。鲸鱼群处于气泡网攻击阶段时, 包含收缩环绕包围与螺旋式更新两种位置更新机制。收缩环绕包围机制确定的下一代位置 $\vec{X}(n+1)$ 表示为:

$$\begin{aligned} D &= |\vec{C}\vec{X}^*(n) - \vec{X}(n)| \\ \vec{X}(n+1) &= \vec{X}^*(n) - \vec{A} \cdot \vec{D} \end{aligned} \quad (7)$$

式中, n 表示迭代次数, 更新系数向量 \vec{A} 与 \vec{C} 由式 (6) 确定, $\vec{X}^*(n)$ 表示第 n 次迭代时鲸鱼的最优位置, $\vec{X}(n)$ 为当前鲸鱼的位置向量。

螺旋式更新机制数学模型表示为:

$$\begin{aligned} d &= |\vec{X}^*(n) - \vec{X}(n)| \\ \vec{X}(n+1) &= \vec{d}e^{bl} \cos(2\pi l) + \vec{X}^*(n) \end{aligned} \quad (8)$$

式中, d 表示当前鲸鱼与猎物一最优解之间的距离。 b 为对数螺旋的形状参数, 一般设置为 1, l 为 $[-1,1]$ 内的随机系数。

WOA 算法采用随机游走机制模拟鲸鱼群的搜索猎物阶段。此阶段鲸鱼相互选择进行位置的随机更新。更新法则为:

$$\begin{aligned} K &= |C \times \vec{X}_{\text{rand}} - \vec{X}| \\ \vec{X}(n+1) &= \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{K} \end{aligned} \quad (9)$$

三类更新机制关于系数 $|A|$ 与随机概率 p 的关系可归纳为:

$$\begin{cases} \vec{X}(n+1) = \vec{X}^*(n) - \vec{A} \cdot \vec{D} p < 0.5, & |\vec{A}| < 1 \\ \vec{X}(n+1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{K} p < 0.5, & |\vec{A}| \geq 1 \\ \vec{X}(n+1) = \vec{d}e^{bl} \cos(2\pi l) + \vec{X}^*(n) & p \geq 0.5 \end{cases} \quad (10)$$

WOA 依靠随机游走机制进行全局搜索, 依靠收缩包围

机制与螺旋式更新机制进行局部开发。

2 基于预警机制鲸鱼优化算法的无人机航迹规划

2.1 改进算法设计

2.1.1 混沌理论的初始化

目前几乎所有的群智能优化算法都属于随机性质的优化算法。鲸鱼群的初始化方式对算法的后续走向有着重要影响。使用传统的随机函数容易引入计算机的伪随机性。混沌映射可以用于替代伪随机数生成器, 其生成的序列在解空间内分布较为均匀, 因此能提高初始种群的遍历性。采用混沌理论中的 Logistic 映射完成对算法随机初始化的改进, 计算式表示为:

$$x(n+1) = \mu x(n)(1 - x(n)) \quad (11)$$

μ 为可调参数, 且 $\mu \in [0,4]$ 。除去 Logistic 映射, Tent 映射也被广泛使用。式 (9) 定义的游走机制通过随机选取种群个体进行更新的方式, 扩大了算法的搜索范围, 一定程度提供了算法跳出局部最优的能力, 但较优个体不可避免地会出现跟随较差个体随机游走地状况, 出现个体搜索倒退、收敛缓慢的情况。为避免过度盲目性, 以最优位置为导向, 添加莱维飞行扰动保证一定程度的随机性。计算方法表示为:

$$\begin{aligned} \vec{X}(n+1) &= \vec{X}(n) + \alpha \oplus \text{Levi}(\beta) \\ \alpha \oplus \text{Levi}(\beta) &= 0.01 \frac{\mu}{|\nu|^{1/\beta}} (\vec{X}(n) - \vec{X}^*(n)) \end{aligned} \quad (12)$$

式中, $\alpha \oplus \text{Levi}(\beta)$ 为莱维飞行添加项。其中 \oplus 表示元素相乘, α 为比例系数, β 为参数, 通常取 1.5, μ 与 ν 均为服从标准正态分布的随机参数。

