

7-15-2024

UAV Path Planning Based on Improved Harris Hawk Algorithm and B-spline Curve

Zhifeng Huang

School of Management, University of Shanghai for Science and Technology, Shanghai 200082, China,
213491529@st.usst.edu.cn

Yuanhua Liu

liuyuanhua@st.usst.edu.cn

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact xtfzxb@126.com.

UAV Path Planning Based on Improved Harris Hawk Algorithm and B-spline Curve

Abstract

Abstract: Aiming at the global path planning problem of unmanned aerial vehicles (UAVs) in dynamic environments, this paper proposes an improved Harris Hawk optimization algorithm. To address the problem of insufficient search performance in the later stage of the algorithm, an adaptive chaos and core population dynamic partitioning strategy is proposed to improve the searchability of the algorithm in the later stage. The Harris Hawk update formula is modified, and the golden sine strategy is introduced to improve the search efficiency of the algorithm. Then, an adaptive dynamic cloud optimal solution perturbation strategy is integrated to improve the ability of the algorithm to jump out of the local extremum. For the three-dimensional grid path planning problem, a valuation function is established. By calculating the cost of reaching the endpoint for each grid, the algorithm is aided in filtering nodes, allowing it to search for a shorter path. For the problem of the non-smooth path, the path angle is processed by using the cubic B-spline curve for three times to make the path more suitable for UAV flight. The effectiveness of the improved algorithm is validated by simulation experiments on international standard test functions and static and dynamic grid maps of varying sizes and complexity. The experimental results demonstrate that the proposed algorithm significantly outperforms the control group algorithm. On average, the planned path is shortened by 14.94% and the number of corners is reduced by 53.31%.

Keywords

Harris Hawk optimization(HHO), three-dimensional path planning, UAV, dynamic environment, self-adaption

Recommended Citation

Huang Zhifeng, Liu Yuanhua. UAV Path Planning Based on Improved Harris Hawk Algorithm and Bspline Curve[J]. Journal of System Simulation, 2024, 36(7): 1509-1524.

基于改进哈里斯鹰和B-spline曲线的无人机路径规划研究

黄志锋, 刘媛华*

(上海理工大学 管理学院, 上海 200082)

摘要: 针对无人机在动态环境中的全局路径规划问题, 提出了一种改进哈里斯鹰优化算法。针对算法后期搜索性能不足等问题, 提出自适应混沌和核心种群动态划分策略, 提高算法后期的搜索能力; 修改哈里斯鹰更新公式, 引入黄金正弦策略, 提高算法搜索效率; 融合自适应动态云最优解扰动策略, 提高算法跳出局部极值的能力; 针对三维栅格路径规划问题, 设置了一种估值函数, 通过计算栅格到达终点的代价, 帮助算法进行节点筛选, 使算法能搜索到更短路径, 并针对路径转角不平滑的问题, 使用3次B-spline曲线对路径转角进行处理, 使路径更适合无人机飞行。通过国际标准测试函数和在不同大小、不同复杂程度的静态、动态栅格地图进行仿真实验。实验结果显示, 本文算法相较于对比算法, 规划出的路径平均缩短了14.94%、转角数量平均减少了53.31%。

关键词: 哈里斯鹰优化算法; 三维路径规划; 无人机; 动态环境; 自适应

中图分类号: TP391.9; TP18 文献标志码: A 文章编号: 1004-731X(2024)07-1509-16

DOI: 10.16182/j.issn1004731x.joss.23-0403

引用格式: 黄志锋, 刘媛华. 基于改进哈里斯鹰和B-spline曲线的无人机路径规划研究[J]. 系统仿真学报, 2024, 36(7): 1509-1524.

Reference format: Huang Zhifeng, Liu Yuanhua. UAV Path Planning Based on Improved Harris Hawk Algorithm and B-spline Curve[J]. Journal of System Simulation, 2024, 36(7): 1509-1524.

UAV Path Planning Based on Improved Harris Hawk Algorithm and B-spline Curve

Huang Zhifeng, Liu Yuanhua*

(School of Management, University of Shanghai for Science and Technology, Shanghai 200082, China)

Abstract: Aiming at the global path planning problem of unmanned aerial vehicles (UAVs) in dynamic environments, this paper proposes an improved Harris Hawk optimization algorithm. To address the problem of insufficient search performance in the later stage of the algorithm, an adaptive chaos and core population dynamic partitioning strategy is proposed to improve the searchability of the algorithm in the later stage. The Harris Hawk update formula is modified, and the golden sine strategy is introduced to improve the search efficiency of the algorithm. Then, an adaptive dynamic cloud optimal solution perturbation strategy is integrated to improve the ability of the algorithm to jump out of the local extremum. For the three-dimensional grid path planning problem, a valuation function is established. By calculating the cost of reaching the endpoint for each grid, the algorithm is aided in filtering nodes, allowing it to search for a shorter path. For the problem of the non-smooth path, the path angle is processed by using the cubic B-spline curve for three times to make the path more suitable for UAV flight. The effectiveness of the improved algorithm is validated by simulation experiments on international

收稿日期: 2023-04-09 修回日期: 2023-05-10

基金项目: 国家自然科学基金(72071130)

第一作者: 黄志锋(1998-), 男, 硕士生, 研究方向为路径规划。E-mail: 213491529@st.usst.edu.cn

通讯作者: 刘媛华(1974-), 女, 副教授, 硕士, 博士, 研究方向为智能算法优化。E-mail: liuyuanhua@st.usst.edu.cn

standard test functions and static and dynamic grid maps of varying sizes and complexity. The experimental results demonstrate that the proposed algorithm significantly outperforms the control group algorithm. On average, the planned path is shortened by 14.94% and the number of corners is reduced by 53.31%.

Keywords: Harris Hawk optimization(HHO); three-dimensional path planning; UAV; dynamic environment; self-adaption

0 引言

无人机广泛应用于军事侦察、物流运输等领域，是一种高效、省时的方法。但是人类操控的飞行路径，可能不是最短路径，这会增加无人机的能耗，因此许多学者使用算法进行无人机的路径规划，以达到提高效率、缩短路径的目的。

目前应用在路径规划当中的算法主要有A*算法^[1-2]、人工势场(artificial potential field, APF)算法^[3]、狮群优化(lion swarm optimization, LSO)算法^[4]、鲸鱼优化算法(whale optimization algorithm, WOA)^[5-6]和哈里斯鹰优化(Harris Hawk optimization, HHO)算法^[7]等。A*算法由Dijkstra发展而来，其原理是通过计算每个栅格到达终点的代价从而选择最短一条路径。文献[8]提出了一种改进步长和变权值路径评估函数的A*算法，使之能够应用在三维空间中，同时为了满足无人机运动约束，提出了一种航线优化算法，能够有效进APF力场中电荷，受到引力和斥力进行移动和避障的算法，其优点是方法简单易于理解。文献[9]针对APF算法部分目标不可达的缺点，引入是否陷入局部最优的判别机制，并结合一种直角反向运动，帮助算法跳出局部最优值。文献[10-11]提出一种狮群算法应用在无人机路径规划当中，但是只考虑了静态问题，没有考虑动态障碍物的影响。WOA是这几年新提出的仿生智能优化算法，已经广泛应用在各个领域。文献[12]对WOA进行改进，提出一种信息交流机制的改进鲸鱼优化算法，能够通过最优解个体和领域最优解来更新个体位置，使之能够更好平衡局部和全局的开发能力，但是该研究没有精确的三维建模，算法性能

有待商榷。HHO算法具有参数少、原理简单和寻优速度快等优点^[13-15]，但性能可以进一步提高。

目前，大多数无人机路径规划相关论文只研究静态环境下的路径规划，没有在动态环境当中对算法性能进行充分验证；此外，相关硬件技术已经能够支撑无人机进行实时动态避障，但动态路径规划理论研究不足，且存在以下3个问题：①已有研究主要为静态无人机路径规划，没有考虑动态障碍物；②算法性能不足，一些已有算法改进后仍然有提高的空间；③规划出的路径不够平滑，不适合无人机的飞行。

本文针对上述3个问题，提出了一种改进哈里斯鹰算法应用于动态环境的无人机路径规划问题。融合自适应混沌和核心种群动态划分策略，平衡算法全局和局部搜索性能；对哈里斯鹰更新公式进行修改，融合黄金正弦策略，提高算法搜索效率；融合自适应动态云扰动策略，提高算法后期跳出局部最优解的能力；针对三维栅格路径规划问题，设置了一种估值函数，进行节点筛选，帮助算法选择下一节点，达到快速寻优的目的；针对栅格法下规划的路径不平滑的缺点，结合3次B-spline曲线，对路径转角处进行平滑处理，使路径更适合无人机飞行。

1 模型建立和路径处理

1.1 建立三维空间

目前，主流的空间环境建模采用栅格法，每个栅格为路径规划中的节点。栅格地图原点坐标为(0,0,0)，以 \mathbf{x}_{star} 为起始点， \mathbf{x}_{end} 为目标点，黑色矩形模拟城市中的建筑，如图1所示。假设建筑物由 a 、 b 和 c 正立方体栅格组成，其中， $a = \text{int}(x_{\text{max}}/N_{\text{length}})$ ，

