

无人机灯光秀建模

胡小忠

摘 要

随着科技的迅猛发展，无人机逐渐走进人们生活的方方面面，不断执行着人类难以完成的任务同时，也在不断地丰富着人类生活。使用无人机进行的灯光汇演在有很强大可塑性的特点上，还仅具备清洁能源和可持续使用的特点。使用无人机作为灯光汇演的工具需要进行大规模的调度和规划，本文就如何调度和排布无人机进行研究。[1]

目录

1 问题提出与分析	2
1.1 问题背景	2
1.2 问题 1	2
1.3 问题 2	2
1.4 问题 3	2
2 模型假设	3
3 符号说明	3
4 模型建立与求解	3
4.1 问题一：各个画面布局模型及计算	3
4.1.1 摩天轮的初始布局及起飞布局	3
4.1.2 功夫熊猫起飞布局	5
4.1.3 CUMT 起飞布局	6
4.2 问题 2：规划无人机飞行轨迹	7
4.3 粒子群算法	7
4.3.1 基于粒子群算法的无人机路径规划	8
5 总结	8

1 问题提出与分析

1.1 问题背景

无人机灯光秀过程包括无人机起飞、画面切换、无人机降落等过程，其中无人机的飞行控制、画面切换方式、飞行路径规划等是关键问题。假设采用 400 架最快飞行速度为 3 米/秒，续航能力为 20 分钟的无人机，进行以下问题的研究：

1.2 问题 1

选定三个画面进行逐个切换表演

1. 摩天轮
2. 功夫熊猫
3. CUMT

确定无人机起飞场地要求和灯光秀时所需天空区域大小。

1.3 问题 2

建立数学模型来规划无人机飞行轨迹。优化“起飞->画面切换->降落”整个过程，确定各阶段无人机的起飞时间、飞行时间和飞行轨迹等相关参数，使得整个无人机灯光秀过程中画面展示的时间最优，论文中至少给出 10 架无人机的飞行参数。

1.4 问题 3

考虑飞行安全问题。若规定飞行过程中两架无人机之间的距离小于 1 米为危险飞行，建立模型判断问题 2 无人机飞行过程中，是否会出现危险飞行的情况，有哪些无人机出现了危险飞行的情况。根据你的判断结果改进问题 2 中的飞行规划模型，并给出改进后的飞行参数及画面展示时间。

2 模型假设

模型假设：

1. 无人机在二维平面空间中进行位移操作。
2. 无人机在静止状态下几何形状等效为一个直径为 $30cm$ 的圆。
3. 无人机的加速度 a 为 ∞ ，即更换方向和加速到某一速度 v 时间 t 可以任意小。
4. 无人机在空中的滞留可以任意稳定。
5. 无人机升空表演时，考虑相互影响作用，更换为直径 df 为 $0.5m$ 的圆。
6. 观测当天天气任意好，无光照污染。

3 符号说明

符号	说明
v	飞行器飞行速度
a	飞行器加速度
t	飞行器加速到某一速度所需时间
df	飞行器直径
c	摩天轮外径周长
d	摩天轮直径

4 模型建立与求解

4.1 问题一：各个画面布局模型及计算

4.1.1 摩天轮的初始布局及起飞布局

将摩天轮分解为一个小圆、一个大圆、四条直径线。每个元素的尺寸代表构成该元素所需要的无人机个数，其每个结构的大小如下表 1 所示，具体推导计算见公式(1)和(2)。其

