

# 基于POA算法的无人机航迹规划优化方法



贾汉博<sup>1</sup>, 马琳<sup>1</sup>, 张忠旺<sup>2</sup>

1. 哈尔滨工业大学, 黑龙江 哈尔滨 150006

2. 中国航空无线电电子研究所, 上海 200241

**摘要:** 针对航迹规划这一无人机(UAV)自动控制领域的优化问题, 本文提出了一种基于传教优化算法(POA)的UAV航迹规划优化方法。该方法设计了包括航迹长度、地形代价以及飞行高度代价的优化目标函数模型, 在初始化传教士位置时引入旋转坐标系以加速算法收敛速度并详细说明了POA算法和UAV航迹规划结合的具体问题及其处理方法。最终, 与综合改进粒子群算法航迹规划的仿真进行对比, 从航迹长度、收敛速度以及对于不同地形图的适应度三个方面证明了本文所提基于POA算法的UAV航迹规划优化方法的有效性。该研究可以应用于UAV航迹规划领域, 对于提升UAV自主导航能力具有重大意义。

**关键词:** 无人机; 航迹规划; 优化算法; 群智能优化; 传教优化算法

中图分类号: V249

文献标识码: A

DOI: 10.19452/j.issn1007-5453.2022.10.012

随着自动控制以及人工智能领域的蓬勃发展, 无人机(UAV)被广泛应用于各种场景之中, 如灾难营救、未知环境勘探、远程精确打击目标、针对某一环境的覆盖式探索以及集群对抗<sup>[1-3]</sup>。UAV的发展可以极大地提高任务效率并避免不必要损失, 对于各种领域都具有重大意义。

为了让UAV自主地完成复杂任务, 许多学者对UAV航迹规划进行了深入研究。基于群智能优化的航迹规划算法因其扩展性强的特点已成为研究热点。目前常见的群智能优化算法包括蚁群算法(ACO)<sup>[4]</sup>、差分进化算法(DE)<sup>[5]</sup>、果蝇优化算法(FOA)<sup>[6]</sup>和粒子群优化算法(PSO)<sup>[7]</sup>等。其中, 参考文献[8]改进了ACO算法中的信息素分布以及转移概率, 实现了复杂障碍环境下的UAV航迹规划。参考文献[9]将DE算法与逼近策略相结合, 提出了一种混合差分进化算法, 并用于固定翼UAV在复杂三维环境下的航迹规划。参考文献[10]提出了一种基于最优参考点果蝇优化算法, 将两个连续航迹点的中点设置为参考点以提高FOA算法的收敛速度。参考文献[11]将综合改进粒子群算法(CIPSO)应用于UAV航迹规划。参考文献[11]利用混沌Logistic映射来令算法的初始值更加随机, 并提出了一种自适应线性变化策略来调整CIPSO算法中的参

数。仿真结果证明了该算法在收敛速度以及航迹规划结果方面的优势, 但其采用的地形图过于简单, 无法评估在复杂地形下的算法性能。

传教优化算法(POA)是Wei Dong等<sup>[12]</sup>于2020年提出的一种群智能优化算法, 该算法结合个体适应度以及位置之间的关系计算权重, 以维持个体的多样性, 且利用精英策略以及人工免疫算法加速收敛。最终, 参考文献[12]通过CEC'17数据集的测试结果说明了该算法在收敛速度以及精度方面的性能。但将POA算法应用于UAV航迹规划时, POA算法中随机初始化传教士位置的方式并没有结合航迹的方向特性, 且对于三维坐标以及边界条件的处理仍有一些问题需要解决。

因此, 本文提出了基于POA算法的UAV航迹规划优化方法。首先, 本文提出了结合航迹长度、地形代价以及飞行高度代价的目标函数。其次, 在初始化传教士位置时引入旋转坐标系, 通过这种方式使得初始化的航迹结果具有一定的方向性, 极大地缩短了算法收敛时间。此外, 本文详细说明了航迹坐标点以及边界条件在POA中的具体处理方式, 解决了POA与航迹规划相结合的问题。最后, 本文分别从航迹长度、算法收敛速度以及针对不同地形图的适应

收稿日期: 2022-02-20; 退修日期: 2022-04-25; 录用日期: 2022-06-19

基金项目: 国家自然科学基金(61971162, 41861134010); 航空科学基金(2020Z066015002)

引用格式: Jia Hanbo, Ma Lin, Zhang Zhongwang. Optimization method of unmanned aerial vehicle path planning using POA[J]. Aeronautical Science & Technology, 2022, 33(10): 96-102. 贾汉博, 马琳, 张忠旺. 基于POA算法的无人机航迹规划优化方法[J]. 航空科学技术, 2022, 33(10): 96-102.

