

# 基于粒子群算法迭代优化的无人机队形调整

## 摘要

对于第一问的第三小问，鉴于题目没有对圆周半径做硬性要求，我们选择优化算法：**粒子群优化算法 (PSO)**，通过**逐轮迭代**的方式，将9架初始位置不精确的无人机，逐步引导至一个以中心无人机 (FY00) 为圆心、且成员间均匀分布的圆形编队。粒子群算法兼顾了全局最优和个体最优，且收敛速度快，适合无人机实时调整的应用场景。

## 问题重述与分析

本小题设定了一个圆形编队场景：1架中心无人机 (FY00) 位于编队圆心，其余9架无人机 (FY01-FY09) 应精确且均匀地分布在以FY00为中心的同一圆周上。然而在实际飞行过程中，无人机群不可避免地会存在初始位置偏差。因此，我们的任务是设计出一个调整方案，使得所有无人机能够仅依赖自身获取的位置信息，自主调整到一个全局最优且均匀的理想圆形编队。

由于编队中所有无人机的位置是高度耦合的，对某一架无人机进行局部调整，很可能引发连锁反应，破坏其他无人机的位置，进而导致整个队形的振荡、恶化。因此，我们要寻求一种能够同时优化所有9架无人机位置的全局优化方法，在18维向量空间（9架无人机，每架2个坐标）中，找到能使整体队形最优的全局解。

## 模型建立

本小问中的数学建模就是将"理想圆形队伍"这一抽象概念，建模为一个可以计算、比较的数学模型。

## 指标

我们设置了两个可量化的指标，来衡量无人机所在位置相对于正确位置的好坏：

- 半径一致性**：所有9架外围无人机到编队中心FY00的欧氏距离应相等。
- 角度均匀性**：从编队中心看去，任意两架相邻无人机之间的夹角应严格等于理论值  $360^\circ / 9 = 40^\circ$ 。

基于这两个指标，我们构建了一个**适应度函数**，将一个编队构型映射为一个实数，其值越小，代表编队质量越高，越接近理想状态，和深度学习里的损失函数概念相似。PSO算法里的一个粒子，也就是一个潜在的解，是由9架无人机的  $9 \times 2 = 18$  个坐标值构成的向量定义的， $P = [x_1, y_1, x_2, y_2, \dots, x_9, y_9]$ 。

## 适应度函数

适应度函数的计算如下：

- 半径偏差成本**：所有无人机半径  $r_i$  与平均半径  $r_{avg}$  的方差之和来度量，之所以使用方差而不是标准差，是因为方差能更严厉地惩罚极端值，更为强力地引导无人机向平均半径收敛。

$$Cost_R = \sum_{i=1}^9 (r_i - \bar{r})^2$$

其中， $r_i = \sqrt{x_i^2 + y_i^2}$ ， $\bar{r}$  为9架无人机半径的算术平均值。

- 角度间隔成本**：所有无人机之间的实际间隔角度  $\Delta\theta_i$  与理想角度  $40^\circ$  的方差之和。

$$Cost_{angle} = \sum_{i=1}^8 (\Delta\theta_i - 40^\circ)^2 + (\Delta\theta_{9,1} - 40^\circ)^2$$

其中,  $\Delta\theta_i$  是对所有无人机按极坐标角度排序后, 第  $i$  架与第  $i+1$  架无人机之间的角度差。

$\Delta\theta_{\{9,1\}}$  是第9架与第1架之间的角度差, 这是为了保证圆周的闭环。

- **适应度**: 将上述两个成本进行加权求和。我们设置权重  $w_1$  和  $w_2$  均为1.0, 表示同等重视半径的一致性和角度的均匀性。若某特定任务对其中一项有更高要求, 则可调整该权重。

$$Fitness = w_1 \cdot Cost_R + w_2 \cdot Cost_{angle}$$

## 粒子群优化算法(PSO)

我们选择的粒子群优化算法是一种有效且计算成本较低的全局优化算法。它通过模拟鸟群觅食, 让每个粒子 (潜在解) 同时借鉴自身的“历史经验”和群体的“集体智慧”, 从而在复杂的解空间中高效地寻找全局最优解。在粒子群算法的每一次迭代中, 每个粒子都要更新自己的速度和位置。

### 更新公式

- **速度更新公式 (Velocity Update):**

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (pbest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gbest - x_i(t))$$

- **位置更新公式 (Position Update):**

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

速度更新是PSO的精髓, 它决定了粒子下一步“飞”是往哪个方向, 以及“飞”多快。PSO中的惯性、认知、社会分别对应速度更新公式中的  $w \cdot v_i(t)$ 、 $c_1 \cdot r_1 \cdot (pbest_i - x_i(t))$ 、 $c_2 \cdot r_2 \cdot (gbest - x_i(t))$ 。其中,

