



中国研究生创新实践系列大赛  
“华为杯”第十六届中国研究生  
数学建模竞赛

学 校 上海交通大学

---

参赛队号 19102480003

---

1.李增

---

队员姓名 2.张许佳

---

3.查伊倩

---

**中国研究生创新实践系列大赛**  
**“华为杯”第十六届中国研究生**  
**数学建模竞赛**

题 目      基于多目标遗传算法的飞行器航迹规划

**摘      要:**

本题目研究智能飞行器在系统定位精度限制下的航迹快速规划问题。考虑多种约束条件下的分配优化方案：

- (1) 仅考虑误差累积，更新规则
- (2) 在(1)的基础上，考虑飞行器最小转弯半径
- (3) 在(1)的基础上，考虑校正点的校正成功率

结果如下：

**问题一：**在不考虑飞行最小转弯半径与校正失败率的前提下，我们对分别含有 610 个校正点与 324 个校正点的飞行空间进行多目标多约束条件下的航迹规划。其中，对于附件 1，即含有 610 个校正点的数据集，航迹校正次数为 10 次，航迹长度为 106331m，对于附件 2，即含有 324 个校正点的数据集，航迹校正次数为 12 次，航迹长度为 109342m。已知起点到终点的直线距离为 100465m，两次飞行路径校正次数较小，航迹长度相比于直线距离仅提高了 5.83%与 8.92%，因此航迹规划合理。

**问题二：**在问题一的基础上考虑飞行器的最小转弯半径。在本问题的讨论上，相邻校正点之间的飞行路程增加，进而影响误差累计情况。由于最小航迹长度是考虑的目标之一，相邻校正点之间的飞行路径为飞行器首先以最小半径沿圆弧运行，在方向指向下一个校正点的时候沿直线飞行。我们求解到，对于附件 1，即含有 610 个校正点的数据集，航迹校正次数为 9 次，航迹长度为 115023m，对于附件 2，即含有 324 个校正点的数据集，航迹校正次数为 14 次，航迹长度为 115481m。因为问题二增加的约束条件，附件一的校正次数相比于问题一减少了 1 次，航迹相长度比问题一增加了 8692m，提高了 8.14%。附件一的校正次数相比于问题一增加了 2 次，航迹相长度比问题一增加了 6139m，提高了 5.61%。新增的约束条件的限制对目标有一定影响，但是差别不大，我们认为是满意的解。

**问题三：**在问题三在问题一的基础上考虑校正成功率。本问题我们基础约束条件的可靠度，即满足约束条件的概率大于一个阈值是，我们认为是满足约束条件的。在该基础上，尽可

能的地最小化校正次数与航迹长度。因为时间的关系，我们未能及时优化算法复杂度，我们仅得到附件一的校正次数为 **12次**。由于附件 2 校正点较为离散，搜索时间长，我们暂时未能得到最优化结果。

本文亮点与创新点：

- (1) 针对于原始数据，我们利用规则引擎，尽可能减小了搜索解空间，并且不丢失最优解存在区域，规则一是飞行器飞行方向在 AB 线上的投影只能指向 B，以防止飞行器为了局部最优路径长度而逆向飞行，符合生活常识也有良好的实验效果，规则二是只取了 AB 周围一个管道内的校正点作为搜索空间，在数据集中的校正点分布相对均匀、分布范围较广的情况下有效地提高了搜索速度，规则三是针对于规则一、二建立起的待选区域中的每个校正点，建立半球邻近胞体模型，把航迹上的相邻校正点限制在较短距离内，有目的地去搜索，也大大加快了搜索速度，这三个规则足够找到一定数量的初代可行解，甚至一定程度上也是较优解；
- (2) 针对于传统的遗传算法，我们针对于变异和适应度计算进行了改进，针对于变异，我们在试验过程中，发现传统的遗传算法在此问题中进化较慢，为了一定程度上改善这个问题，我们改进了变异思路，将传统中的单位变异改成单位状态转移，不仅将选中的变异位状态变异，并且在它的邻近区域选择另外一个点代替它之前的状态，使得变异后的个体有更大几率是可行解。针对于适应度计算，我们采用了多目标函数加权和评价方法，把约束条件和多目标函数的各自惩罚系数联系起来，实现了群体最优；
- (3) 针对于问题 2，我们建立了一个稳定的强规则，即文中所描述的环形管道规则，并且构造出了飞行器在两个校正点之间的最短路径模型，即先沿最小半径圆弧再直线飞向目标点的路径，在求解路径长度、飞行方向的过程中运用了较多的空间几何和平面几何理论，有效性较好；
- (4) 针对于问题 3，我们在问题 1 的基础上构建了机会约束规划模型，即在可信度的范围之外，给予多目标函数各自一定的惩罚系数，以体现较小机会下的惩罚。