2.1.2 预警机制的自适应概率选取

式 (10) 表明 WOA 执行螺旋式更新或其他两种机制取决于 p , 各自发生率为 50%。概率 p 实质控制种群处于局部开发与全局搜索的何种阶段。本文依据个体适应度对种群个体的 p 数值进行自适应选取。对排名靠后的个体发出预警, 适应度越差, 在 $|A| > 1$ 时, 其游走机率越大。在保持个体差异化的同时, 优化收敛能力。个体概率 p_i 表示为:

$$p_i = \frac{\text{rank}(\text{fitness}_i)}{N} \quad (13)$$

第 i 个体游走的概率与自身适应度排名有关, N 为个体总数, $\text{rank}(\text{fitness}_i)$ 为个体适应度排名。游走概率 p_i 限定在范围 $\text{rank}(\text{fitness}_i)$ 。

2.1.3 自适应权重系数

收缩包围与游走机制处于 $|A| > 1$ 范围内, $|A| < 1$ 范围内仍需进行探索能力与开发能力的平衡。在 $|A| < 1$ 内的螺旋更新机制内, 添加权重因子 w 控制螺旋更新的范围, 更新机制表示为:

$$X_i^{t+1} = |X_{\text{best}}^t - X_i^t| \cdot e^{bl} \cdot \cos(2\pi l) * w + X_{\text{best}}^t \quad (14)$$

其中: w 参数的大小依据式 (13) 预警机制的概率 p_i 决定, 表达式为:

$$\omega_i = 0.75 + p_i \quad (15)$$

式 (15) 表明个体 i 的权重因子 ω_i 限定范围 $[0.75, 1.25]$ 。当 $\omega_i < 1$ 时, 收紧位置更新, 当 $\omega_i > 1$ 时, 放宽位置更新, 扩大更新范围。由 ω_i 所处范围不难看出收紧与放宽概率各占 50%。

综上, APWOA 算法的基本步骤如下:

APWOA 的 Python 伪代码

Setting: lb, ub, N, D, ObjFun, and T

Logistic 映射随机初始化种群位置; 初始化种群最优解, 式 (11);

while ($t \leq T$):

边界检测与处理;

根据航机代价定义公式计算种群个体适应度;

更新最优解 $best_fitness, best_position$

for i in range(1, N) do

根据适应度的排序计算各个体的预警概率 p , 式 (13);

根据预警概率 p 计算出自适应权重 ω ,

式 (15);

If $p < 0.5$:

If $|A| < 1$:

依据添加莱维飞行的游走机制更新位置信息, 式 (12);

else:

包围猎物, 依据收缩包围机制更新位置信息, 式 (7);

else:

依据自适应权重的螺旋机制进行位置更新, 式 (14);

更新最优解 $best_fitness, best_position$

$t = t + 1$

end while

return $best_fitness, best_position$

2.2 考虑算法时间的航迹代价

常规航迹规划指标考虑无人机自身性能约束条件、以及地形威胁、雷达威胁等威胁条件。自身性能约束可分为角度代价、航程代价、高度代价。

航程代价指标是航迹代价函数的重要组成部分, 航迹规划在多数情况下以寻求最短的安全可飞行航程为主要目标。算法规划得出的节点序列, 将航路划分为对应的航迹分段, 则总航程代价为各分段之和, 表示为:

$$J_{\text{path}} = \sum_{i=1}^{N-1} \text{path}(i, i+1) \quad (16)$$

i 为当前航迹节点, path 为第 $(i, i+1)$ 段航程, 通过计算节点之间距离获得。

考虑爬升角/下滑角为主要角度代价指标。航迹序列中的每一个航迹段均需满足爬升角度与下滑角度的约束条件。超出最大角度约束则认为该航迹不可飞行, 优化失败。

角度代价指标 J_{angle} 表示为:

$$J_{\text{angle}} = \begin{cases} \sum_{i=1}^n |\theta_i| & \text{if all } \theta_i \leq \theta_{\max} \\ C_a & \text{other} \end{cases} \quad (17)$$

若存在任意航迹段的爬升角/下滑角 θ_i 大于最大角度,

则给予较大的惩罚值 C_a , 约束条件内, 总的角度代价 J_{θ} 为各代价指标总和。 J_{θ} 越小, 表明角度变化越小, 则规划出的轨迹越平滑, 这是角度纳入航迹代价的重要原因之一。