程度三个方面对比CIPSO航迹规划,证明了基于POA算法的UAV航迹规划的有效性。

## 1 问题建模

### 1.1 地图限制

对于UAV航迹规划这一优化问题而言,需要考虑多方面的因素。首先,UAV要在地图限制的区域飞行。其次,为了令UAV安全飞行,UAV不能和地面发生碰撞并与地面保持一定的安全距离,且飞行高度不能超出最大飞行高度的限制。最后,UAV的飞行距离要尽可能短,以节省燃料并尽快到达目的地。

本文利用三维坐标 $(x,y,z)$ 来表示地形图,其中 $x$ 与 $y$ 分别表示地形图在水平面上的横、纵坐标, $z$ 表示 $(x,y)$ 处的地形高度。通过控制 $z$ 的取值即可生成不同的地形图。为了验证算法在多种地形条件下的性能,本文采用Foxhole Shekel优化问题的数学函数方法生成地形<sup>[13]</sup>,即

$$z = \sum_{i=1}^{N_T} \frac{0.1}{(x-a_i)^2 + (y-b_i)^2 + c_i} \quad (1)$$

式中, $N_T$ 表示山峰个数; $a_i$ 与 $b_i$ 用于控制山峰位置; $c_i$ 用于控制山峰高度。

就UAV的具体航迹而言,若航迹点的总个数为 $N$ ,其任意的第 $n$ 个航迹点 $w_n = (x_n, y_n, z_n)$ 应在地图范围内,即

$$\begin{cases} x_{\min} \leq x_n \leq x_{\max} \\ y_{\min} \leq y_n \leq y_{\max} \\ z_{\min} \leq z_n \leq z_{\max} \end{cases} \quad (2)$$

式中, $x_{\min}$ 、 $x_{\max}$ 、 $y_{\min}$ 、 $y_{\max}$ 、 $z_{\min}$ 与 $z_{\max}$ 分别对应于地图坐标 $x$ 、 $y$ 、 $z$ 的最小值以及最大值。

### 1.2 目标函数设计

#### 1.2.1 总代价

设计合理的目标函数对于UAV航迹规划这一优化问题至关重要。本文设计的优化目标函数综合考虑了航迹长度、UAV和地形之间的避碰以及UAV飞行高度的限制,该目标函数可以表示为

$$f = p_1 f_L + p_2 f_T + p_3 f_H \quad (3)$$

式中, $f_L$ 、 $f_T$ 以及 $f_H$ 分别为航迹长度代价、地形代价以及飞行高度代价; $p_1$ 、 $p_2$ 以及 $p_3$ 为对应的权重。通过调整权重可以调整优化目标,为了方便,本文采用 $p_1 = p_2 = p_3 = 1$ 。

#### 1.2.2 航迹长度代价

考虑到UAV的燃料限制,更短的航迹意味着UAV可以在燃料消耗完毕之前完成任务。另外,飞行时间越短,被未知威胁发现的概率同样更低。本文采用参考文献<sup>[13]</sup>提到

的路径长度比率(PLR)来描述航迹长度代价 $f_L$ ,即

$$f_L = \frac{\sum_{n=2}^N \|w_n - w_{n-1}\|_2}{\|w_N - w_1\|_2} \quad (4)$$

$f_L$ 越小表示航迹长度越短,越有利于飞行任务的完成。且 $f_L$ 这种表示方式并不会因为地图大小或起始点以及终点位置的改变而显著影响航迹长度代价数值,即 $f_L$ 对于不同地形图的适应能力更强。

#### 1.2.3 地形代价

为了实现安全飞行,UAV在整个航行过程中不能与地形发生任何碰撞,且UAV与地面的距离要满足最低安全距离的要求。虽然本文所提航迹规划算法的优化变量为航迹点 $w_n$ ,但UAV在航迹点之间航行时同样要避免同地形发生碰撞,因此本文将相邻航迹点按固定距离 $d$ 分为 $M_n$ 份,且为了降低算法复杂度, $w_{n-1}$ 与 $w_n$ 之间的坐标 $w_{m,n}$ 通过线性插值获得。另外, $d$ 的取值应综合考虑算法复杂度以及航迹规划结果的精度。那么,本文所提地形代价 $f_T$ 可以表示为

$$f_T = \sum_{n=2}^N \sum_{m=1}^{M_n} A_{m,n} \quad (5)$$

且

$$A_{m,n} = \begin{cases} 1, & Z_{m,n} \leq Z_{m,n}^T + d_{\text{safe}} \\ 0, & \text{其他} \end{cases} \quad (6)$$