$b = \text{int}(y_{\max}/N_{\text{length}})$, $c = \text{int}(z_{\max}/N_{\text{length}})$, N_{length} 为每个栅格立方体的长度, x_{\max} 、 y_{\max} 和 z_{\max} 为所选长方体区域的长度、宽度和高度, $\text{int}(\cdot)$ 为取整函数。考虑城市存在信号干扰区域, 使用红色球体表示信号干扰源的范围, 无人机需要绕开信号干扰源进行飞行。

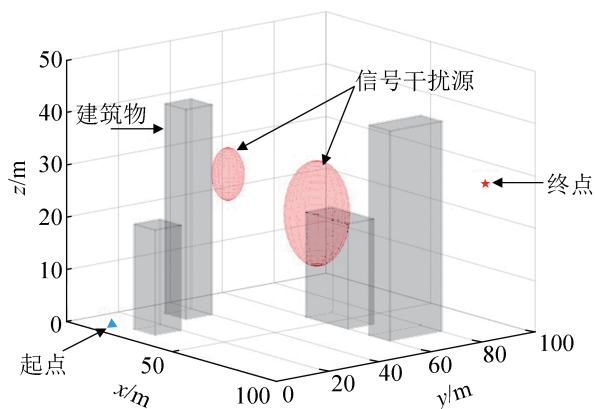


图 1 三维空间示例图
Fig. 1 Three-dimensional space examples

$(x_1, y_1), (x_2, y_2), \dots, (x_{\beta_\gamma}, y_{\beta_\gamma})$ 为第 γ 个建筑物, 建筑物高度为 h_γ , β_γ 为 γ 建筑物坐标的平面坐标数量, 其中 $\gamma \in (1, n)$, 建筑物数量为 n 。第 δ 个信号干扰源的坐标为 $(x_\delta, y_\delta, z_\delta)$, 设置信号干扰源的半径为 R , 范围则是 $(4/3)\pi R^3$ 。则无人机飞行的约束区域为

$$\sum_{\gamma=1}^n \left(\begin{matrix} (x_1, y_1, 0), (x_2, y_2, 0), \dots, (x_{\beta_\gamma}, y_{\beta_\gamma}, 0), \\ (x_1, y_1, h_\gamma), (x_2, y_2, h_\gamma), \dots, (x_{\beta_\gamma}, y_{\beta_\gamma}, h_\gamma) \end{matrix} \right) + \sum_{\delta=1}^n \left((x_\delta, y_\delta, z_\delta) [(4/3)\pi R_\delta^3] \right)$$

1.2 动态障碍物和路径规划

城市空中路径规划是指无人机根据获取的城市建筑和环境信息, 规划出一条无碰撞且适合无人机飞行的路径。目前, 已知大疆 T50 无人机配备两组有源相控阵雷达和双目视觉系统, 能够将地形与障碍物转换为三维空间地图中占用的栅格, 相控阵雷达能够侦测 2 km 范围内的动态障碍物。考虑到实际空间中存在动态障碍物和 UAV 无法到

达的区域, 对 UAV 约束条件和有效路径的生成规则做如下补充:

(1) 无人机飞行过程需要和障碍物保持安全距离, 设置最低飞行高度为 5 m。

(2) 有效路径集不能在静态障碍物的范围内, 见式(1), 无人机在飞行过程中会实时测量其和障碍物的距离, 进行实时避障。

(3) 无人机搭载相控阵雷达和双目视觉系统, 在扫描完周围地图环境信息以后, 可以在地图上设定飞机起飞和降落的坐标, 并且无人机搭载的有源相控阵雷达, 可以侦测和追踪动态障碍物的飞行轨迹和速度, 再通过本文算法计算其与动态障碍物的距离, 执行实时避障策略, 规划出最短路径。

(4) 本文动态障碍物采用变速或匀速运动, 无人机可以选择不同路径进行避障, 无人机飞行速度为 3~10 m/s, 有效避障速度 ≤ 10 m/s。

(5) 空中动态障碍物的形状不一, 本文以动态障碍物的中心为原点, 将动态障碍物进行膨胀, 膨胀为一个球状, 设置无人机和动态障碍物的最小安全距离为 2.5 m, 如图 2 所示。在仿真软件中, 本文通过坐标点设置移动障碍物的运动路径, 动态障碍物可以做直线运动、直线往返运动、不规则路径运动。

1.3 避障策略和判定

无人机搭载的有源相控阵雷达, 能够实时跟踪动态障碍物的轨迹, 雷达每秒扫描 10 次, 每次都会更新动态障碍物的坐标位置。因此, 利用雷达每次扫描得到的数据, 对二者距离进行实时测量, 判定动态障碍物与无人机是否会发生碰撞。

雷达每秒平均扫描 10 次, 因此, 对单位时间 1 进行离散化处理, 平均分成 10 个时间点, 每个时间点都会进行扫描和判定无人机与动态障碍物之间的距离是否小于最小安全距离, 若小于最小安全距离则不能经过, 无人机会选择改变路线。

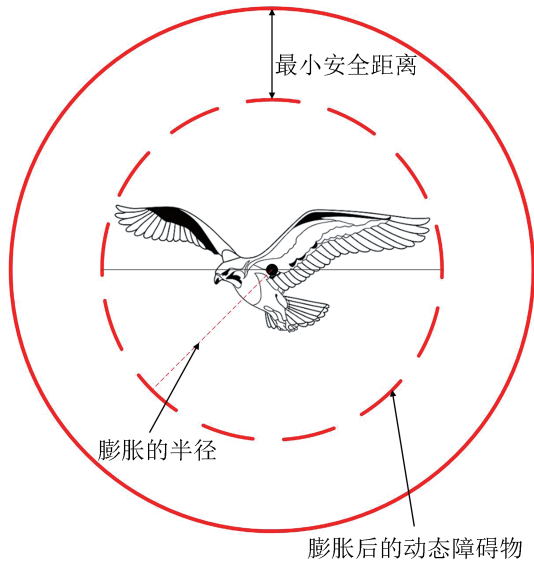


图2 动态障碍物
Fig. 2 Dynamic obstacle

无人机经过某个位置坐标记为 $P(x(t_i), y(t_i), z(t_i))$, 简记为 $P(t_i)$, 则在单位时间内的移动坐标集合为

$$R = \{P(t_1), P(t_2), \dots, P(t_{10})\} \quad (1)$$

动态障碍物经过某个位置时, 中心点的坐标记为 $M(x(t_i), y(t_i), z(t_i))$, 简记为 $M(t_i)$, 则在单位时间内移动的坐标集合为

$$\prod = \{M(t_1), M(t_2), \dots, M(t_{10})\} \quad (2)$$

任意时间点无人机与动态障碍物中心之间距离为

$$d(P(t_i), M(t_i)) = \sqrt{(p_x - m_x)^2 + (p_y - m_y)^2 + (p_z - m_z)^2} \quad (3)$$

设 r_{safe} 为安全距离; r_p 为障碍物膨胀半径。对于任意时间点 $d(P(t), M(t)) \geq r_{\text{safe}} + r_p$, 则无人机与障碍物之间不会碰撞, 反之, 会发生碰撞, 无人机不会停下等待, 接下来规划的路线会绕过动态障碍物。动态障碍物 t 时刻所在区域和静态障碍物区域均为禁行区域。

1.4 3次 B-spline 曲线路径处理

栅格法的路径规划存在一个缺点, 就是飞行轨迹的转角角度较大, 不够平滑, 不平滑的飞行

轨迹会导致无人机飞行姿态不稳定, 且增加无人机的能耗和零部件的磨损。因此, 本文采用伯恩斯坦多项式演变而来的3次 B-spline 曲线, 对飞行轨迹转角处进行平滑处理。

B-spline 曲线可以把栅格当作节点, 并根据节点生成平滑且连续的曲线。K次 B-spline 表示为

$$O_u = \sum_{i=0}^n O_i B_{i,k}(u), i = 1, 2, \dots, n \quad (4)$$

式中: O_i 为曲线的控制点; $B_{i,k}(u)$ 为 k 阶 B 样条在节点 i 上的基函数; k 为阶数, 通常 k 越大, 平滑度越高, 计算复杂度越大, 为平滑复杂度和平滑度, 本文选取 $k=3$, 得到3次 B-spline 曲线, 由4个节点构成; $u = [u_0, u_1, \dots, u_{n+k+1}]$ 为节点向量。

$b_0 \sim b_3$ 分别表示每个节点对应的曲线基函数, 因此 B-spline 曲线表示为

$$\begin{cases} b_0 = \frac{1}{6}(-u^3 + 3u^2 - 3u + 1) \\ b_1 = \frac{1}{6}(3u^3 - 6u^2 + 4) \\ b_2 = \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1) \\ b_3 = \frac{1}{6}u^3 \end{cases} \quad (5)$$