战场存在雷达威胁的条件下, 还应考虑飞行高度代价指标。较高的飞行高度会加大被雷达发现的概率, 而过低的飞行高度则难以应对起伏的地形。定义安全高度 H_{safe} 为当前飞行高度与地面的最小安全距离。高度代价指标 J_{height} 表示为:

$$J_{\text{height}} = \sum_{i=1}^n |h_i - H_{\text{safe}}| \quad (18)$$

威胁 J_{threat} 主要考虑地形威胁与雷达威胁两种。式 (2) 为山峰地形威胁的计算方式。为保证存在雷达威胁情况下的无人机安全性, 定义雷达绝对威胁领域, 该领域内无人机毁伤概率为 100%, 则雷达威胁指标表示为:

$$J_{\text{radar}} = \begin{cases} \sum_{j=1}^m T_{\text{Radar}}^j & d_r \geq R^j \\ C_r & d_r < R^j \end{cases} \quad (19)$$

d_r 为无人机与雷达 j 之间的距离, R^j 为雷达 j 的领域半径, C_r 为无人机落在领域内的惩罚值。由此可得总威胁代价 J_{threat} 为:

$$J_{\text{threat}} = T_{\text{terrain}} + J_{\text{radar}} \quad (20)$$

综上所述, 本文定义常规航迹代价 J_{cost} 为:

$$J_{\text{cost}} = \omega_1 J_{\text{path}} + \omega_2 J_{\text{angle}} + \omega_3 J_{\text{height}} + \omega_4 J_{\text{threat}} \quad (21)$$

式中, J_{path} 为航程代价, J_{angle} 为爬升角代价, J_{height} 为高度代价, 存在雷达威胁的战场环境下, 应控制无人机的飞行高度, J_{threat} 为威胁代价, 包括第二章节提及的地形山峰威胁以及雷达威胁。 $\omega_1, \omega_2, \omega_3, \omega_4$ 为各航迹代价的权重系数。

为完成实时航迹规划的实时性要求, 本文将优化算法的求解时间内引入航迹代价函数。时间代价 J_{time} 与航程代价 J_{path} 具有相同的表达式, 即为:

$$J_{\text{time}} = \begin{cases} \text{time} & \text{time} < T_{\max} \\ C & \text{time} > T_{\max} \end{cases} \quad (22)$$

time 为算法的实际运行时间, T_{\max} 为预定的规划时间上限, 应与威胁数量、规划距离成正相关。 C 为施加的惩罚值, 惩罚值数值根据实际情况自定义选取, 实时性要求极高的环境下, C 可设置为无穷大, 保证实时性的最高优先级。

综合常规航迹代价以及时间代价, 总航迹代价 J_{total} 表示为:

$$J_{\text{total}} = \alpha 1 J_{\text{cost}} + \beta 1 J_{\text{time}} \quad (23)$$

$\alpha 1$ 与 $\beta 1$ 为常规航迹代价 J_{cost} 、运行时间代价 J_{time} 相对应的权重系数。在实际使用时依据需求自适应选取, 面临威胁环境复杂、远距离航迹情况, 优先保证无人机安全抵达, 应着重考量威胁代价以及航迹代价, 可适当提高 $\alpha 1$ 权重。反之, 短距离、实时性要求高的情况, 提高 $\beta 1$ 权重以增加机动性。

2.3 可变维度的航迹节点序列

WOA 算法中，每个个体对应一条候选航迹规划，航迹的生成方法定义个体信息的表达方式。节点数量决定算法优化问题的维度。针对三维航迹规划，本文采用坐标等分法将其简化为二维航迹规划，以此，算法的优化维度简化为水平面内的 (x, y) 两个维度。 x 轴坐标等分法示意图如图 2 所示，仅为作图方便，山峰威胁与雷达威胁投影视为圆形。