式中, $A_{m,n} = 1$ 表示 $w_{m,n}$ 与地形图坐标 $(x_{m,n}, y_{m,n}, z_{m,n}^T)$ 之间的垂直距离小于最小安全距离 $d_{\text{safe}}$ ;  $A_{m,n} = 0$ 则表示该垂直距离大于最小安全距离 $d_{\text{safe}}$ 。

#### 1.2.4 飞行高度代价

受UAV动力学性能的影响,其飞行高度受限,且对于一些特殊的任务类型而言,飞行高度越高,被未知威胁发现的概率就越大。本文提出的飞行高度代价和地形代价类似,即飞行高度代价 $f_H$ 可以表示为

$$f_H = \sum_{n=2}^N \sum_{m=1}^{M_n} B_{m,n} \quad (7)$$

且

$$B_{m,n} = \begin{cases} 1, & Z_{m,n} \geq Z_{m,n}^T + H \\ 0, & \text{其他} \end{cases} \quad (8)$$

式中, $H$ 为UAV最大飞行高度。但和地形代价不同的是,飞行高度的限制往往不十分严格,因此可以通过调整目标函数中飞行高度代价所对应的权重 $p_3$ 来实现优化目标的动态调整,以满足任务需求。

针对UAV的其他性能约束(如最大水平飞行速度、最大旋转角速度、最大上升下降速度等)将在未来的研究中体现。

## 2 基于POA算法的UAV航迹规划

### 2.1 POA算法概述

POA算法受传教行为启发,分为初始化个体位置与目标函数数值计算、影响权重因子计算、向继承人传播知识、文化竞争以及文化发展5个步骤。本文则将POA算法用于UAV航迹规划之中。

POA算法首先生成 $p$ 个传教士,初始化其位置并完成目标函数数值的计算。然后归一化其目标函数值即计算影响权重因子 $f_{\text{aff}}(i, n_{\text{iter}})$ , $i$ 表示第 $i$ 个传教士, $n_{\text{iter}}$ 表示第 $n_{\text{iter}}$ 次迭代。在每个传教士向其对应的 $in$ 个继承人传播知识时,POA算法结合影响权重因子并根据人工免疫算法实现了可变量 $r/3$ 的全局搜索。为了加速算法收敛,POA算法在文化竞争步骤采用精英策略,即选择 $ein$ 个个体作为精英个体直接成为传教士进入下一步。同时为了保持个体的多样性,POA算法基于个体中心位置以及对应的目标函数数值计算了文化竞争步骤中的权重 $\text{weight}(i)$ ,该权重表示了个体与样本中心之间的差别,POA算法选择差别最大的 $p - ein$ 个个体作为传教士进入下一步,以维持样本多样性。最终,POA算法通过莱维飞行或高斯分布实现了文化发展,其本质为优化算法中的局部搜索。将上述过程迭代 $N_{\text{iter}}$ 次或达到算法收敛条件即可退出循环。

### 2.2 航迹点初始化

和其他优化问题不同的是,若航迹规划在算法初始阶段完全通过随机的方式设置UAV的航迹点会显著降低优化算法的优化效率。因此,本文采用了参考文献[14]所提的旋转坐标系来生成初始航迹。旋转坐标系本质上是一种沿 $xoy$ 平面旋转后的笛卡尔坐标系,为了方便,称之为旋转坐标系。图1给出了旋转坐标系和笛卡尔坐标系之间的关系。