**关键词：**航迹规划、遗传算法、多目标函数加权和评价、四元数、机会约束规划、规则引擎

# 1 问题重述

## 1.1 问题背景

复杂环境下航迹快速规划是智能飞行器控制的一个重要课题。由于系统结构限制，这类飞行器的定位系统无法对自身进行精准定位，一旦定位误差积累到一定程度可能导致任务失败。因此，在飞行过程中对定位误差进行校正是智能飞行器航迹规划中一项重要任务。本题目研究智能飞行器在系统定位精度限制下的航迹快速规划问题。

## 1.2 相关约束

假设飞行器的飞行区域如图 1 所示，出发点为 A 点，目的地为 B 点。其航迹约束如下：

- 1) 飞行器在空间飞行过程中需要实时定位，其定位误差包括垂直误差和水平误差。飞行器每飞行 1m，垂直误差和水平误差将各增加 $\delta$ 个专用单位，以下简称单位。到达终点时垂直误差和水平误差均应小于 $\theta$ 个单位，并且为简化问题，假设当垂直误差和水平误差均小于 $\theta$ 个单位时，飞行器仍能够按照规划路径飞行。
- 2) 飞行器在飞行过程中需要对定位误差进行校正。飞行区域中存在一些安全位置（称之为校正点）可用于误差校正，当飞行器到达校正点即能够根据该位置的误差校正类型进行误差校正。校正垂直和水平误差的位置可根据地形在航迹规划前确定（如图 1 为某条航迹的示意图，黄色的点为水平误差校正点，蓝色的点为垂直误差校正点，出发点为 A 点，目的地为 B 点，黑色曲线代表一条航迹）。可校正的飞行区域分布位置依赖于地形，无统一规律。若垂直误差、水平误差都能得到及时校正，则飞行器可以按照预定航线飞行，通过若干个校正点进行误差校正后最终到达目的地。
- 3) 在出发地 A 点，飞行器的垂直和水平误差均为 0。
- 4) 飞行器在垂直误差校正点进行垂直误差校正后，其垂直误差将变为 0，水平误差保持不变。
- 5) 飞行器在水平误差校正点进行水平误差校正后，其水平误差将变为 0，垂直误差保持不变。
- 6) 当飞行器的垂直误差不大于 $\alpha_1$ 个单位，水平误差不大于 $\alpha_2$ 个单位时才能进行垂直误差校正。
- 7) 当飞行器的垂直误差不大于 $\beta_1$ 个单位，水平误差不大于 $\beta_2$ 个单位时才能进行水平误差校正。
- 8) 飞行器在转弯时受到结构和控制系统的限制，无法完成即时转弯（飞行器前进方向无法突然改变），假设飞行器的最小转弯半径为 200m。