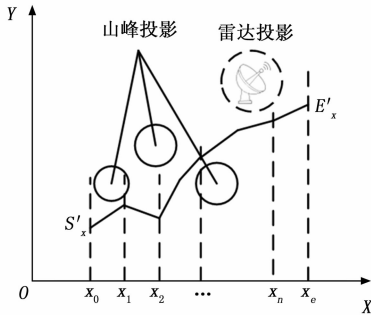


图 2 x 轴坐标等分法示意图

单独使用 x 坐标等分法或 y 坐标等分法会造成算法模型失效的风险。如只采用 x 坐标等分法，模型会在无人机起始点与目标点 x 轴坐标相等情况时失效，且当 x 坐标变化远小于 y 坐标变化时，计算的规划航迹点过少，导致航迹不具有飞行价值。为防止上述问题，本文采用联合坐标等分法。选取 x 坐标等分法与 y 轴坐标等分法的结果最大值确定航迹待规划节点数量。避免单一坐标等分法失效的同时，规划较多数量的节点保证航迹的可行性。节点数量 $point_n$ 表示为：

$$point_n = \frac{\max\{dis(x_i, x_e), dis(y_i, y_e)\}}{step} \quad (24)$$

$step$ 为规划步长，固定起始点与目标点情况下控制节点数量，改变 $step$ 即实现航迹规划问题的维度。航迹规划节点之间，常使用采样的方法进行碰撞点检测，起到降低规划维度的同时还可以对轨迹完成优化。虽然改变起始点与坐标点的方法可以实现对不同维度的算法仿真，但会引入不同雷达威胁与山峰威胁，变量过多导致对比困难。本节设计步长 $step$ 控制的可变维度航迹生成方法则更方便仿真数据对比。

3 仿真实验与分析

3.1 基准函数测试与分析

基准测试函数是进行算法性能测试的基本工具，为分析本文改进 WOA 算法的优化性能，采用文献提及的 18 个广泛使用的基准测试函数进行性能测试。选取群智能优化算法中典型的 PSO 算法、原 WOA 算法^[11]、使用莱维飞行该进的灰狼算法 LGWO^[20]，以及文献最新改进的基于莱维飞行的 WOA 算法 LWOA^[21] 进行测试比较。测试函数的基本信息参考文献 [22]。仿真实验的硬件平台为联想笔记本 AIR14 计算机，运行内存为 16 g，CPU 为 AMD Ryzen 4800U，集成显卡为 AMD 512mb。软件平台为：MATLAB

R2016b，以及 Pycharm 2021。

设置种群数目为 30，迭代次数 500 次。除此之外，上文提及的对比算法的必要参数设置如表 1。关于基准测试函数， $f1 \sim f5$ 为高维基准函数，选定维度为 30。 $f14 \sim f18$ 为多峰值基准函数，维度以及搜索范围遵循自定义。为降低实验偶然性，每类算法独立运行 30 次取平均值。测试结果（最小值、平均值、方差）汇总至表 2。

表 1 各算法参数设置

算法	参数
PSO	惯性权重上限 $\omega_{\max} = 0.8$ ， 下限 $\omega_{\min} = 0.2$ 学习因子 $c_1 = c_2 = 2$
WOA	$a1 = 2 - t * 2/T, a2 = -1 + t * (-1)/T$
LGWO	$a = 2 - t * 2/T$
LWOA	$a1 = 2 - t * 2/T, a2 = -1 + t * (-1)/T$

$f1 \sim f5$ 的高维函数测试中，WOA 算法相比 PSO 算法，本身就具有较高的收敛能力。但本文改进的 APWOA 算法比只添加莱维飞行扰动的 LWOA 算法具备更高的收敛精度与收敛稳定性。APWOA 在 $f1 \sim f3$ 的测试中表现出优异特性，但 APWOA 与 LWOA 算法在 $f4 \sim f5$ 表现落后于 LGWO 算法。关于多峰函数 $f15 \sim f18$ 测试，APWOA 能够搜索出 $f16 \sim f18$ 的理论最优解，并在 $f15$ 上搜索出相比于其他算法的较小值。综合而言，基准函数的测试仿真表明，在保证低维函数的局部寻优能力的同时，APWOA 在高维函数的收敛速度与精度得到明显改善。

3.2 三维航迹规划测试

为验证改进的 APWOA 算法对于时间约束的航迹规划的有效性，基于 Pycharm 开发平台和 python 开发语言进行仿真验证，并使用 Matlab2016b 进行数据处理与图像绘制。