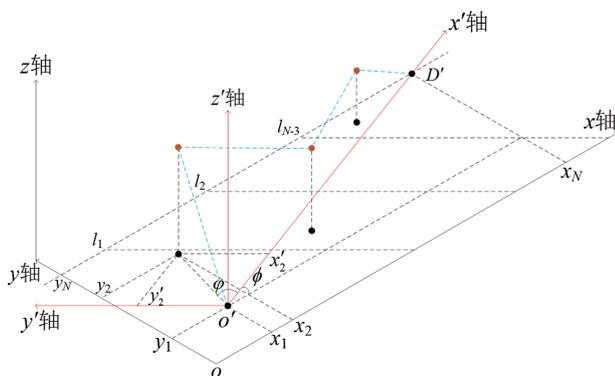


图1 旋转坐标系与笛卡尔坐标系示意图

Fig.1 Schematic diagram of rotated coordinate and Cartesian coordinate

图1中, $xyz$ 表示笛卡儿坐标系, $x'y'z'$ 表示旋转坐标系,旋转坐标系的 $x'$ 轴方向为UAV航迹起始点和终点的连线方向,而 $y'$ 轴方向则与 $x'$ 轴方向垂直, $x'O'y'$ 平面和 $xoy$ 平面平行, $z'$ 轴方向与 $z$ 轴方向相同,浅色实心点代表航迹点,黑色实心点代表航迹点在 $x'O'y'$ 平面的投影,旋转坐标系原点 $o'$ 为航迹起始点, $D'$ 点为航迹终点。笛卡儿坐标系内的坐标点 $(x_n, y_n, z_n)$ 与对应的旋转坐标系坐标点 $(x'_n, y'_n, z'_n)$ 之间的关系满足

$$\begin{cases} x_n = x_1 + \cos(\varphi + \phi) \times \sqrt{x_n'^2 + y_n'^2} \\ y_n = y_1 + \sin(\varphi + \phi) \times \sqrt{x_n'^2 + y_n'^2} \\ z_n = z_n' \end{cases} \quad (9)$$

式中, $\varphi$ 为航迹点与旋转坐标系原点连线和旋转坐标系 $x'$ 轴的夹角, $\phi$ 为旋转坐标系 $x'$ 轴和笛卡儿坐标系 $x$ 轴的夹角。

为了利用旋转坐标系生成初始航迹点,首先通过 $N - 3$ 条与 $y'$ 轴平行且在 $x'O'y'$ 平面上的直线将 $x'O'y'$ 平面沿 $x'$ 轴方向等分为 $N - 2$ 份,除了起始点和终点外的航迹点坐标依次在这 $N - 2$ 个区域内随机选择且服从均匀分布。对应的 $z_n \sim U(z_n^T + d_{\text{safe}}, z_n^T + H)$ 。根据该种方式生成的初始航迹由于结合了UAV航迹的方向特性,可以极大加速航迹规划算法的收敛速度,解决了原始POA算法的初始化问题。

### 2.3 UAV航迹规划优化方法

在利用POA算法进行UAV航迹规划时,需要利用旋转坐标系生成 $p$ 组初始航迹以代表 $p$ 个传教士, $\text{loc}_i$ 表示第 $i$ 个传教士的位置,对应随机生成的第 $i$ 组航迹点的集合 $\text{loc}_i = \{w_n | 1 \leq n \leq N\}$ 。将生成的初始值代入式(3)中即可完成目标函数的计算,值得注意的是,虽然在计算地形代价以及飞行高度代价时需要利用线性插值之后的结果,但航迹规划的优化变量仍为 $N$ 个航迹点 $w_n$ 。

此外,由于航迹点的 $x$ 、 $y$ 以及 $z$ 坐标无特殊的对应关系,因此在利用POA进行后续步骤时应将其作为三个独立的优化变量进行看待,仅在计算目标函数以及对应权重时将三者结合。又由于POA算法中各步骤均利用高斯分布进行搜索,因此基于POA算法航迹规划的每个步骤的输入以及输出变量应满足关系

$$\begin{cases} x_n^s = x_n + \text{randn}(0, \sigma_x^2) \\ y_n^s = y_n + \text{randn}(0, \sigma_y^2) \\ z_n^s = z_n + \text{randn}(0, \sigma_z^2) \end{cases} \quad (10)$$

式中, $x_n^s$ 、 $y_n^s$ 以及 $z_n^s$ 分别为航迹点坐标 $x_n$ 、 $y_n$ 以及 $z_n$ 每步优化后的结果, $\text{randn}(\mu, \sigma^2)$ 表示概率分布服从均值为 $\mu$ ,方差为

$\sigma^2$  高斯分布的随机变量,  $\sigma_x^2$ 、 $\sigma_y^2$  以及  $\sigma_z^2$  为其对应的高斯分布方差。

由于 POA 优化在搜索过程中涉及诸多服从高斯分布的随机变量, 即在每个步骤中优化后的航迹点  $(x_n^s, y_n^s, z_n^s)$  都可能超出地图范围, 因此需要进行边界条件处理。本文针对超出地图范围的航迹点结合均匀分布进行边界处理。即经过边界条件处理后的航迹点  $(x_n^b, y_n^b, z_n^b)$  满足  $x_n^b \sim U(x_{\min}, x_{\max})$ 、 $y_n^b \sim U(y_{\min}, y_{\max})$  以及  $z_n^b \sim U(z_{\min}, z_{\max})$ 。与优化过程类似的是, 边界条件处理过程同样将三维坐标视作三个独立的变量, 仅对超出边界的坐标进行边界条件处理。