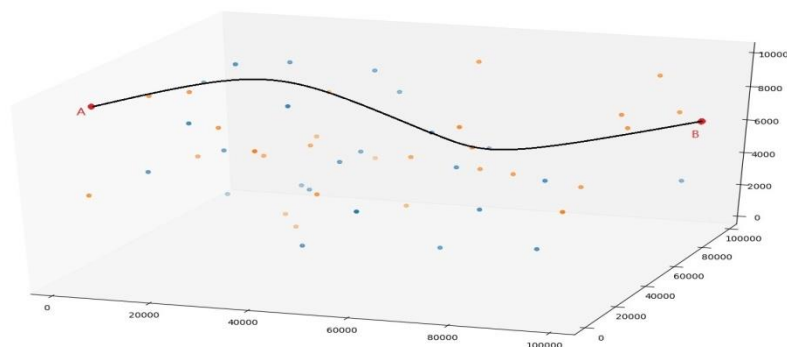


图 1 飞行器航迹规划区域示意图

### 1.3 需要解决的问题

问题 1:

针对附件 1 和附件 2 中的数据分别规划满足条件 (1)~(7) 时飞行器的航迹, 并且综合考虑以下优化目标:

(A) 航迹长度尽可能小; (B) 经过校正区域进行校正的次数尽可能少。

并讨论算法的有效性和复杂度。

其中附件 1 数据的参数为:

$\alpha_1=25$ ,  $\alpha_2=15$ ,  $\beta_1=20$ ,  $\beta_2=25$ ,  $\theta=30$ ,  $\delta=0.001$

附件 2 中数据的参数为:

$\alpha_1=20$ ,  $\alpha_2=10$ ,  $\beta_1=15$ ,  $\beta_2=20$ ,  $\theta=20$ ,  $\delta=0.001$

问题 2:

针对附件 1 和附件 2 中的数据 (参数与第一问相同) 分别规划满足条件 (1)~(8) 时飞行器的航迹, 并且综合考虑以下优化目标:

(A) 航迹长度尽可能小; (B) 经过校正区域进行校正的次数尽可能少。

讨论算法的有效性和复杂度。

问题 3:

飞行器的飞行环境可能随时间动态变化, 虽然校正点在飞行前已经确定, 但飞行器在部分校正点进行误差校正时存在无法达到理想校正的情况 (即将某个误差精确校正为 0), 例如天气等不可控因素导致飞行器到达校正点也无法进行理想的误差校正。现假设飞行器在部分校正点 (附件 1 和附件 2 中 F 列标记为 “1” 的数据) 能够成功将某个误差校正为 0 的概率是 80%, 如果校正失败, 则校正后的剩余误差为  $\min(\text{error}, 5)$  个单位

(其中  $\text{error}$  为校正前误差,  $\min$  为取小函数), 并且假设飞行器到达该校正点时即可知道在该点处是否能够校正成功, 但不论校正成功与否, 均不能改变规划路径。请针对此情况重新规划问题 1 所要求的航迹, 并要求成功到达终点的概率尽可能大。

## 2 模型假设

根据题目要求, 提出如下假设:

问题一假设：

- (1) 飞行器在相邻校正点之间直线飞行；
- (2) 飞行器总体飞行方向是从 A 飞向 B；
- (3) 飞行器可以向任意方向转弯，不考虑前后飞行方向的角度变化差异；

解释：

- (1) 飞行器的目标之一是航迹长度最短，在确定相邻校正点的情况下，两点之间直线距离最短，因此采用直线飞行；
- (2) 在飞行过程中，飞行器的目标始终为终点 B 点，为保证目标为航迹长度最短，飞行器的大致方向始终为朝着 B 点飞行，即不掉头；
- (3) 问题一中未涉及对转弯的要求，为了简化模型，并考虑到目标函数的优化，不考虑转弯限制；

问题二假设：

- (1) 飞行器总体飞行方向是从 A 飞向 B；

问题三假设：

- (1) 飞行器在相邻校正点之间直线飞行；
- (2) 飞行器总体飞行方向是从 A 飞向 B；
- (3) 飞行器可以向任意方向转弯，不考虑前后飞行方向的角度变化差异；
- (4) 每个问题校正点被正确校正的概率是独立同分布的；