具体形状和排布方式如下图 1 所示。在该布局中，将摩天轮分解为三个基本元素，并将其映

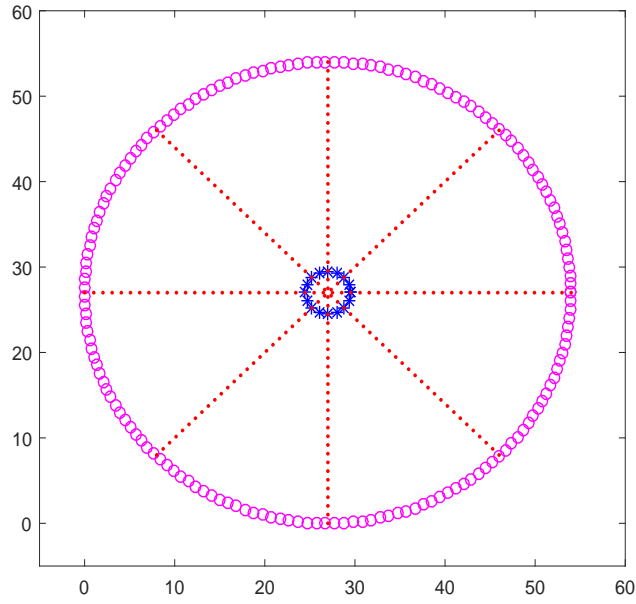


图 1: 摩天轮初始状态排布图

射到笛卡尔坐标系中，可以得到每个基本元素满足的方程为

$$c_1 + C_1 + 4d = 400 \quad (1)$$

$$d\pi = C_1 \quad (2)$$

其中 c_1 为小圆的周长， C_1 为大圆的周长， d 为线型结构的长度。易推导得到一个比例大小较为合适的结果为 在无人机进行起飞之前所需场地由摩天轮这一画面所决定，可以使用很

表 1: 摩天轮所需无人机数目

元素	尺寸 (个)
小圆	16*1
大圆	168*1
线段	56*8

少的占地面积，也方便之后的画面切换。根据计算，当大圆直径 d 为 56 单位无人机时，所

需占地面积为无人机直径 $df = 0.3m$ 与单位数的乘积。则起飞所需占地面积为：

$$\begin{aligned} S &= \left(\frac{df \times d}{2}\right)^2 \times \pi \\ &= \left(\frac{0.3 \times 56}{2}\right)^2 \times \pi \\ &= 221.7m^2 \end{aligned}$$

当无人机起飞后，考虑到无人机之间的相互影响，以及灯光秀时两个发光无人机灯效的连接性，可以将无人机尺寸模型估计为直径 df 为 $0.5m$ 的圆，这样可以尽可能扩大表演空间，也可以很好的保证表演效果不至于特别散。那么在起飞过程中，根据无人机发散飞行参数可以推导得出最终的天空所占区域面积为：

$$\begin{aligned} S &= \left(\frac{df \times d}{2}\right)^2 \times \pi \\ &= \left(\frac{0.5 \times 56}{2}\right)^2 \times \pi \\ &= 615.8m^2 \end{aligned}$$

4.1.2 功夫熊猫起飞布局

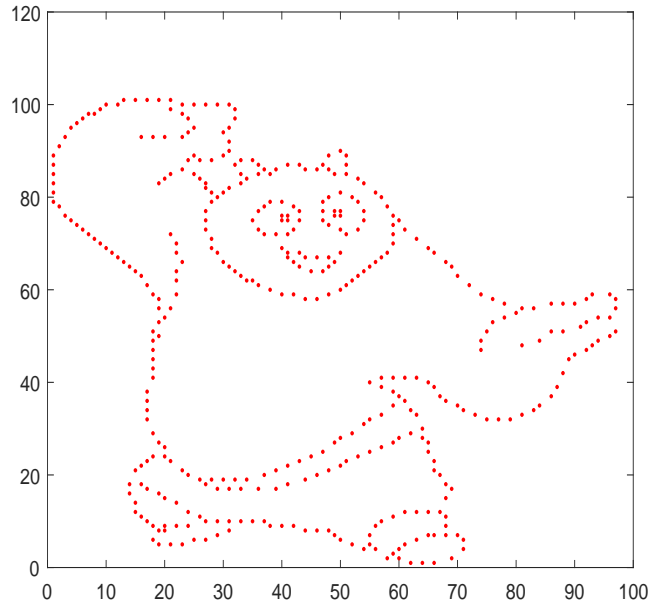


图 2: 功夫熊猫排布图

功夫熊猫形状数据由题目给出，将其读出并映射到笛卡尔坐标系中可以得到如图2所示结果。由于该画面不在地面活动，仅在天空进行表演，在所给数据中，数据大小为 101×97 ，无人机升空后模型仍等效为直径 df 为 $0.5m$ 的圆。所以其占地面积计算由所给数据可直接计算得出：