2 改进哈里斯鹰优化算法

2.1 哈里斯鹰优化算法

哈里斯鹰优化算法, 模拟了哈里斯鹰突袭野兔的行为, 当逃离能量 $|E| \geq 1$ 时该算法执行探索行为, 通过 $|E| \geq 0.5$ 判断处于软包围还是硬包围, 并设置一个随机数 $r \geq 0.5$ 判断是否执行俯冲行为。逃离能量变化为

$$E = 2E_0 \left(1 - \frac{t}{t_{\text{max}}}\right) \quad (6)$$

式中: E_0 为逃离能量在 $(-1, 1)$ 的随机初始值; t 为当前迭代次数; t_{max} 为最大迭代次数。

2.1.1 探索行为

当逃离能量 $|E| \geq 1$ 时, 算法进行随机探索, 其位置更新公式为

$$X(t+1) = \begin{cases} X_k(t) - r_1 |X_k(t) - 2r_2 X(t)| \\ (X_r(t) - X_m(t)) - r_3 (b_1 - r_4 (b_u - b_l)) \end{cases} \quad (7)$$

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (8)$$

式中: X_k 为种群中随机个体位置; X_r 为哈里斯鹰最佳值个体位置, 即野兔位置; X_m 为种群平均位置; N 为种群规模; $r_1 \sim r_4$ 为4个独立服从正态分布 $[0,1]$ 的随机数; b_u 和 b_l 为所求问题的上下界。

2.1.2 开发行为

当逃离能量 $|E| < 1$ 时, 算法进入局部搜索的开发阶段, 根据逃离能量值和野兔是否逃离, 选择4种不同策略进行更新位置。

(1) 当逃离能量值 $|E| \geq 0.5$ 且 $r \geq 0.5$ 时, 算法进行软包围, 其位置更新公式为

$$X(t+1) = \Delta X(t) - E |JX(t) - X(t)| \quad (9)$$

$$\Delta X(t) = X_r - X(t) \quad (10)$$

式中: J 为算法的步长, $J \in [0,2]$; $\Delta X(t)$ 为最佳位置与当前位置之间的距离。

(2) 当逃离能量值 $|E| \geq 0.5$ 且 $r < 0.5$ 时, 算法进行硬包围, 野兔没有足够的能量逃离, 直接捕获, 其位置更新公式为

$$X(t+1) = X_r(t) - E |\Delta X(t)| \quad (11)$$

(3) 当逃离能量值 $|E| < 0.5$ 且 $r \geq 0.5$ 时, 野兔能量不足, 哈里斯鹰采取更为智能的软包围, 具体式为

$$Y = X_r(t) - E |JX_r(t) - X(t)| \quad (12)$$

$$Z = Y + S \times Levy(D) \quad (13)$$

$$Levy(D) = \frac{u\sigma}{100 \times |v|^{\frac{1}{\beta}}}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{1/\beta} \quad (14)$$

$$X(t+1) = \begin{cases} Y, Y < X(t) \\ Z, Z < X(t) \end{cases} \quad (15)$$

式中: D 为所求问题的维度; S 为 D 维上一个随机向量; u 、 v 为 $(0,1)$ 的随机数; β 的值为 1.5。

(4) 当逃离能量值 $|E| < 0.5$ 且 $r < 0.5$ 时, 野兔能量不足, 但有机率逃离, 哈里斯鹰采取渐近的软包围, 逐步缩短与野兔之间的距离, 具体公式为

$$Y = X_r(t) - E |JX_r(t) - X_m(t)| \quad (16)$$

$$Z = Y + S \times Levy(D) \quad (17)$$

2.2 改进哈里斯鹰优化算法

2.2.1 基于自适应的混沌和核心种群动态划分策略

哈里斯鹰优化算法在前期和后期运算当中, 探索行为和开发行为并不是先后发生, 也不是等概率发生。根据式(6), 当 $t=1$ 也就是第一次迭代时, 探索概率约为 50%; 在 $t=t_{max}/2$ 时, $E=E_0$, 探索概率为 0, 且之后算法也不会再进行探索行为。因此, 在 $t < t_{max}/2$ 时, 算法的探索概率随迭代次数增加而线性递减, 在 $t=t_{max}/2$ 时递减为 0。所以哈里斯鹰算法在 $t < t_{max}/2$ 时总体探索概率仅为 25% (50% 探索概率 \times 前 50% 迭代次数会探索), 探索次数占总迭代次数的 12.5%。哈里斯鹰算法在迭代的后期缺乏探索行为, 会导致算法陷入局部最优值。

因此, 本文提出一种自适应混沌和核心种群动态划分策略。将混沌种群的选取和智能优化算法的收敛优化策略相融合, 可以使算法在全局探索和局部搜索之间取得平衡, 提高算法的搜索性能和多样性, 在哈里斯鹰算法每次行为完成以后且迭代次数超过 $t_{max}/2$ 时进行该操作, 这样能弥补哈里斯鹰算法在后期缺乏探索行为的缺点, 能够将部分种群释放出去, 在远离最优值的位置进行全局随机探索, 同时保证在最优值附近充分搜索。

设置一种自适应准线 *baseline* 作为划分阈值, 之后的每次迭代结束后都会将种群重新划分为两种, 第一种是以混沌规则在解空间进行全局探索; 第二种是遵循智能优化算法收敛规则在领先混沌

种群 X_t 的局部范围内进行逼近搜索，流程如下：

- (1) 计算 $baseline = \frac{1}{T} \sum_{i=1}^T fit$ ， T 为总迭代次数；
- (2) 若当前解 X 对应的 $fit < baseline$ ，则将 X 划分为核心种群，称为 $core_swarm$ ；
- (3) 若当前解 X 对应的 $fit \geq baseline$ ，则将 X 划分为混沌离散种群，称为 $chaos_swarm$ ；
- (4) 设 t 为当前迭代次数， t_{max} 为最大迭代次数，计算参数 $order = size(chaos_swarm) \cdot (t/t_{max})^2$ ， $size$ 为种群规模，称 $order$ 为划分变量；将混沌离散种群从第1个到第 $order$ 个划分出到核心种群 $core_swarm$ 中，因为混沌本身具有随机性，所以无需增加随机选取的过程。

如此完成一次划分， $baseline$ 会随着迭代次数的变化自适应收敛于 fit ；此外，混沌中种群并不会永远处于混沌当中，在领域周围进行搜索，而是随着迭代次数的增加而减少，会逐渐回归到核心种群，最后只剩极少数还远离核心种群进行随机搜索，且在该过程中游离的混沌种群也会随基线作微小的振荡，反复释放少量的群粒，在混沌操作以后重新进行领域逼近搜索。

本文选择的混沌系统为Tent映射：

$$\begin{cases} x_{t+1} = x_t / \varphi, & 0 < x_t < \varphi \\ x_{t+1} = (1 - x_t) / (1 - \varphi), & \varphi < x_t < 1 \end{cases} \quad (18)$$

当 $\varphi \in (0, 1)$ 且 $x \in [0, 1]$ 时，系统(18)处于混沌状态。完成混沌种群初始化构造后，将所有的群粒以矩阵运算方式输入求解最优适应度 fit^* 及其对应的 X_t ，该个体称为领先混沌群粒。该划分策略相较于其他论文使用的混沌种群初始化能够保证算法在搜索过程中的多样性。

2.2.2 融合黄金正弦策略

根据式(12)~(15)，哈里斯鹰算法采用了一种贪婪策略，在软包围冲刺阶段如果该策略有效，即找到更优解，则更新位置；若该策略失败，没有更优解，则采取式(13)策略，进行莱维飞行扰动，若莱维飞行随机游走失败，没有找到更优解，

则退回原解。为了测试莱维飞行游走的效率，采用标准测试函数进行测试，发现加入莱维飞行游走以后，算法收敛精度并没有明显提高，此外，根据文献[15]，软包围策略失败后，莱维飞行的有效率仅为0.03%。因此，本文融合黄金正弦策略，替换莱维飞行游走，提高算法性能。

黄金正弦策略能够快速遍历最优解附近单位圆的所有点，相比莱维随机游走，有更强的遍历性。在搜索过程中引入黄金分割系数，确保扫描范围控制在最优解的周围领域，加快算法收敛速度。因此，式(13)和(17)的更新公式为

$$\mathbf{Z} = \mathbf{Y} \times \sin R_2 + \mathbf{S} \times \sin R_1 \times |x_1 \cdot \mathbf{X}_t - x_2 \cdot \mathbf{X}(t)| \quad (19)$$

式中： R_1 为 $0 \sim 2\pi$ 之间的随机数； R_2 为 $0 \sim \pi$ 的随机数， R_1 和 R_2 决定了算法在下次迭代时的搜索方向和移动距离； x_1 和 x_2 为黄金分割系数。

$$x_1 = -\pi + (1 - \delta) \cdot 2\pi \quad (20)$$

$$x_2 = -\pi + \delta \cdot 2\pi \quad (21)$$

$$\delta = 0.5(\sqrt{5} - 1) \quad (22)$$

2.2.3 融合自适应正态云最优解扰动策略

哈里斯鹰算法后期易陷入局部极值，为了增强算法后期跳出局部最优解的能力，本文融合了一种自适应正态云最优解扰动策略，越到算法后期，受扰动影响就越大。正态云模型由云滴论域空间期望 Ex 、云团不确定性程度熵 En 和熵不确定性程度超熵 He 三个参数进行描述，正态云是服从正态分布云滴的一种算法，每一次运行都会生产一个云滴，直到生成期望数量的云滴，其生成过程为

$$\mathbf{X}[x_1, x_2, \dots, x_n] = Gnc(Ex, En, He, Nd) \quad (23)$$

$$G_{best} = \mathbf{X}_t \cdot [\lambda Gnc(Ex, En, He, Nd)] \quad (24)$$

$$\lambda = t^2 / T^2 \quad (25)$$

式中： Nd 为生成的云滴数量； G_{best} 为正态云扰动后的解； λ 随着迭代次数增大而增大，到算法后期，最优解受正态云扰动影响也越大。正态云分布如图3所示。