解释：

(4) 问题三中提到，无论在某校正点是否校正成功，均不影响航迹，由此不影响之后校正点被选择以及校正成功的概率。

### 3 符号说明

符号	含义
$m$	校正点总个数
$X_i$	第 <i>i</i> 个校正点, $i = 1, 2, \dots, m$
$g_j$	种群中第 <i>j</i> 个染色体
$r_k$	随机数表, $k = 1, 2, \dots, 500$
$x_i$	第 <i>i</i> 个校正点的 <i>x</i> 轴坐标值
$y_i$	第 <i>i</i> 个校正点的 <i>y</i> 轴坐标值
$z_i$	第 <i>i</i> 个校正点的 <i>z</i> 轴坐标值
$\vec{Z}$	目标向量
$p_1$	惩罚参数 1
$p_2$	惩罚参数 2
$r_i$	随机数表, $i = 1, 2, \dots, m$
$eval$	评价函数
$p_k$	选择率
$q_k$	累计选择概率

$gen$	代数
$popSize$	种群大小
$maxGen$	迭代次数
$p_c$	交叉率
$p_m$	变异率
$\alpha_1$	垂直误差校正点的垂直误差条件
$\alpha_2$	垂直误差校正点的水平误差条件
$\beta_1$	水平误差校正点的垂直误差条件
$\beta_2$	水平误差校正点的水平误差条件
$\theta$	飞行器飞行过程中的水平和垂直误差条件
$\delta$	飞行器飞行 1 米垂直误差和水平误差增加的单位
$z_1$	航迹通过的校正点个数
$z_2$	飞行器航迹长度

## 4 问题 1 模型求解及算法设计

### 4.1 问题分析

问题 1 中为了使飞行器的航迹长度尽可能短，进行校正的次数尽可能少，即飞行器要在满足约束条件的情况下，解决一个多目标优化的问题。我们以最小化总校正次数和最小化飞行器轨迹长度为目标函数，求解问题 1。需要满足的约束条件如下：

(1) 飞行器按规划路径飞行条件：飞行器在空中实时定位，假设当垂直误差和水平误差均小于 $\theta$ 个单位时，飞行器仍能够按照规划路径飞行。我们在飞行器到达某校正点后，进行飞行约束条件的判断，不在飞行过程中判断筛选。

(2) 飞行器误差增加方式：飞行器每飞行 1m，垂直误差和水平误差将各增加 $\delta$ 个专用单位；

(3) 在出发地 A 点，飞行器的垂直和水平误差均为 0；

(4) 飞行器进行误差校正的方式：飞行器垂直误差校正后，其垂直误差变为 0，水平误差保持不变，飞行器进行水平误差校正后，其水平误差变为 0，垂直误差保持不变；

(5) 飞行器进行校正的条件：当垂直误差不大于 $\alpha_1$ 个单位，水平误差不大于 $\alpha_2$ 个单位时，飞行器才能进行垂直误差校正，当垂直误差不大于 $\beta_1$ 个单位，水平误差不大于 $\beta_1$ 个单位时，飞行器才能进行水平误差校正；

(6) 飞行器到达终点 B 点时需要满足垂直和水平距离均小于 $\theta$ 个单位

如下图所示，作一条连接 A 点和 B 点的直线，并将所有校正点投影到这条直线上，得到相同数目的校正点的投影点，由于航迹空间较大，投影点几乎不可能重合，根据投影点距离 A 点的距离对校正点进行排序，出于路径尽量短的原因，假设飞行器只能从排序后离 A 点较近的点飞向投影离 A 点较远的点，而不能反向飞行。