$$S = 101 \times 0.5 \times 97 \times 0.5 = 99m^2$$

4.1.3 CUMT 起飞布局

在该布局中，仍然采用分割元素的方法将其拆分为一个一个的小元素，然后将其拼接完成形成一个完整的图，其完整的排布效果如下图3所示。该图可完整拆分为两个半圆和七

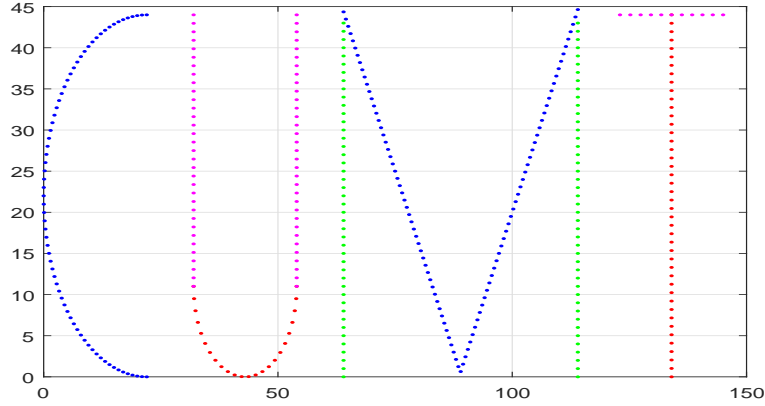


图 3: CUMT 排布图

条直线，根据该结果计算每个元素所需的无人机数目：

$$\begin{cases} \frac{1}{2}\pi d_1 + \frac{1}{2}\pi d_2 + 2l_2 + 2l_{3_1} + 2l_{3_2} + l_{4_1} + l_{4_2} = 400 \\ d_1 = 2d_2 \\ l_{3_2} = \frac{2}{\sqrt{3}}l_{3_1} \\ l_{4_1} = \frac{1}{2}l_{4_2} \end{cases} \quad (3)$$

根据方程组(3)可计算得到下表2参数。在该布局下，每个无人机升空后仍等效为一个直径 df 为 $0.5m$ 的圆形。在每个节点处均设置唯一一个无人机，防止有无人机的位置交叉。

表 2: CUMT 布局各个参数

参数	所需无人机单位数 (个)
d_1	69
d_2	35
l_2	22
l_{3_1}	44
l_{3_2}	50
l_{4_1}	22
l_{4_2}	44

4.2 问题 2: 规划无人机飞行轨迹

4.3 粒子群算法

粒子群算法模拟鸟群随即搜寻事物的捕食行为, 用于解决优化问题。在粒子群算法中, 每个优化问题的潜在解都可以想象成搜索空间的一只鸟, 称之为“粒子”。所有的粒子都有一个被优化的函数决定的适应度值, 每个粒子还有一个函数决定他们的飞行方向和距离。然后粒子就在当前最优解空间中搜索。粒子群算法初始化一群随机粒子, 然后通过迭代找到最优解。在每一次迭代过程中, 粒子通过跟踪两个极值来更新自己: 第一个就是粒子本身所找到的最优解, 这个解称为个体极值, 另一个极值是整个种群目前找到的最优解, 这个极值是全局极值。[2]

假设在一个 d 维的目标搜索空间中, 有 n 个粒子组成一个群落, 其中第 i 个粒子是一个 d 维向量:

$$X_i = (x_{i1}, x_{i2}, \dots, x_{id}), i = 1, 2, 3, \dots, n \quad (4)$$

第 i 个粒子的飞行速度也是一个 d 维向量, 记为

$$V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{id}), i = 1, 2, 3, \dots, n \quad (5)$$

第 i 个粒子搜索到的最优位置称为个体极值, 记为

$$P_{best} = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{id}), i = 1, 2, 3, \dots, n \quad (6)$$