- **惯性部分:  $w \cdot v_i(t)$** 
  - **含义**: 代表粒子保持其当前运动状态的“惯性”或“记忆”。
  - $v_i(t)$ : 是粒子在当前时刻  $t$  的速度。
  - $w$  (**惯性权重**): 我们设置为 0.5。这个参数用于平衡算法的**全局探索**和**局部开发**能力。
    - $w$  值增大会使粒子倾向于保持当前方向。
    - $w$  值减小会减弱原始速度的影响, 使粒子更容易被“认知”和“社会”部分吸引, 从而在已知最优解的附近进行局部搜索。
- **认知部分:  $c_1 \cdot r_1 \cdot (pbest_i - x_i(t))$** 
  - **含义**: 代表粒子的“个体经验”, 将粒子拉向其**自身历史上找到过的最佳位置**。
  - $pbest_i$ : 是第  $i$  个粒子从开始到现在所经历过的最低适应度值的位置。
  - $(pbest_i - x_i(t))$ : 是一个向量, 指向从粒子当前位置到其自身历史最佳位置的方向。
  - $c_1$  (**认知系数**): 在代码中被设置为 0.8, 是一个学习因子, 决定粒子飞向自身最佳位置的“加速度”。
  - $r_1$ : 一个在  $(0, 1)$  区间内的随机数。它的存在为搜索过程增加了随机性, 使得粒子的行为不是完全确定的, 有助于跳出局部最优。
- **社会部分:  $c_2 \cdot r_2 \cdot (gbest - x_i(t))$** 
  - **含义**: 代表粒子向群体学习的“集体智慧”, 将粒子拉向**整个种群目前为止发现的最佳位置**。
  - $gbest$ : 是整个粒子群从开始到现在所找到的全局最佳位置。

- $(g_{best} - x_i(t))$ : 是一个向量, 指向从粒子当前位置到全局最佳位置的方向。
- $c_2$  (**社会系数**): 在代码中被设置为 0.9, 也是一个学习因子, 决定粒子飞向种群全局最佳位置的“加速度”。
- $r_2$ : 另一个在 (0, 1) 区间内的随机数。

位置更新用来模拟例子在物理世界中的运动, 非常直观, 下一时刻的位置  $x_i(t + 1)$  就是当前所在的位置  $x_i(t)$  加上新速度的位移  $v_i(t + 1)$ 。我们设置时间步长  $\Delta t$  为1。

## 算法流程

在预设的搜索空间内, 随机生成一群粒子。每个粒子被赋予一个随机的位置 (一套完整的无人机坐标) 和速度。然后通过适应度函数, 计算每个粒子的当前适应度值。

在每一次迭代中, 每个粒子都会更新其速度和位置。其更新方向是三个向量的合成: 自身的惯性、飞向自身历史最优位置 (pbest, 个体经验) 的趋势、以及飞向整个种群至今为止发现的全局最优位置 (gbest, 社会知识) 的趋势。

经过足够多的迭代后, 整个粒子群将大概率收敛到适应度函数值最低的区域。此时全局最优位置 gbest 所对应的坐标, 即为我们寻求的最优编队方案。

## 参数设置

以下是PSO算法的超参数设置, 其选择是在兼顾收敛速度与求解质量后确定的:

- **粒子数量 (n\_particles)**: 500。我们设置了较大的种群规模, 使之能更好地搜索18维空间, 降低陷入局部最优解的风险。
- **迭代次数 (n\_iterations)**: 1000。尽管PSO能快速收敛, 但我们仍然设置了1000轮次的迭代, 让它更精确地收敛到全局最优。
- **惯性权重 (w)**: 0.5。我们设置为0.5, 相当折中, 使粒子能在前期广泛搜索, 在后期精细调整。
- **认知系数 (c1)**: 0.8。它决定粒子多大程度上相信并返回自己曾经找到过的最优位置。
- **社会系数 (c2)**: 0.9。它决定了粒子多大程度上跟随整个群体发现的最优位置。我们把 c2 设置得略大于 c1 有助于加快种群的收敛速度。
- **搜索空间**: 基于初始位置, 在每个坐标维度上设定 +/- 50 米的动态边界。

## 优化效果对比

(到时候图片放这里)

左图 (初始位置): 无人机 (红色圆点) 较为散乱地分布在中心点 (蓝色方块) 周围。肉眼可直接看出, 各无人机的半径和角度间隔均不一致, 队形松散。

右图 (优化后位置): 所有无人机 (绿色圆点) 如规则地排列在一个的圆周上, 间距均匀, 达到了理想的效果。

初始的FY01-FY09的极坐标:

FY01: 半径=100m, 角度=0°

FY02: 半径=98m, 角度=40.10°

FY03: 半径=112m, 角度=80.21°

FY04: 半径=105m, 角度=119.75°

FY05: 半径=98m, 角度=152.249220°

FY06: 半径=112m, 角度=159.86°  
FY07: 半径=105m, 角度=240.07°  
FY08: 半径=98m, 角度=280.17°  
FY09: 半径=112m, 角度=320.28°

优化后的FY01-FY09的极坐标:  
FY01: 半径=98.559534m, 角度=-7.714802°  
FY02: 半径=98.560885m, 角度=32.290028°  
FY03: 半径=98.558919m, 角度=72.293749°  
FY04: 半径=98.560312m, 角度=112.275301°  
FY05: 半径=98.563620m, 角度=152.249220°  
FY06: 半径=98.571438m, 角度=-167.771245°  
FY07: 半径=98.544763m, 角度=-127.784741°  
FY08: 半径=98.574347m, 角度=-87.775436°  
FY09: 半径=98.562958m, 角度=-47.737670°

| 指标         | 初始编队   | 优化后编队  | 提升效果       |
|------------|--------|--------|------------|
| 半径方差 (m)   | 31.429 | 0.0001 | 显著降低, 半径一致 |
| 角度间隔方差 (°) | 0.04   | 0.0004 | 显著降低, 均匀间隔 |

根据数据分析, 可以发现优化效果极为显著。初始编队中较大的半径和角度方差, 反映了其混乱无序的状态。经过PSO迭代优化后, 这两项核心指标均降至几乎为零, 这不仅仅是数值上的提升, 更代表了编队从一个无序状态到高度有序的、精确的几何结构的质的飞跃, 表明算法成功找到了满足我们所有约束条件的完美解。