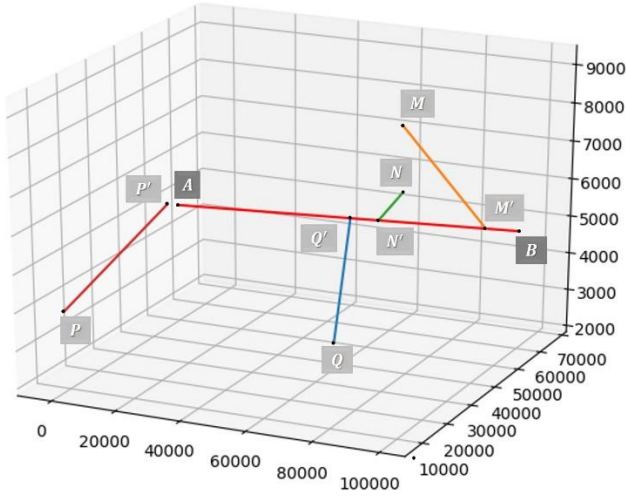


图 2 投影示意图

此问题是一个组合优化问题，而且梯度信息难以求得，加上搜索空间较大，我们基于遗传算法框架建立求解问题的模型。

由于搜索空间较大，在建立初代种群的时候，我们利用约束条件建立了邻近胞体模型以增加初代种群可行解的数量。

基于多个目标条件，建立了多目标函数加权和评价方法。

此外还建立了轮序模型、管道模型来进一步缩小搜索空间。

#### 4.1.1 数据定义与初始化

##### (1) 常量

$chromosome\_length$ ——个体染色体长度，取决于 AB 之间校正点的数目

$popSize$ ——种群大小，本题取值为 100

$maxGen$ ——最大迭代次数，本题取值为 500

$p_c$ ——交叉率，本题取值为 0.25

$p_m$ ——变异率，本题取值为 0.02

$\theta$ ——飞行器按照规划路径飞行的水平和垂直误差条件，小于此参数，即可按照规划路径飞行，附件 1 中为 30，附件 2 中为 20

$\delta$ ——飞行器飞行 1 米垂直误差和水平误差增加的单位，附件 1 和 2 中都为 0.001

$\alpha_1$ ——垂直误差校正的飞行器垂直误差条件，附件 1 中为 25，附件 2 中为 20

$\alpha_2$ ——垂直误差校正的飞行器水平误差条件，附件 1 中为 15，附件 2 中为 10

$\beta_1$ ——水平误差校正的飞行器垂直误差条件，附件 1 中为 20，附件 2 中为 15

$\beta_2$ ——水平误差校正的飞行器水平误差条件，附件 1 中为 25，附件 2 中为 20

$g_j$ ——种群中第  $j$  个染色体

$x_i$ ——第  $i$  个校正点的  $x$  轴坐标值

$y_i$ ——第  $i$  个校正点的  $y$  轴坐标值

$z_i$ ——第  $i$  个校正点的  $z$  轴坐标值



$type_i$ ——第 $i$ 个校正点的校正点类型

$p_1$ ——惩罚参数

$p_2$ ——惩罚参数 2

(2) 变量

$gen$ ——迭代数

$z_1$ ——经过校正区域进行校正次数

$z_2$ ——飞行器轨迹长度

$eval$ ——评价函数

$p_j$ ——选择率

$q_j$ ——累计选择概率

(3) 中间变量

$d_j$ —— $d_j$ 为染色体 $v_j$ 对应校正点与 A 点的距离, 由它的范围我们可以求出

## 4.2 模型建立及准备

### 4.2.1 数据预处理

(1) 轮序模型:

将原始校正点数据投影到 AB 线段上, 排序后剔除 AB 线段以外的校正点, 航迹所要经过的校正点只在 AB 之间的校正点中选择, 以保证飞行器不会逆向飞行, 在一定程度上缩小了解空间。

(2) 管道模型:

在经历多次尝试之后, 我们发现, AB 线段附近的校正点已经足够于建立满足条件的航迹, 且相对于距离 AB 线段距离较远的校正点航迹, 总航迹长度要更短, 经过的校正点个数也相对较少。

先建立 AB 线段的方程, 求解出所有校正点与 AB 线段的垂直距离, 以最大距离的三分之一建立管道, 将航迹校正点的搜索空间限制在管道以内, 但同时变异、交叉不对其限制范围, 保证多样性。