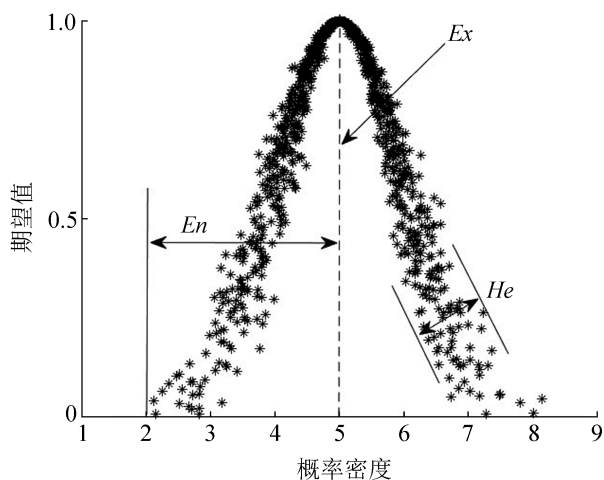


图3 正态云分布图
Fig. 3 Normal cloud distribution map

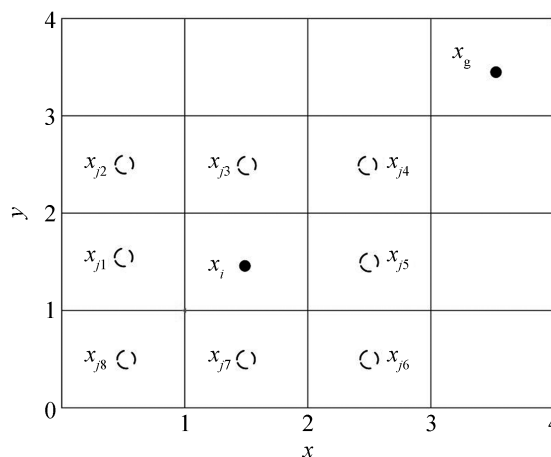


图4 二维节点选择
Fig. 4 Two-dimensional node selection

2.2.4 设置估值函数帮助算法进行节点筛选

在实验中发现, 由于三维环境下栅格数量众多, 算法规划出的路径高度差大、转角多、规划时间长, 影响无人机飞行, 因此本文设置估值函数帮助算法快速进行栅格节点筛选。

(1) 设置距离估值函数

为方便表述, 用二维栅格图进行展示, 如图4所示。\$x_i\$为当前点, \$x_{jn}\$为下一可行节点, \$x_g\$为目标节点。假设当前点周围不存在障碍物, 都为可行区域, 则\$n=1,2,\dots,8\$。

设置下一节点为\$x_{jn}\$, 则距离估值函数为

$$h(d_n) = \frac{(ds - \min ds)}{(\max ds - \min ds) + eps} \quad (26)$$

$$ds = d_{ijn} + d_{jg}$$

式中: \$d_{ijn}\$为当前节点到下一可行节点\$n\$点的距离, \$d_{jg}\$为下一可行节点\$n\$点到目标节点的距离; \$\min ds\$为可行节点到目标节点和当前节点距离之和最短的距离, 图4中当\$x_{j4}\$为下一个节点时, 距离之和最短; \$\max ds\$为可行节点到目标节点和当前节点距离之和最长的距离, 图4中当\$x_{j8}\$为下一个节点时, 距离之和最长; \$eps\$为很小的数, 避免分母为0。

(2) 设置角度估值函数

判断\$x_{jn}\$和\$x_{goal}\$的连线与\$x_i\$和\$x_{jn}\$连线的延长线之间的夹角\$\theta\$并赋值, 如图5所示。

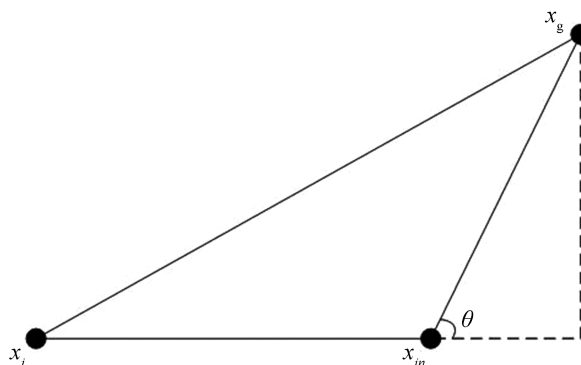


图5 夹角\$\theta\$
Fig. 5 Included angle \$\theta\$

根据三角函数余弦定理, \$\theta\$的计算公式为

$$\theta = \arccos\left(\frac{(d(x_g, x_i))^2 + (d(x_{jn}, x_i))^2 - (d(x_{jn}, x_g))^2}{2d(x_g, x_i) \cdot d(x_{jn}, x_i)}\right) + \arccos\left(\frac{(d(x_g, x_{jn}))^2 + (d(x_g, x_i))^2 - (d(x_{jn}, x_i))^2}{2d(x_g, x_{jn}) \cdot d(x_g, x_i)}\right) \quad (27)$$

式中: \$d(x_1, x_2)\$为\$x_1\$、\$x_2\$之间的距离。

角度估值函数为

$$h(a_n) = \begin{cases} 0, & \theta > 60^\circ \\ 1, & \theta \leq 60^\circ \end{cases} \quad (28)$$

选择下一节点\$x_{jn}\$, 总估值函数为

$$S_n = a \cdot h(d_n) + b \cdot h(a_n) \quad (29)$$

式中: \$a\$、\$b\$分别为权重系数, 取值为\$[0.1, 1]\$。\$a\$、\$b\$之间非线性关系, 路径代价\$h(d_n)\$和\$h(a_n)\$角度代

价对路径搜索的长度影响不同。合理的选择权重 a 、 b ，有利于路径规划的合理性。通过仿真实验发现， $a=0.3$ 且 $b=0.8$ 时，规划出的路径最短，因此下文实验就使用此参数。

通过设置一个简单地图实验，验证估值函数

的作用。未加入估值函数的HHO算法规划的路径为红色，加入估值函数的HHO算法规划出的路径为蓝色。从图6明显看出，没有估值函数的HHO算法即使在简单地图上规划出的路径高度落差大、转角多，相比之下，就不适合无人机的飞行。

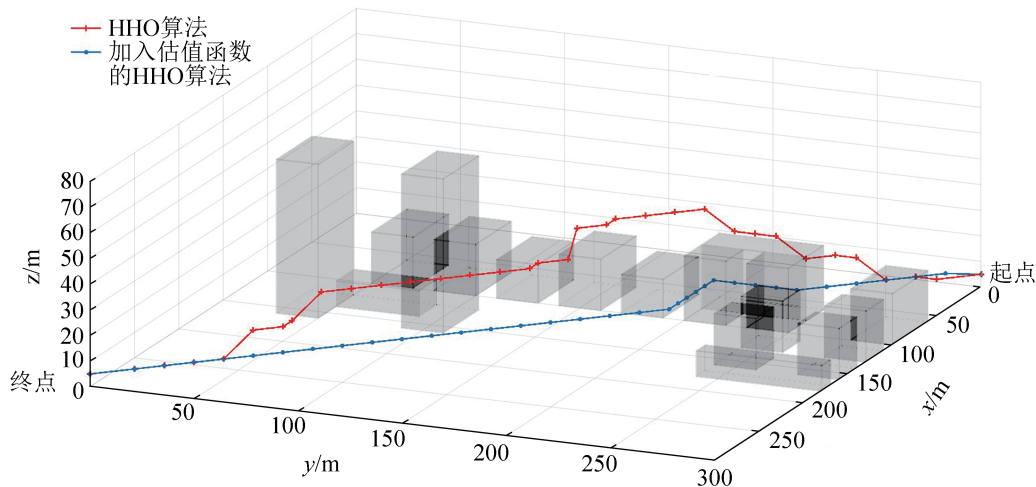


图6 估值函数验证

Fig. 6 Valuation function verification

2.3 算法流程

改进哈里斯鹰优化算法 (improved HHO, IHHO)具体伪代码步骤如下。

算法: IHHO

1: Generate of N population and maximum number of iterations T

2: The location of rabbit and its fitness value.