经过向继承人传播知识、文化竞争以及文化发展三步之后即完成了一次 POA 算法的迭代, 对应的传教士位置分别记作  $loc'(i)$ 、 $loc''(i)$  以及  $loc'''(i)$ , 在进行影响权重因子的数值比较后利用  $loc'''(i)$  对  $loc(i)$  进行更新。最终, 整个基于 POA 算法的 UAV 航迹规划将在达到最大迭代次数  $N_{iter}$  或满足迭代停止条件  $|f_{\min}(n_{iter}) - f_{\min}(n_{iter} - 1)| \leq \tau$  后停止迭代。

$f_{\min}(n_{iter})$  的定义为

$$f_{\min}(n_{iter}) = \min_{1 \leq i \leq p} \{f(i, n_{iter})\} \quad (11)$$

式中,  $f(i, n_{iter})$  为第  $n_{iter}$  次迭代中第  $i$  个传教士的目标函数值。综上所述, 基于 POA 算法的 UAV 航迹规划伪代码如下:

算法 1 基于 POA 算法的 UAV 航迹规划	
输入: 地形参数、最小安全距离 $d_{safe}$ 、最大飞行高度 $H$ 、线性插值步长 $d$ 、传教士数量 $p$ 、迭代次数 $N_{iter}$ 、继承人个数 $in$ 以及精英数量 $ein$	
输出: 最优航迹点集合 $loc_j = \{w_n   1 \leq n \leq N\}$	
1	根据旋转坐标系生成初始航迹 $loc_i$ ;
2	$n_{iter} = 1, f_{\min}(0) = +\infty$ ;
3	while $n_{iter} \leq N_{iter}$ do
4	计算 $f_{aff}(i, n_{iter})$ 以及 $f_{\min}(n_{iter})$ ;
5	if $ f_{\min}(n_{iter}) - f_{\min}(n_{iter} - 1)  \leq \tau$
6	Break;
7	end if
8	结合影响权重因子以及人工免疫算法确定高斯分布方差 $r_x/3, r_y/3$ 以及 $r_z/3$ , 并向继承人传播知识生成 $loc'(i)$ ;
9	边界条件处理;
10	选择 $ein$ 个精英作为传教士进入下一步, 计算其余权重 $weight(i)$ 并选择剩余 $p - ein$ 个个体作为传教士, 生成 $loc''(i)$ ;
11	进行文化发展生成 $loc'''(i)$ ;
12	边界条件处理;
13	比较影响权重因子并对 $loc(i)$ 进行更新;
14	end while
15	输出最优航迹 $loc_j, j = \arg \min_{1 \leq i \leq p} \{f_{aff}(loc_i)\}$ 。

表 1 POA 算法参数

Table 1 Parameters setting of POA

传教士数量 $p$	500
继承人数量 $in$	5
精英数量 $ein$	375
人工免疫算法收缩系数最大值/最小值	0.01/0.0001
最大迭代次数 $N_{iter}$	100

### 3 仿真结果

#### 3.1 仿真参数设置

本文将从航迹规划结果、算法收敛速度以及对于不同地形的适应程度三个方面对比本文所提出的基于 POA 算法的 UAV 航迹规划以及基于 CIPSO<sup>[11]</sup> 航迹规划的性能。其中, 地图大小为  $10\text{km} \times 10\text{km} \times 0.5\text{km}$ , 式中各个参数服从  $a_i \sim U(0, 10)$ 、 $b_i \sim U(0, 10)$  以及  $c_i \sim U(0.2, 0.5)$  (此处设置山峰的最低高度为 200m), 山峰总数  $N_T = 30$ , 最小安全距离  $d_{safe} = 5\text{m}$ , 最大飞行高度  $H = 120\text{m}$ , 航迹点个数  $N = 15$ , 线性插值步长  $d = 20\text{m}$ 。POA 算法的参数见表 1, CIPSO 算法的参数见表 2。综合考虑到算法收敛速度以及精度, 将退出 POA 迭代的参数设置为  $\tau = 0.002$ 。航迹起始点坐标为  $(0, 0, 0.0827)$ , 航迹终点坐标为  $(10, 10, 0.0737)$ 。