### 4.2.2 模型建立

(1) 目标函数: 由问题 1 可知, 为了使飞行器的航迹长度尽可能小, 并且在此基础上使飞行器经过校正区域进行校正的次数尽可能少, 因此本文以总校正次数和飞行器轨迹长度组成的一个向量为目标函数, 目标函数如下:

$$\vec{Z} = (f_1(c), f_2(x, y, z)) \quad (4-1)$$

$$f_1(c) = \sum_{i=1}^m c_i \quad (4-2)$$

其中， $c_i$ 为第 $i$ 个校正点所对应的决策变量，若校正点被选择为航迹，则 $c_i=1$ ，否则 $c_i=0$ ； $f_1(c)$ 为校正次数求和函数。

$$f_2(x, y, z) = \sum_{i=1}^m \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad (4-3)$$

其中， $x_i$ 、 $y_i$ 、 $z_i$ 分别为第 $i$ 个校正点所在三维空间对应点的 $x$ 轴坐标值、 $y$ 轴坐标值和 $z$ 轴坐标值， $f_2(x, y, z)$ 为飞行器轨迹长度求和函数。

(2) 飞行器垂直误差和水平误差的计算方法：

$$e_{v_i} = e_{v_{i-1}} + \delta * \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad (4-4)$$

$$e_{h_i} = e_{h_{i-1}} + \delta * \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad (4-5)$$