Initialize the random population $X_i (i = 1, 2, \dots, N)$

3: While stopping condition is not met do

4: Calculate the fitness values of hawks

5: Set X as the location of rabbit best location

6: for each hawk (X_i) do

7: Update the initial energy E_0 and step size J

8: Update the E using Eq.(6)

9: if $|E| \geq 1$ then

10: Update the location vector

using Eq.(8)

11: if $|E| < 1$ then

12: if $|E| \geq 0.5$ and $r \geq 0.5$ then

13: Update the location vector

using Eq.(9)

14: else if $|E| \geq 0.5$ and $r < 0.5$ then

15: Update the location vector

using Eq.(11)

16: else if $|E| < 0.5$ and $r \geq 0.5$ then

17: Update the location vector

using Eq.(12) and (19)

18: else if $|E| < 0.5$ and $r < 0.5$ then

19: Update the location vector

using Eq.(16) and (19)

20: end for

21: While the number of iterations satisfies $t > T/2$ do

22: Calculate the fitness values of baseline

23: According to fitness adaptive chaotic population division using Eq.(18)

24: Normal cloud perturbation of the best fitness using Eq.(24)

25: Calculate the fitness value all updated hawks positions

26: $t=t+1$

27: end while

28: end while

3 实验结果及分析

实验环境: 操作系统 Windows 11(64bit), 处理器 AMD Ryzen 7 5800H with Radeon Graphics. 3.20 GHz, 运行内存 16 G, 仿真平台 Matlab 2021a。

实验类别: 为了充分验证 IHHO 的性能以及其在无人机路径规划应用当中的优越性, 设置4组实验: ①数值实验, 选择国际通用测试函数, 验证融合前3个策略的 IHHO 算法的性能是否有明显提升。②静态地图无人机路径规划实验, 设置2张不同大小、不同复杂程度的静态地图进行路径规划实验。③动态地图无人机路径规划实验, 设置2张不同大小、不同复杂程度、动态障碍物数量不同的地图进行实验, 充分验证其性能。④3次 B-spline 曲线路径平滑处理。

对照组设置: 数值实验选取6种算法作为对照组: GA、PSO、GWO、HHO、螺旋离散混沌种群振荡搜索算法(dispersed chaotic swarm oscillation algorithm merged with spira, IDCSSA-S)^[16]和融合多策略的精英反向学习改进 HHO(golden-sine HHO, GHHO)^[17]。路径规划实验选取5种算法作为对照组: GA、PSO、HHO、融合莱维飞行和领域交流机制的改进鲸鱼优化算法(improved WOA, IWOA)^[12]以及融合二次规划和柯西变异的狮群(combining two variations LSO, CTLSSO)算法^[11]。

参数设置: GA 交叉概率 $p_c=0.8$, 变异概率 $p_m=0.1$; PSO 学习因子 $c_1=1.5$, $c_2=1.5$, 惯性权

重0.8; GWO 系数 $A=1$; WOA 算法和 IWOA 的狩猎行为概率 $p_s=0.5$, 螺旋形态常数 $b=1$; CTLSSO 成年狮比例因子 $\beta=0.3$ 。

数据统计: 为保证公平性, 每次实验重复10次, 每次算法运行10轮, 统计其平均值和标准差。

3.1 数值实验

进行国际通用测试函数的数值实验, 使用融合前3个策略的 IHHO 算法, 验证3个策略的有效性(估值函数策略仅使用在路径规划实验当中)。为避免偶然性, 所有算法参数设置一致, 最大迭代次数 $t_{max}=500$, 维度 $D=30$, 种群规模均取 $N=30$ 。表1为部分标准测试函数。其中, $F_1\sim F_4$ 为单峰测试函数, 验证算法局部搜索能力; $F_5\sim F_8$ 为多峰测试函数, 验证算法全局搜索能力; $F_9\sim F_{12}$ 为固定维数多峰测试函数, 验证算法平衡全局搜索和局部搜索的能力。

表1 国际标准测试函数
Table 1 International standard test function

函数	名称	取值范围	维度	最值
F_1	Sphere	[-100,100]	n	0
F_2	Schwefel2.22	[-10,10]	n	0
F_3	Schwefel1.2	[-100,100]	n	0
F_4	Schwefel2.21	[-100,100]	n	0
F_5	Schwefel's 2.26	[-500,500]	n	-12 569
F_6	Rastrigin	[-5.12,5.12]	n	0
F_7	Ackley	[-32,32]	n	0
F_8	Griewank	[-600,600]	n	0
F_9	Foxholes	[-65,65]	2	0.998
F_{10}	Kowalik's	[-5,5]	4	3.075e-04
F_{11}	Six-Hump	[-5,5]	2	-1.031 628
F_{12}	Branin	[-5,5]	2	0.398

由表2和图7测试结果和收敛曲线可知, 对于单峰测试函数 $F_1\sim F_4$, 本文算法能找到最优值, 并且算法在10轮运行中, 均值和标准差最小, 说明改进后的算法具有更好的局部开发能力。

对于多峰测试函数 $F_5\sim F_8$, 本文算法相较于对比算法平均值和标准差最小, 说明 IHHO 算法在

处理更复杂的问题时，有更强跳出局部最优解的能力。

对于固维函数 $F_9 \sim F_{12}$ ，本文算法搜索结果最

接近理论最优值，标准差最小，说明 IHHO 算法相比对照组算法具有更好平衡全局开发和局部探索的能力。

表 2 测试函数结果对比
Table 2 Comparison of test function results

函数		GA	PSO	GWO	HHO	DCSOA-S	GHHO	IHHO
F_1	ave	8.57e+02	2.60e+03	1.11e-31	5.11e-96	1.71e-152	9.08e-186	0
	std	1.84e+02	3.31e+03	5.02e-31	2.79e-95	1.86e-156	0	0
	f_{best}	4.75e+02	2.99e+01	5.81e-32	3.34e-153	—	7.58e-150	0
F_2	ave	1.01e+02	1.51e+02	5.30e-09	2.69e-49	5.21e-108	7.77e-105	0
	std	1.39e+01	2.45e+01	2.80e-09	1.45e-48	6.13e-108	1.11e-104	0
	f_{best}	8.66e+01	1.29e+02	2.07e-09	4.09e-88	—	2.24e-131	0
F_3	ave	4.62e+04	9.18e+03	1.33e-08	2.89e-75	1.34e-85	4.70e-76	0
	std	1.25e+04	8.67e+03	1.97e-08	1.08e-74	4.82e-86	2.57e-75	0
	f_{best}	3.09e+04	3.23e+03	3.09e-09	8.70e+03	—	5.10e-114	0
F_4	ave	5.05e+01	2.31e+01	4.44e-08	4.88e-50	1.07e-94	7.95e-65	0
	std	8.28+00	5.16+00	5.55e-08	1.93e-49	2.08e-95	3.26e-64	0
	f_{best}	4.01e+01	1.71e+01	7.11e-09	4.86e-10	—	2.58e-74	0
F_5	ave	-1.12e+04	-6.94e+03	-3.111e+03	1.26e+04	-1.25e+04	—	-1.25e+04
	std	2.47e+02	1.04e+03	3.05e+02	9.45e-01	0	—	0
	f_{best}	-1.17e+04	-8.05e+03	-3.68e+03	-12 567	—	—	-12 569
F_6	ave	5.09e+01	1.13e+02	0	0	1.39e-06	0	0
	std	1.16e+01	3.39e+01	0	0	1.49e-06	0	0
	f_{best}	3.48e+01	5.36e+01	0	0	—	0	0
F_7	ave	7.34e+00	1.23e+01	1.72e-14	8.88e-16	3.86e-16	8.88e-16	8.88e-16
	std	1.29e+00	1.81e+00	4.49e-15	0	2.53e-16	0	0
	f_{best}	5.07e+00	1.00e+01	1.15e-14	8.88e-16	—	8.88e-16	8.88e-16
F_8	ave	8.36e+00	1.77e+01	0	0	5.74e-03	0	0
	std	3.02e+00	1.01e+01	0	0	5.53e-03	0	0
	f_{best}	3.64e+00	4.86e+00	0	0	—	0	0
F_9	ave	9.98e-01	1.09e+00	8.24e+00	1.39e+00	—	—	9.98e-01
	std	3.01e-04	3.14e-01	3.75e+00	9.91e-01	—	—	6.35e-10
	f_{best}	0.998	0.998	1.04e+00	0.998	—	—	0.998
F_{10}	ave	1.93e-02	3.50e-03	2.70e-03	3.79e-04	3.00e-03	4.11e-04	3.08e-04
	std	1.71e-02	6.40e-03	6.20e-03	2.33e-04	0	1.76e-04	0
	f_{best}	2.20e-03	3.07e-04	4.54e-04	5.96e-04	—	3.07e-04	3.07e-04
F_{11}	ave	-1.02e+00	-1.03e+00	-1.03e+00	-1.03e+00	-1.03e+00	-1.03e+00	1.03e+00
	std	4.20e-03	1.48e-16	1.50e-08	4.88e-08	0	9.63e-09	2.95e-04
	f_{best}	-1.031 6	-1.031 6	-1.031 6	-1.031 6	—	-1.0316	-1.031 6
F_{12}	ave	4.16e-01	3.97e-01	3.97e-01	3.98e-01	—	—	3.97e-01
	std	2.79e-02	0	8.91e-07	6.68e-04	—	—	0
	f_{best}	0.398 5	0.397 9	0.397 9	0.397 9	—	—	0.397 9

在选取的 12 个国际标准测试函数中, 本文算法。相比 GHHO 算法有明显的提升。相较于对比算法, 大部分测试函数寻优过程中, 本文算法平均值和标准差最小, 稳定性最高。总体而言, IHHO 算法在 7 种算法中表现最好。