整个粒子群搜索到的最优位置为全局极值，记为

$$g_{best} = (p_{g1}, p_{g2}, \dots, p_{gd}) \quad (7)$$

在找到这两个最优值时，粒子根据式 8、式 9、来更新自己的速度和位置

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id}) \quad (8)$$

$$x_{id} = x_{id} + v_{id} \quad (9)$$

4.3.1 基于粒子群算法的无人机路径规划

无人机的路径规划目的是在满足燃料消耗、安全区域的约束条件下，为无人机规划出一条从起始点到目标点评价最高的一条路径。飞行路径规划是一个空间搜索问题，主要是分为启发式算法和进化算法。其中的启发式算法由于搜索空间大，计算量大，并且规划效果对启发函数的依赖性强；进化算法是一种基于种群的优化算法，常用的算法有蚁群算法、遗传算法等。这些算法在一定程度上有很大缺陷，比如搜索量大、效率不高等问题。所以在此处使用粒子群算法进行无人机的路径规划。[3]

5 总结

基于粒子群算法的无人机路径规划算法可以很大程度上优化在安全区域、燃料消耗限制条件下的路径规划问题，并且可以在计算量不是很大、计算难度不是很高的情况下快速取得最优解。理论证明，基于粒子群的路径规划算法是一种相对有效和优越的解决方法。

参考文献

- [1] 易文; 雷斌;. 基于一致性的无人机编队飞行几何构型控制. 武汉科技大学学报, 42:150–154, 2019.
- [2] 李文光; 王强; 曹严. 基于 gpu 的大规模无人机编队控制并行仿真方法. 弹箭与制导学报, pages 1–5.
- [3] 张建南; 刘以安; 王刚;. 基于优化粒子群算法的无人机航路规划. 传感器与微系统, pages 58–61, 2017.

6 附录代码

```
%% 无人机灯光秀数学建模无人机绘图
```

```
% 作者：胡小忠
```

```
% 日期：2019年4月5日
```

```
%% 初始化工作空间
```

```
clear;close all;clc;
```

```
%% 绘制摩天轮原有布局
```

```
figure(1);
```

```
D = 54;d = 5;
```

```
R = D / 2;r = 5 / 2;
```

```
theta = linspace(0, 2 * pi, 168);
```

```
x = R * cos(theta);
```

```
y = R * sin(theta);
```

```
plot(x + R, y + R, 'om');
```

```
hold on;
```

```
theta = linspace(0, 2 * pi, 17);
```

```
x = r * cos(theta);
```

```
y = r * sin(theta);
```

```
plot(x + R, y + R, '*b');
```

```
x = linspace(-19, 19, 54);
```

```
plot(x + R, x + R, '.r');
```

```
plot(x + R, -x + R, '.r');
```

```
x = linspace(-27, 27, 54);
```

```
plot(0 + R, x + R, '.r');
```

```
plot(x + R, 0 + R, '.r');
```

```
axis([-5, 60, -5, 60])
```

```
%% 绘制功夫熊猫原有布局
```

```

figure(2);
data = xlsread(' ./ data.xlsx ');
[height, width] = size(data);
x = zeros(1, []);
y = zeros(1, []);
x_i = 1;
y_i = 1;
for j = 1:height
    for i = 1:width
        if data(height - j + 1, i) == 1
            x(x_i) = i;
            y(y_i) = j;
            x_i = x_i + 1;
            y_i = y_i + 1;
        end
    end
end
plot(x, y, ' .r ');

```

% 绘制CUMT原有布局

```

figure(3);
theta = linspace(pi/2, 3/2 * pi, 69);
x = 22 + 22 * cos(theta);
y = 22 + 22 * sin(theta);
plot(x, y, ' .b ');
hold on;
grid on;
theta = linspace(pi, 2 * pi, 24);
x = 43 + 11 * cos(theta);
y = 11 + 11 * sin(theta);

```

```
plot(x, y, '.r');  
plot(32, linspace(11, 44, 33), '.m');  
plot(54, linspace(11, 44, 33), '.m');  
plot(64, linspace(0, 43, 44), '.c');  
plot(114, linspace(0, 43, 44), '.c');  
x = linspace(64, 88.5, 50);  
y = -1.76 * x + 157;  
plot(x, y, '.b');  
x = linspace(89, 114, 50);  
y = 1.76 * x + -156;  
plot(x, y, '.b');  
plot(134, linspace(0, 44, 44), '.r');  
plot(linspace(123, 145, 11), 44, '.m');
```