表 2 CIPSO 算法参数

Table 2 Parameters setting of CIPSO

种群大小	500
惯性权重最大值/最小值	0.9/0.4
加速度常量最大值/最小值	3.5/0.5
速度常量 1	0.25
速度常量 2	0.05
最大迭代次数 $N_{iter}$	100

#### 3.2 航迹规划仿真结果对比

图 2 给出了本文所提算法的航迹规划结果以及基于 CIPSO 航迹规划结果的三维视图以及俯视图。从图 2 中可以看出, 在该地形条件下, 本文所提算法成功实现了航迹规划, 没有和地形产生任何碰撞并且满足最小安全距离限制以及最大飞行高度限制, 航迹总长度为 14.3km。而基于 CIPSO 的航迹规划输出的航迹长度为 17.5km, 这说明本文所提出的目标函数结合旋转坐标系的初始化方法以及 POA 算法可以实现在多种限制条件下的航迹规划, 且输出航迹长度优于基于 CIPSO 的航迹规划方法。

图 3 给出了本文所提算法的收敛过程, 所采用的地形图和图 2 所采用的地形图相同。通过 POA 算法中收缩系数以及人工免疫算法的加入, 并结合精英策略以及文化竞争

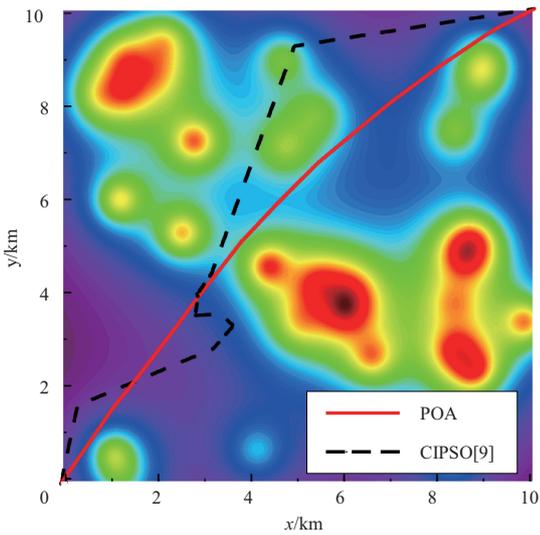
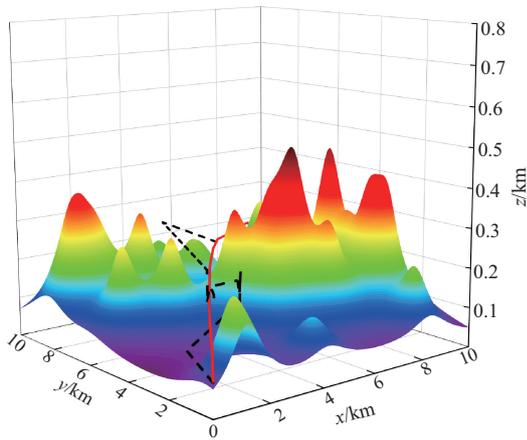


图2 航迹规划长度对比

Fig.2 Comparison between path planning length

步骤,本文所提算法在60次迭代后实现了收敛。此外,旋转坐标系的加入使得POA算法初始值的目标函数数值明显低于CIPSO,且收敛速度以及收敛后的目标函数大小均优于CIPSO。

本文同样利用随机生成地形图的方法评估了所提算法对于不同环境的适应能力。图4给出了在100个不同地形图情况下本文所提算法航迹规划结果的航迹长度,∞表示航迹不满足地形限制或飞行高度限制。可以看到在地形变化剧烈的情况下,本文所提算法在100次不同的地形条件下均成功完成UAV航迹规划,而基于CIPSO方法的成功率为72%。本文所提算法的平均航迹长度为14.5km,而基于CIPSO的航迹规划方法输出的平均航迹长度为16.6km。综上所述,本文所提算法在针对不同地形的稳定性以及输出