3.2.1 变维度的航迹求解精度测试

仿真场景数据由式 (1) 计算得出，DEM 的范围为 $100 \text{ km} \times 100 \text{ km} \times 500 \text{ m}$ 。式 (1) 地形函数设置为 $a = 0.1, b = 0.01, c = 1, d = 0.1, e = 0.2, f = 0.4, g = 0.02$ 。除去基础地形外，添加雷达作为战场威胁的代表。初始两个雷达的水平面坐标为 $(65, 10)$ 、 $(12, 85)$ 。雷达不可飞行的区域半径为 10 km，设定该领域内无人机毁伤概率为 100%，领域外雷达威胁遵从式 (5) 计算。航迹代价求解的维度为待规划的节点数目，本文使用改变步长的方法实现节点数目、以及维度的变化。航迹步长分别选取 8 km、5 km 以及 3 km。为进行维度变化的测试，选定起始点为 $(3, 3, 60)$ ，目标点为 $(92, 58, 60)$ ，并固定每个航迹分段的碰撞点检测数目，以控制计算量的数据误差。比较上文基准函数对比试验参与的 PSO 算法、WOA 算法与本文的 APWOA 算法的航迹解。为同时获取算法收敛精度与收敛速度信息，迭代次数选取 200 次。每类算法的每个步长维度运行 10 次取平均值，汇总结果（平均航迹代价、单次迭代运行时间、运行方差）至表 3。

表 2 基准函数测试结果

F	结果	PSO	WOA	LWOA	LGWO	APWOA	均值排名
f1	最小值	9.3717×10^{-3}	1.4378×10^{-85}	7.2878×10^{-92}	1.2694×10^{-32}	3.4833×10^{-174}	1
	平均值	5.227	1.5731×10^{-75}	9.4947×10^{-84}	7.0686×10^{-30}	1.5536×10^{-162}	
	方差	6.6795	2.7385×10^{-68}	4.6242×10^{-78}	2.1889×10^{-29}	7.6998×10^{-159}	
f2	最小值	1.8075×10^{-2}	1.9034×10^{-47}	2.1911×10^{-61}	1.2684×10^{-19}	1.1781×10^{-101}	1
	平均值	4.6260×10^{-2}	1.3005×10^{-43}	3.0238×10^{-54}	8.7857×10^{-19}	2.1416×10^{-94}	
	方差	0.1145	6.8830×10^{-43}	1.6166×10^{-51}	1.0765×10^{-18}	6.9401×10^{-92}	
f3	最小值	1110.7538	6269.5566	374.8022	5.3209×10^{-10}	1.6232×10^{-100}	1
	平均值	3733.6097	14860.13530	18486.4419	7.0820×10^{-17}	3.6377×10^{-85}	
	方差	1936.0792	5712.9469	374.8022	3.0067×10^{-06}	1.8186×10^{-83}	
f4	最小值	9.3732	0.1338	0.1101	3.1662×10^{-08}	6.4577×10^{-05}	2
	平均值	16.45038	24.8624	35.9412	6.2880×10^{-07}	5.8673×10^{-4}	
	方差	3.6379	17.4758	16.8817	7.7521×10^{-07}	1.6916×10^{-2}	
f5	最小值	29.8183	26.9914	28.5332	27.5294	28.7026	3
	平均值	180.1614	27.8590	28.7500	27.7748	28.7195	
	方差	166.1605	0.4619	0.4802	0.6540	0.0079	
	平均值	0.9980	2.9615	5.9641	4.7822	2.2727	
	方差	0.0	3.1789	5.0468	4.3030	2.1756	
f15	最小值	3.0748×10^{-4}	3.0996×10^{-3}	3.5458×10^{-3}	3.0749×10^{-4}	3.0964×10^{-5}	1
	平均值	5.5251×10^{-4}	5.9×10^{-3}	0.0015	4.5696×10^{-4}	2.1149×10^{-4}	
	方差	2.5138×10^{-4}	2.6759×10^{-4}	0.0018	3.4875×10^{-4}	3.200×10^{-4}	
f16	最小值	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	1
	平均值	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	
	方差	4.3284×10^{-16}	1.8728×10^{-9}	1.0018×10^{-6}	2.2112×10^{-8}	6.5034×10^{-11}	
f17	最小值	0.3979	0.3979	0.3979	0.3979	0.3979	1
	平均值	0.3979	0.3979	0.3979	0.3979	0.3979	
	方差	0	3.5787×10^{-6}	3.0300×10^{-5}	1.0872×10^{-6}	1.0602×10^{-8}	
f18	最小值	3.0000	3.0000	3.0000	3.0000	3.0000	1
	平均值	3.0000	3.0000	6.0183	3.0000	3.0000	
	方差	1.8400×10^{-14}	1.8400×10^{-14}	9.0824	1.9910×10^{-6}	0	