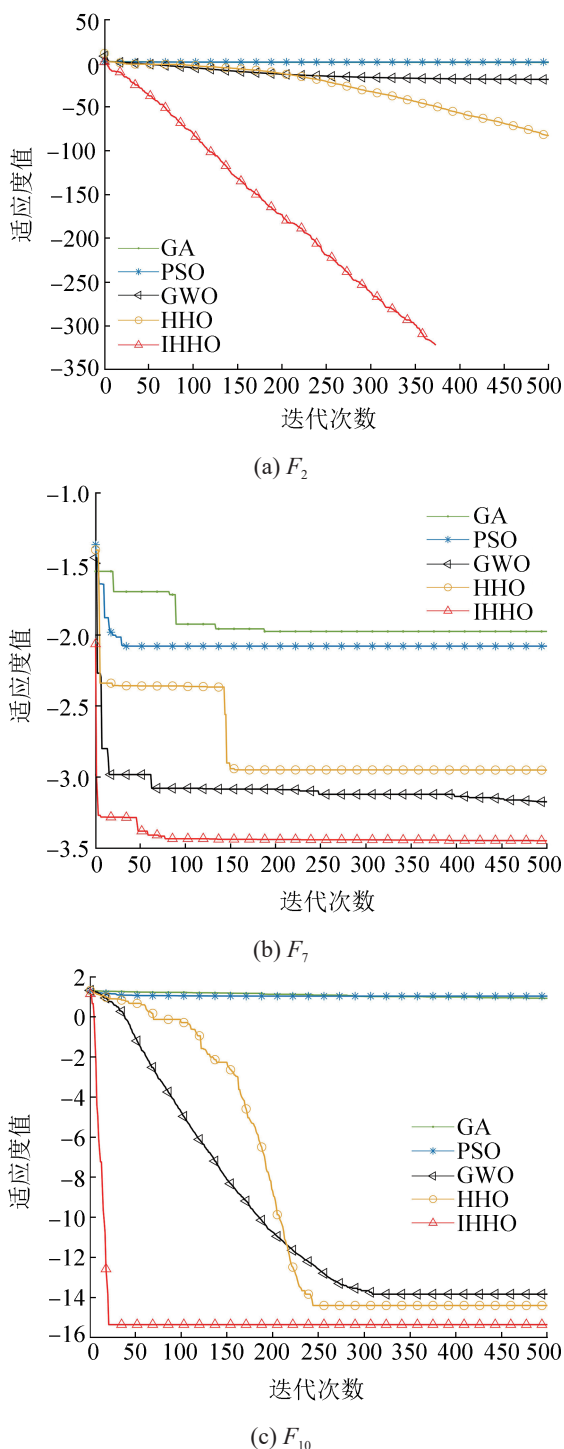


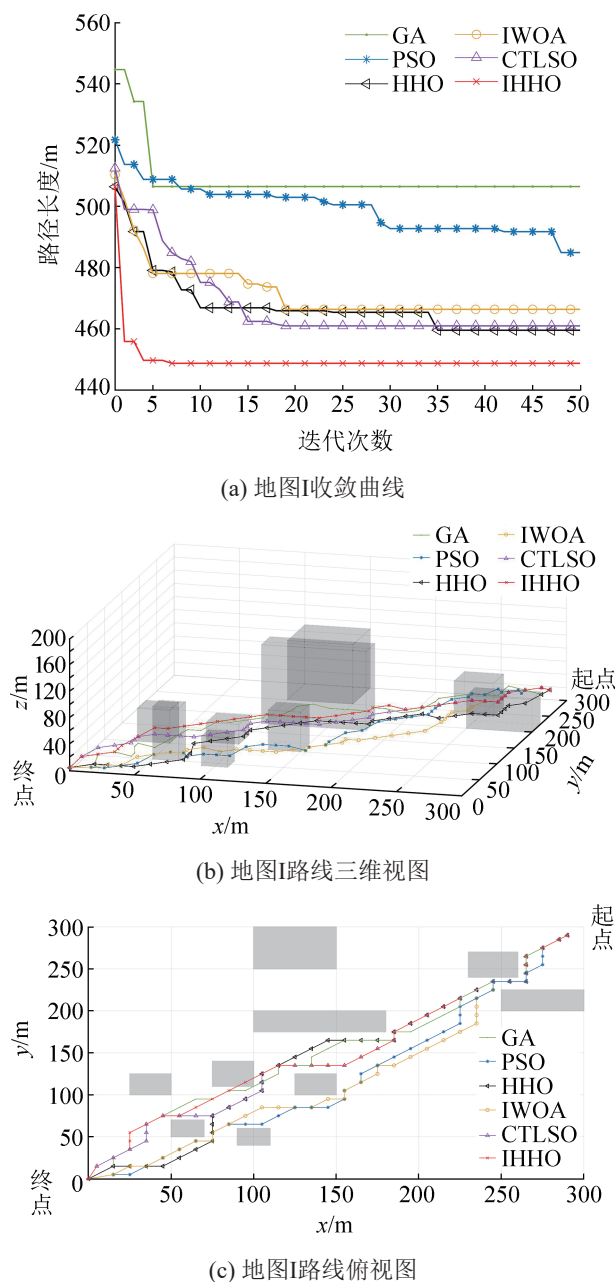
图 7 部分测试函数收敛曲线

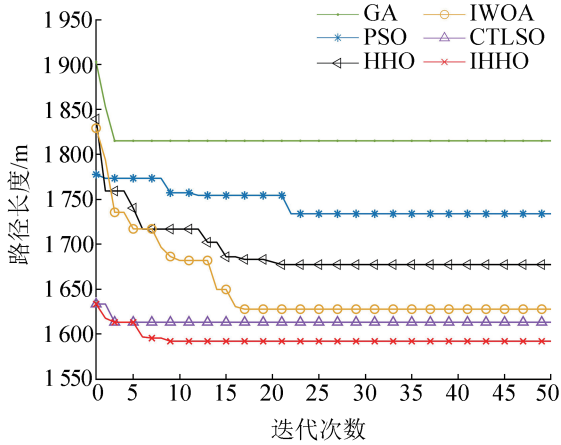
Fig. 7 Test function convergence curves (part)

<http://www.china-simulation.com>

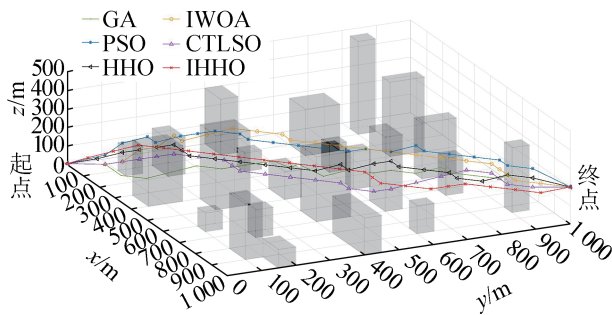
3.2 静态地图路径规划实验

3.1 节数值实验验证了融合前 3 个策略的 HHO 算法在性能上具有一定优势, 但是仅仅是数值实验, 不能充分验证算法性能, 因此, 将算法运用在高纬度的问题当中是必要的。首先进行静态地图路径规划实验, 设置地图 I(300×300×200)和地图 II(1 000×1 000×500)。所有算法参数设置一致, 最大迭代次数 $t_{max} = 50$, 种群规模均取 $N=30$ 。路径规划结果如图 8 所示。

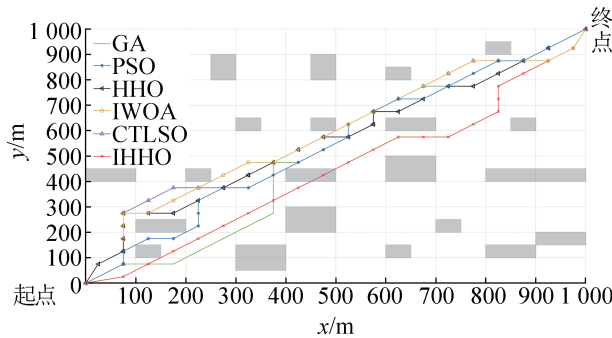




(d) 地图II收敛曲线



(e) 地图II路线三维视图



(f) 地图II路线俯视图

图8 静态地图实验结果

Fig. 8 Experimental results of static map

为了更好地对比改进前后的效果，定义优化率为

$$g_n = \frac{|\mu_m - \mu_0|}{\mu_0} = \frac{\Delta\mu}{\mu_0} \quad (30)$$

$$g_{\text{avg}} = \frac{\sum_{n=1}^n g_n}{n} \quad (31)$$

式中： μ_m 为改进后算法规划路径的长度、算法运

行时间和路径转角数量； μ_0 为改进前算法的路径长度、算法运行时间和路径转角数量； g_n 为算法 n 的优化率； g_{avg} 为平均优化率。该等式能够直接反映算法的优化程度。

从表3实验结果得出，IHHO算法在静态路径规划实验中，平均路径长度分别优化了5.52%和5.81%，提升明显，在路径转角数量上，都减少了30%以上，算法规划出的路径更适合无人机的飞行；IHHO运行时间最短，说明改进策略没有增加算法的时间复杂度，节点筛选策略能够有效减少路径转角数量。