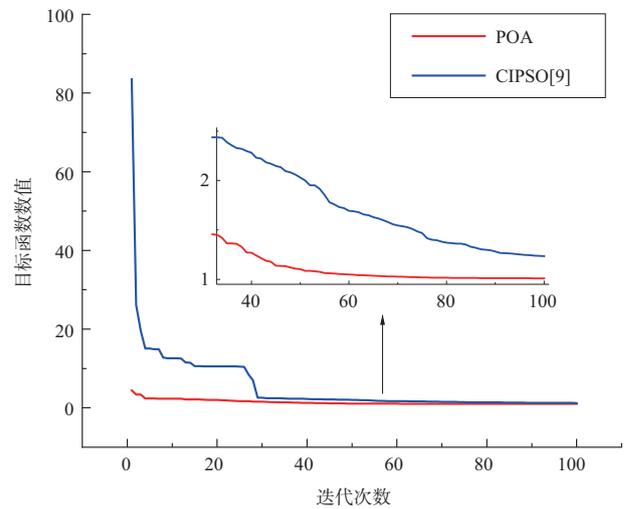


图3 收敛速度对比

Fig.3 Comparison between convergence speed

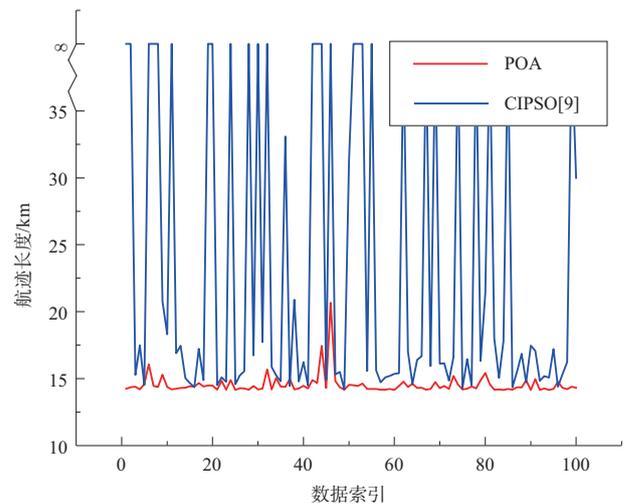


图4 不同地形适应度对比(100次)

Fig.4 Comparison between fitness in 100 different terrain conditions

航迹长度两个方面均优于CIPSO。

在解算实时性方面,本文通过在MATLAB平台上的运行时间说明了二者在实际运行速度上的差别。采用的计算机配置见表3,运行时间对比结果见表4。本文所提算法的运行时间为138.37s,CIPSO算法的运行时间为149.65s。

表3 计算机参数

Table 3 Computer parameters

CPU	Intel i5-8250U
硬盘大小	256GB
内存大小	8GB
Windows版本	Windows 10
MATLAB版本	MATLAB R2018a

表4 运行时间对比结果

Table 4 Results of running time

算法名称	运行时间/s
POA	138.37
CIPSO	149.65

下面分析本文所提算法性能优于CIPSO的具体原因。首先,CIPSO在航迹初始化时没有采用旋转坐标系进行初始航迹的生成,而是采用Chaos-based<sup>[15]</sup>的方法在整个规划空间进行航迹点的初始化。这种做法的确可以提升初始航迹的随机性,但由于没有结合UAV航迹的方向特性,因此在初始化时其目标函数数值明显高于基于旋转坐标系的生成方法。

此外,CIPSO算法为了加快算法收敛速度,在粒子位置更新时将目标函数较小的粒子叠加适当的偏移量替代目标函数较大的粒子,并将目标函数较小粒子的速度直接替换目标函数较大粒子的速度。这种策略的确可以提升收敛速度,但同时降低了样本的多样性。而POA算法为了提升样本的多样性在文化竞争中保留了部分代价函数较大的个体并作为传教士进入下一次迭代。维持样本多样性对于航迹规划而言十分重要,由于地形以及UAV自身限制条件的复杂性,UAV航迹规划将存在多个局部最优,而样本多样性对于跳出局部最优具有重大意义。

## 4 结论

本文针对UAV航迹规划问题,将POA算法用于UAV航迹规划之中。首先,本文提出了结合航迹长度、地形限制以及UAV飞行高度限制的优化目标函数。其次,将旋转坐标系引入POA算法的初始化步骤之中以加快算法收敛,并说明了基于POA算法的UAV航迹规划流程。最后,通过和CIPSO算法的仿真对比结果可知,对于同一地形图,本文所提算法规划的航迹长度为14.3km,CIPSO为17.5km,且本文所提算法在60次迭代后实现收敛,初始值的目标函数数值以及收敛速度同样优于CIPSO。针对100个不同地形图的仿真结果可知,本文所提算法规划成功率为100%,平均航迹长度为14.5km,而CIPSO成功率为72%,平均航迹长度为16.6km。本文所提算法在计算机平台上的运行时间为138.37s,而CIPSO为149.65s。以上对比结果均说明了本文所提基于POA算法的航迹规划优化方法优于CIPSO,证明了算法的有效性。