表 3 不同维度的航迹规划测试结果

算法	规划节点数目	碰撞检测节点数目	平均航迹代价	单次运行时间/ms	方差
PSO	10	36	62.1322	58.75	4.5101
	20	76	76.8585	90.80	30.94496
	32	124	129.1628	147.09	529.1148
WOA	10	36	61.7642	61.12	14.2111
	20	76	72.5404	95.91	0.5674
	32	124	92.6017	153.50	0.2651
APWOA	10	36	56.8233	60.35	8.5650
	20	76	66.2732	96.09	0.8493
	32	124	87.5005	154.17	0.1261

从表 3 看出, APWOA 算法以及 WOA 算法相较于 PSO 算法, 面对高维航迹规划仍然可以得出较优解, 且规划解优势明显。而相较于 WOA 算法, APWOA 算法在运行时间基本持平的情况下, 可获得代价更小的规划解。10 维度规划中相较 WOA 算法提升 8.0%, 32 维度规划中相较 WOA 算法提升 5.5%。这表明 APWOA 算法具备更高的收

敛精度。实际航迹规划采用的步长应根据目标点起始点距离、地形复杂度、威胁数目综合考量, 部分文献甚至设有最小规划步长。表中选取 20, 30 高维度情况多作为极限测试以及数据分析使用。根据表 3, 实验选取 3 种算法的航迹均值作为典型代表, 关于 8 km 步长 10 维度的航迹规划路线如图 3 所示。关于 3 km 步长 32 维度的航迹规划路线如图 4 所示。

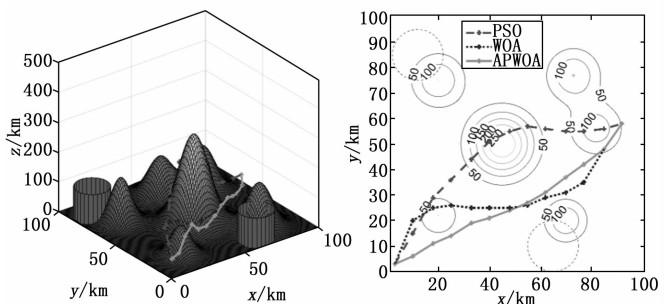


图 3 8 km 步长的航迹规划结果对比

图 3 中, 在 8 km 步长, 维度较低的测试中, 3 种算法

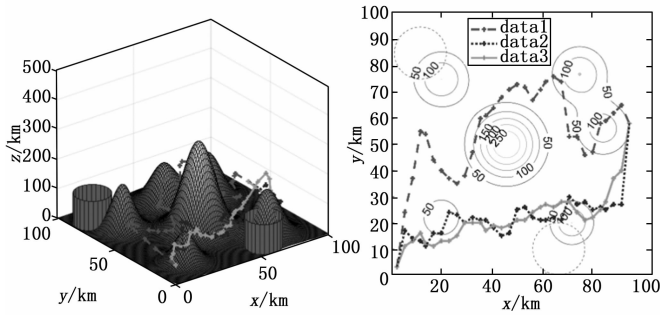


图 4 3 km 步长的航迹规划结果对比

均可以搜索到可飞行的航迹路线。其中 APWOA 算法规划出的路线最短, 表明其优化能力相对较强, PSO 算法相对于其他算法, 相同迭代次数花费较少的运行时间, 所以 3 种算法求解出的时间约束的航迹代价差距较小。在图 4 的 3 km 步长的高维度测试下, APWOA 算法解算出的航迹代价则远优于 PSO 算法, 相较于原 WOA 算法也具有明显提升。