表3 静态地图实验数据

Table 3 Experimental data of static map

环境	算法	平均路径长度/m	转角数	运行时间/s
地图I	GA	506.485	29	14.430
	PSO	484.931	23	15.036
	HHO	459.572	22	14.337
	IWOA	466.393	19	13.822
	CTLSO	460.992	14	13.112
	IHHO	448.778	13	9.538
IHHO 平均优化率/%		5.52	35.65	32.43
地图II	GA	1 815.114	30	19.022
	PSO	1 733.927	31	18.741
	HHO	1 677.342	20	17.698
	IWOA	1 627.704	20	16.776
	CTLSO	1 613.145	14	15.830
	IHHO	1 592.054	14	11.753
IHHO 平均优化率/%		5.81	33.63	32.95

3.3 动态地图路径规划实验

动态障碍物会对算法性能产生影响，性能不足的算法就会陷入局部最优甚至无法规划出路径。本节设置2张不同大小的地图，并加入不同数量的动态障碍物进行实验。

动态地图I：在300×300×200地图中加入一个动态障碍物 M_{ob1} ，干扰无人机的飞行，障碍物在无人机飞行路径上随机生成，障碍物的加速度为 0.1 m/s^2 ，速度由 8 m/s 递增至 10 m/s ，无人机的速度为 10 m/s 。图9为动态地图I实验的结果。

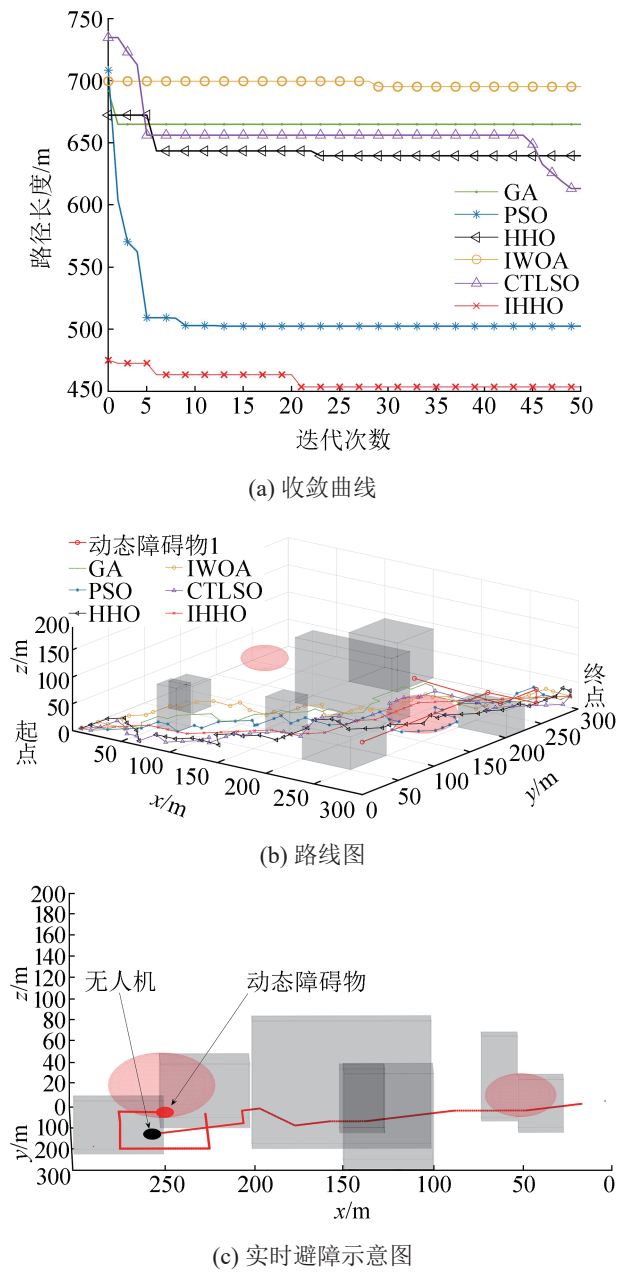


图9 动态地图I实验结果
Fig. 9 Experimental results of 300×300×200 dynamic map I

动态地图II: 在 1 000×1 000×500 的地图进行模拟仿真实验, 地图设置了 22 栋建筑, 2 处雷达干扰区域, 加入 3 个动态障碍物, 分别是 M_{ob1} 、 M_{ob2} 和 M_{ob3} 。3 个动态障碍物会在算法规划的路径上随机生成。其中, M_{ob1} 的初始速度为 5 m/s, 加速度为 0.2 m/s², 最高速度为 8 m/s; M_{ob2} 的初始速度为 8 m/s, 加速度为 0.2 m/s², 最高速度为 10 m/s; M_{ob3} 的初始速度为 10 m/s, 加速度为 -0.5 m/s², 最

低速度为 5 m/s; 无人机的飞行速度为 10 m/s。图 10 为动态地图II实验的结果。

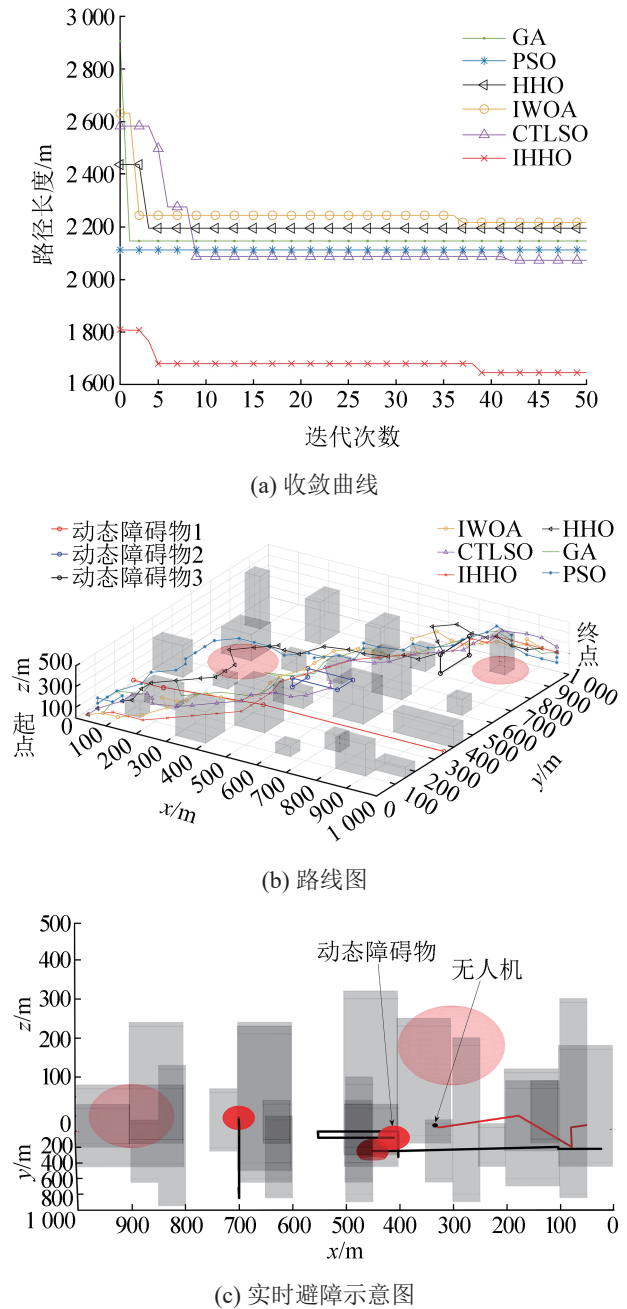


图10 动态地图II实验结果
Fig. 10 Experimental results of dynamic map II

从图9(a)的收敛曲线可以明显看出: IHHO性能最优, 收敛最快。IHHO除了对算法本身的改进, 还针对路径规划加入了节点筛选策略, 帮助算法选择更短路径。从图10(a)的收敛曲线可以看出, 在动态障碍物更大、地图更复杂的实验中,

IHHO性能提升十分明显。图10(c)中,无人机向上飞行躲避动态障碍物。

从表4实验数据得出, IHHO算法运行时间最短, 相比其他算法运行时间减少了35.40%和43.33%。在路径寻优方面, 随着计算量增大, IHHO算法的优势更加显著, 平均路径缩短了26.27%和23.34%。在路径转角数量方面, 加入估值函数的IHHO规划出的路径转角最少, 转角数量减少比率为68%以上, 路径更适合无人机的飞行。相较于静态地图规划结果, IHHO在动态地图中性能提升更加明显, 主要是因为计算量增大, 普通算法的性能不足。

表4 动态地图实验结果
Table 4 Experimental results of dynamic map

环境	算法	平均路径长度/m	转角数	运行时间/s
地图I	GA	664.796	37	28.632
	PSO	502.411	15	23.889
	HHO	639.471	31	47.592
	IWOA	695.042	19	50.993
	CTLSSO	613.115	27	46.634
	IHHO	453.553	6	20.393
IHHO平均优化率/%		26.27	74.13	43.33
地图II	GA	2 146.167	25	31.565
	PSO	2 112.059	24	28.076
	HHO	2 194.731	21	60.631
	IWOA	2 216.877	20	56.057
	CTLSSO	2 072.962	23	62.250
	IHHO	1 646.203	7	27.442
IHHO平均优化率/%		23.34	68.81	35.40