## 参考文献

- [1] 吴兆香,欧阳权,王志胜,等.基于人工智能的无人机区域侦察方法研究现状与发展[J].航空科学技术,2020,31(10):57-68.  
Wu Zhaoxiang, Ouyang Quan, Wang Zhisheng, et al. Status and development of regional reconnaissance methods of UAV based on artificial intelligence[J]. Aeronautical Science & Technology, 2020, 31(10):57-68.(in Chinese)
- [2] 张广林,李国知.美国陆军航空装备体系研究[J].航空科学技术,2021,32(1):65-69.  
Zhang Guanglin, Li Guozhi. Research on U.S. army aviation equipment system [J]. Aeronautical Science & Technology, 2021, 32(1):65-69.(in Chinese)
- [3] Zhang Hao, Xin Bin, Dou Lihua, et al. A review of cooperative path planning of an unmanned aerial vehicle group[J]. Frontiers of Information Technology & Electronic Engineering, 2020, 21(12): 1671-1694.
- [4] Dorigo M, Maniezzo V. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996, 26(1): 29-41.
- [5] Storn R, Price K. Minimizing the real functions of the ICEC'96 contest by differential evolution[C]// Proceedings of IEEE International Conference on Evolutionary Computation. IEEE, 1996.
- [6] Pan Wen-Tsao. A new fruit fly optimization algorithm: Taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012, 26(2):69-74.
- [7] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]// MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. IEEE, 1995.
- [8] Tian Wei, Yang Zhihua. A grid-map-oriented UAV flight path planning algorithm based on ACO algorithm[C]// 7th International Conference on Communications, Signal Processing, and Systems, 2020.
- [9] Zhang Hao, Dou Lihua, Cai Chunxiao, et al. Three-dimensional unmanned aerial vehicle route planning using hybrid differential evolution[J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2020, 24(7):820-828.
- [10] Li Kun, Ge Fawei, Han Ying, et al. Path planning of multiple

- UAVs with online changing tasks by an ORPFOA algorithm[J]. *Engineering Applications of Artificial Intelligence*, 2020, 94: 103807.
- [11] Shao Shikai, Peng Yu, He Chenglong, et al. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization[J]. *ISA Transactions*, 2020, 97: 415-430.
- [12] Wei Dong, Wang Zhongbin, Si Lei, et al. Preaching-inspired swarm intelligence algorithm and its applications[J]. *Knowledge-Based Systems*, 2021, 211(9):106552.
- [13] Yang Peng, Tang Ke, Lozano J, et al. Path planning for single unmanned aerial vehicle by separately evolving waypoints[J]. *IEEE Transactions on Robotics*, 2015, 31(5):1130-1146.
- [14] Yang Peng, Tang Ke, Lozano J. Estimation of distribution algorithms based unmanned aerial vehicle path planner using a new coordinate system[C]// 2014 IEEE Congress on Evolutionary Computation. IEEE, 2014.
- [15] Tian Dongping, Shi Zhongzhi. MPSO: Modified particle swarm optimization and its applications[J]. *Swarm and Evolutionary Computation*, 2018,41:49-68.

## Optimization Method of Unmanned Aerial Vehicle Path Planning Using POA

Jia Hanbo<sup>1</sup>, Ma Lin<sup>1</sup>, Zhang Zhongwang<sup>2</sup>

1. Harbin Institute of Technology, Harbin 150006, China

2. China National Aeronautical Radio Electronics Research Institute, Shanghai 200241, China

**Abstract:** Aiming at the optimization problem of path planning in the automatic control field of unmanned aerial vehicle (UAV), a UAV path planning optimization method based on preaching optimization algorithm (POA) is proposed in this paper. In this method, the optimization objective function model including path length, terrain cost and flight altitude cost is designed, the rotating coordinate system is introduced to accelerate the convergence speed of the algorithm when initializing the preacher's position, and the specific problems and processing methods of the combination of POA and UAV path planning are described in detail. Finally, the simulation results prove the effectiveness of the UAV path planning optimization method based on POA in terms of path length, convergence speed and adaptability to different terrain maps compared with the comprehensive improved particle swarm optimization based method.

**Key Words:** UAV; path planning; optimization algorithm; swarm intelligence optimization; preaching optimization algorithm

---

**Received:** 2022-02-20; **Revised:** 2022-04-25; **Accepted:** 2022-06-19

**Foundation item:** National Natural Science Foundation of China (61971162, 41861134010); Aeronautical Science Foundation of China (2020Z066015002)