3.2.2 航迹求解实时性测试

使用上节的测试数据, 取 3 km 步长与 8 km 步长规划中接近航迹代价平均值的测试路线, 绘制 3 种算法关于迭代次数的收敛曲线, 如图 5 所示。

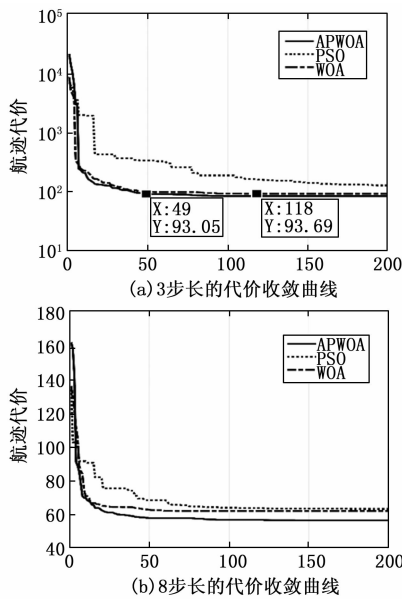


图 5 不同步长各算法的收敛曲线

图 5 (a) 中, APWOA 早 69 次迭代获得 WOA 算法可获得的最优解, 依据表 3 中 WOA 的单个迭代运行时间, 则 APWOA 算法在 3 km 步长 32 维度的极限测试下可节省时间成本 6.7 s 左右, 占 WOA 算法总用时的 58.5%。在正常维度的 (b) 图中, 基于 APWOA 算法的航迹规划实时性依旧具有不俗提升, APWOA 算法 20 次迭代内即可获取优于 WOA 算法的航迹代价, 总耗时 1.2 s。

4 结束语

本文针对无人机三维航迹规划问题, 主要进行两方面

的改进研究。针对航迹优化模型, 将算法运行时间计入航迹代价, 建立起规划航迹与运行时间的互反馈通道。针对优化求解算法, 本文提出基于预警机制的鲸鱼优化算法。对 3 种机制的选取概率实现自适应, 并依次概率实现对螺旋更新机制的收缩控制, 实现全局探索与局部优化的平衡。关于算法验证, 使用可变维度的航迹生成方法, 验证了算法的有效性与优越性。

参考文献:

- [1] 付鑫, 赵然, 梁延峰, 等. 反无人机蜂群技术发展综述 [J]. 中国电子科学研究院学报, 2022, 17 (5): 421-428.
- [2] 张远, 黄旭, 路坤锋, 等. 高超声速飞行器控制技术研究与展望 [J]. 宇航学报, 2022, 43 (7): 866-879.
- [3] 陈香敏, 吴莹. 基于 Voronoi 图的 UAV 攻击多移动目标的路径规划算法研究 [J]. 信息通信, 2020 (6): 36-37.
- [4] 李克玉, 陆永耕, 鲍世通, 等. 基于改进 RRT 算法的无人机三维避障规划 [J]. 计算机仿真, 2021, 38 (8): 59-63, 96.
- [5] 李宪强, 马戎, 张伸, 等. 蚁群算法的改进设计及在航迹规划中的应用 [J]. 航空学报, 2020, 41 (S2): 213-219.
- [6] 付兴武, 胡洋. 基于改进粒子群算法的三维路径规划 [J]. 电光与控制, 2021, 28 (3): 86-89.
- [7] DEWANGAN R K, SHUKLA A, GODFREY W W. Three dimensional path planning using Grey wolf optimizer for UAVs [J]. Applied Intelligence, 2019, 49 (6): 2201-2217.
- [8] 吴坤, 谭劭昌. 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.
- [9] CUI Z, WANG Y. UAV path planning based on multi-layer reinforcement learning technique [J]. IEEE Access, 2021, 9: 59486-59497.
- [10] 于盛. 基于强化学习多无人机路径规划算法研究及实现 [D]. 哈尔滨: 哈尔滨工业大学, 2020.
- [11] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95 (12): 51-67.
- [12] ZHANG H, TANG L, YANG C, et al. Locating electric vehicle charging stations with service capacity using the improved whale optimization algorithm [J]. Advanced Engineering Informatics, 2019, 41: 100901.
- [13] LIU M, YAO X F, LI Y X. Hybrid whale optimization algorithm enhanced with Lévy flight and differential evolution for job shop scheduling problems [J]. Applied Soft Computing, 2020, 87: 105954.
- [14] JIANG R Y, YANG M, WANG S Y, et al. An improved whale optimization algorithm with armed force program and strategic adjustment [J]. Applied Mathematical Modelling, 2020, 81: 603-623.
- [15] KAVEH A, MOGHADDAM M R. A hybrid WOA-CBO algorithm for construction site layout planning problem [J]. Scientia Iranica, 2018, 25 (3): 1094-1104.

(下转第 173 页)