对比表3~4的实验结果, 在相同地图的情况下, 动态环境中IHHO算法规划出的最短路径长度和算法运行时间明显增加, 在300×300×200地图中, IHHO算法在动态环境规划出的最短路径长度和运行时间比静态环境增加了1.103%和53.22%; 在1 000×1 000×500地图中, IHHO算法在动态环境中规划的最短路径长度和运行时间增加了3.29%和57.17%。

动态环境中算法规划的路径长度增加主要是因为动态障碍物的影响, 无人机绕开障碍物, 增

加了飞行距离; 运行时间增加主要是因为算法除了需要规划最短路径, 还需要每秒计算10次动态障碍物的坐标以及动态障碍物与自身的相对位置进行动态实时避障, 计算量增加导致了运行时间增加。

3.4 3次B-spline曲线路径平滑处理

栅格法路径规划存在转角不够平滑的缺点, 不符合无人机飞行约束, 本文使用3次B-spline曲线, 对路径转角处进行平滑处理, 避免无人机在飞行过程中的不良飞行轨迹。如图11所示, 对平滑处进行放大, 并和未平滑处理的路径进行对比。

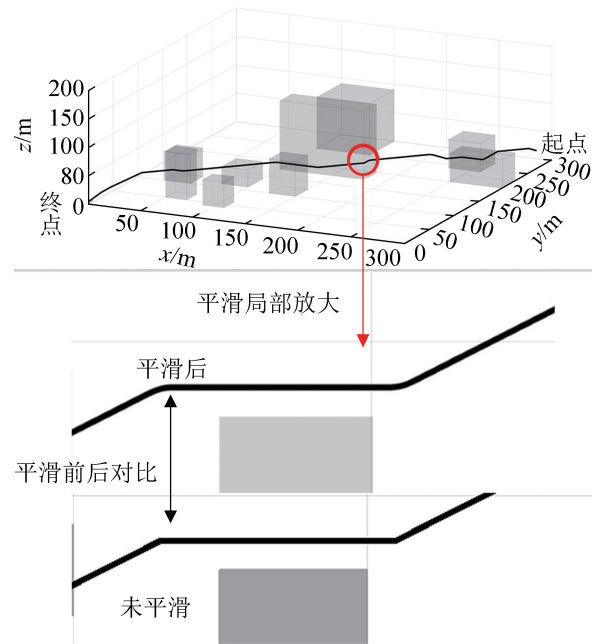


图11 路径平滑处理
Fig. 11 Path smoothing

4 结论

自动避障路径规划算法能够帮助无人机实现自主导航、自主避障, 更好地完成飞行任务。本文针对无人机应用场景, 提出一种改进哈里斯鹰算法, 融合3种改进策略, 提高算法的计算性能, 能够更好地完成路径规划任务, 实现无人机实时避障。采用B-spline曲线圆滑处理方法解决路径转

角问题, 避免急停旋转, 使路径更适合无人机飞行。实验验证表明: 在不同大小、不同复杂程度和不同移动障碍物数量的场景下, 本文算法规划出的路径最短。

后续将针对算法运行时间缩短、多无人机避障等方面开展进一步研究, 完善算法使其更符合实际应用。

参考文献:

- [1] Hart P E, Nilsson N J, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. *IEEE Transactions on Systems Science and Cybernetics*, 1968, 4(2): 100-107.
- [2] 程传奇, 郝向阳, 李建胜, 等. 融合改进A*算法和动态窗口法的全局动态路径规划[J]. *西安交通大学学报*, 2017, 51(11): 137-143.
Cheng Chuanqi, Hao Xiangyang, Li Jiansheng, et al. Global Dynamic Path Planning Based on Fusion of Improved A* Algorithm and Dynamic Window Approach [J]. *Journal of Xi'an Jiaotong University*, 2017, 51(11): 137-143.
- [3] Brock O, Khatib O, Viji S. Task-consistent Obstacle Avoidance and Motion Behavior for Mobile Manipulation [C]//*Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. Piscataway, NJ, USA: IEEE, 2002: 388-393.
- [4] 黄志锋, 刘媛华, 张聪. 多策略融合的改进狮群算法及其工程优化[J]. *小型微型计算机系统*, 2024, 45(4): 838-844.
Huang Zhifeng, Liu Yuanhua, Zhang Cong. Multi-strategy Fusion Improved Lion Swarm Optimization Algorithm and Its Application of Project Optimization[J]. *Journal of Chinese Computer Systems*, 2024, 45(4): 838-844.
- [5] Liu Kun, Xü Cheng, Huang Daqing, et al. UAV Path Planning Based on Improved Whale optimization Algorithm[C]//*2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. Piscataway, NJ, USA: IEEE, 2021: 569-573.
- [6] Ma Guoyuan, Yue Xiaofeng. An Improved Whale Optimization Algorithm Based on Multilevel Threshold Image Segmentation Using the Otsu Method[J]. *Engineering Applications of Artificial Intelligence*, 2022, 113: 104960.
- [7] Heidari A A, Mirjalili S, Hossam Faris, et al. Harris Hawks Optimization: Algorithm and Applications[J]. *Future Generation Computer Systems*, 2019, 97: 849-872.
- [8] 马云红, 张恒, 齐乐融, 等. 基于改进A*算法的三维无人机路径规划[J]. *电光与控制*, 2019, 26(10): 22-25.
Ma Yunhong, Zhang Heng, Qi Lerong, et al. A 3D UAV Path Planning Method Based on Improved A* Algorithm [J]. *Electronics Optics & Control*, 2019, 26(10): 22-25.
- [9] 湛海云, 陈华胄, 刘强. 基于改进人工势场法的多无人机三维编队路径规划[J]. *系统仿真学报*, 2020, 32(3): 414-420.
Chen Haiyun, Chen Huazhou, Liu Qiang. Multi-UAV 3D Formation Path Planning Based on Improved Artificial Potential Field[J]. *Journal of System Simulation*, 2020, 32(3): 414-420.
- [10] 黄志锋, 刘媛华. 基于四阶贝塞尔曲线和改进狮群优化算法求解路径规划问题[J]. *信息与控制*, 2023, 52(2): 176-189.
Huang Zhifeng, Liu Yuanhua. Solving Path Planning Problem Based on Fourth-order Bezier Curve and Improved Lion Swarm Optimization Algorithm[J]. *Information and Control*, 2023, 52(2): 176-189.
- [11] 黄志锋, 刘媛华. 基于改进狮群算法的城市无人机低空路径规划[J]. *信息与控制*, 2023, 52(6): 747-757, 772.
Huang Zhifeng, Liu Yuanhua. Low Altitude Path Planning of Urban UAV Based on Improved Lion Swarm Optimization[J]. *Information and Control*, 2023, 52(6): 747-757, 772.
- [12] 郭启程, 杜晓玉, 张延宇, 等. 基于改进鲸鱼算法的无人机三维路径规划[J]. *计算机科学*, 2021, 48(12): 304-311.
Guo Qicheng, Du Xiaoyu, Zhang Yanyu, et al. Three-dimensional Path Planning of UAV Based on Improved Whale Optimization Algorithm[J]. *Computer Science*, 2021, 48(12): 304-311.
- [13] 高建瓴, 喻明毫. 混合差分进化与二次插值改进哈里斯鹰算法[J]. *小型微型计算机系统*, 2023, 44(7): 1352-1359.
Gao Jianling, Yu Minghao. Hybrid Differential Evolution and Quadratic Interpolation Improved Harris Hawks Optimization Algorithm[J]. *Journal of Chinese Computer Systems*, 2023, 44(7): 1352-1359.
- [14] 苏佳, 杨志华, 刘彦明. 基于改进哈里斯鹰算法的系统能效优化[J]. *华中科技大学学报(自然科学版)*, 2024, 52(3): 58-64.
Su Jia, Yang Zhihua, Liu Yanming. System Energy Efficiency Optimization Based on Improved Harris Hawk Algorithm[J]. *Journal of Huazhong University of Science and Technology(Natural Science Edition)*, 2024, 52(3): 58-64.
- [15] 刘小龙, 梁彤纓. 基于方形邻域和随机数组的哈里斯鹰优化算法[J]. *控制与决策*, 2022, 37(10): 2467-2476.

- Liu Xiaolong, Liang Tongying. Harris Hawk Optimization Algorithm Based on Square Neighborhood and Random Array[J]. Control and Decision, 2022, 37(10): 2467-2476.
- [16] 林之博, 刘媛华. 融合螺旋策略的离散混沌群粒振荡搜索算法[J]. 计算机应用研究, 2021, 38(10): 3060-3066, 3071.
- Lin Zhibo, Liu Yuanhua. Dispersed Chaotic Swarm Oscillation Algorithm Merged with Spiral Strategy[J]. Application Research of Computers, 2021, 38(10): 3060-3066, 3071.
- [17] 郭雨鑫, 刘升, 高文欣, 等. 精英反向学习与黄金正弦优化的HHO算法[J]. 计算机工程与应用, 2022, 58(10): 153-161.
- Guo Yuxin, Liu Sheng, Gao Wenxin, et al. Elite Opposition-based Learning Golden-sine Harris Hawks Optimization[J]. Computer Engineering and Applications, 2022, 58(10): 153-161.