(3) 飞行器按照规划路径飞行的条件：飞行器在空中实时定位，假设当垂直误差和水平误差均小于 $\theta$ 个单位时，飞行器仍能够按照规划路径飞行。即

$$e_{v_i} < \theta \quad (4-6)$$

$$e_{h_i} < \theta \quad (4-7)$$

其中， $e_{v_i}$ 为垂直误差， $e_{h_i}$ 为水平误差。

(4) 在出发地 A 点，飞行器的垂直和水平误差均为 0：

$$e_{v_0} = 0 \quad (4-8)$$

$$e_{h_0} = 0 \quad (4-9)$$

(4) 飞行器进行误差校正的方式：飞行器垂直误差校正后，其垂直误差将变为 0，水平误差保持不变，即

$$e_{v_i} = 0 \quad (4-10)$$

$$e_{h_i} = e_{h_{i-1}} \quad (4-11)$$

同理，飞行器进行水平误差校正后，其水平误差将变为 0，垂直误差保持不变，即

$$e_{h_i} = 0 \quad (4-12)$$

$$e_{v_i} = e_{v_{i-1}} \quad (4-13)$$

(5) 飞行器进行校正的条件：当飞行器的垂直误差不大于 $\alpha_1$ 个单位，水平误差不大于 $\alpha_2$ 个单位时才能进行垂直误差校正，即

$$e_{v_i} \leq \alpha_1, e_{h_i} \leq \alpha_2 \quad (4-14)$$

时才能进行垂直误差校正；当飞行器的垂直误差不大于 $\beta_1$ 个单位，水平误差不大于 $\beta_2$ 个单位时才能进行水平误差校正，即

$$e_{v_i} \leq \beta_1, e_{h_i} \leq \beta_2 \quad (4-15)$$

时才能进行水平误差校正。

(6) 特别地，飞行条件的判断在校正之前，即飞行器每到达一个校正点，先判断是否符合约束条件，再根据实际情况做误差校正。

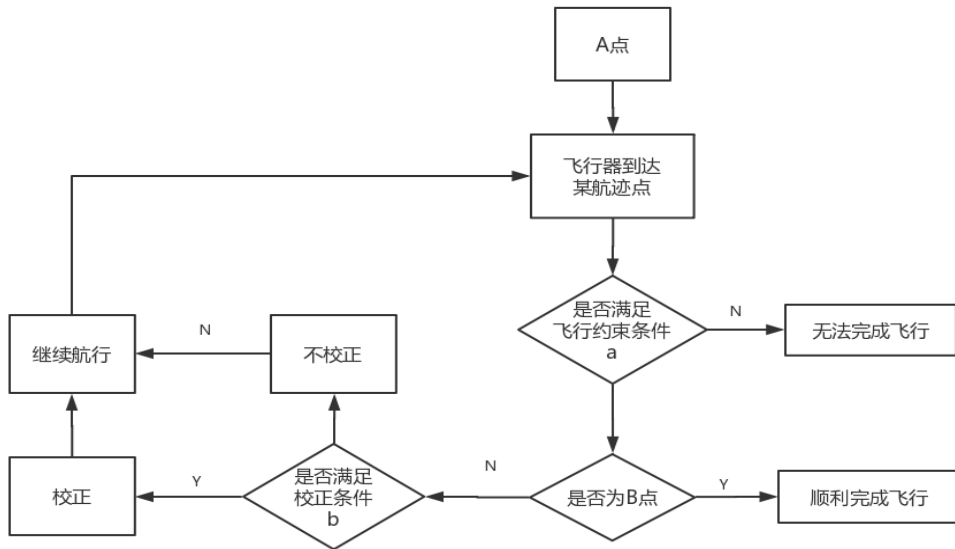


图 3 模型流程图

综上，可以建立问题 1 的数学模型，如 MOD1 所示：

MOD1

$$\min z_1 = f_1(c), z_2 = f_2(x, y, z)$$

$$s. t. \begin{cases} e_{v_i} < \theta, e_{h_i} < \theta, i = 1, 2, \dots, m \\ e_{v_0} = 0, e_{h_0} = 0 \\ e_{v_i} = 0, e_{v_i} = e_{v_{i-1}} \\ e_{h_i} = 0, e_{v_i} = e_{v_{i-1}} \\ e_{v_i} \leq \alpha_1, e_{h_i} \leq \alpha_2 \\ e_{v_i} \leq \beta_1, e_{h_i} \leq \beta_2 \end{cases}$$

### 4.3 模型的求解及分析

飞行器航迹规划问题是一个组合优化的问题，这个问题解决起来较为复杂棘手。二十世纪六七十年代提出的遗传算法可以有效的解决这种复杂组合优化的问题。遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。其优点是原理和操作简单、通用性强、不受限制条件的约束, 且具有隐含并行性和全局解搜索能力, 在组合优化问题中得到广泛应用。一般而言，遗传算法是一种将自然淘汰和遗传现象的机理简单化之后得到的超启发式算法，其中被称为个体的染色体（即研究对象的候选解）的集合能适应外部环境（研究对象的评价函数），并基于下面的规划在每代中生成新个体集合：

- （1）越是适应性强的个体，其生存几率越高（自然选择）；
- （2）以原有个体为基础通过遗传操作生成新个体（遗传现象）。<sup>[1]</sup>

#### 4.3.1 算法设计

遗传算法是一种基于“适者生存”的高度并行、随机和自适应的优化算法, 通过复制、交叉、变异将问题解编码表示的“染色体”群一代代不断进化, 最终收敛到最适应的群体, 从而求得问题的最优解或满意解。

##### STEP1 个体编码设计

我们采用 0 和 1 两个数对校正点进行编码，1 代表实际进行校正的校正点，0 代表实际不进行校正的校正点， $m$  为校正点总个数，基因顺序为投影到 AB 线段上的投影位置顺序。例如 0 1 1 0 0 0 ...1 依次代表  $m$  个校正点是否参与校正的情况：编号为 1 的校正点在航迹中，编号为 2 的校正点不在航迹中，依次类推。

0	1	1	...	...	...	...	0	1	0
---	---	---	-----	-----	-----	-----	---	---	---

染色体长度  $chromosome\_length$  取决于 AB 之间校正点的数目。

##### STEP2 初始种群生成

第一步 作一条连接 A 点和 B 点的直线，并将各个校正点投影到这条直线上，这样，这条直线上存在若干个校正点投影点，投影点几乎不可能重合，由此根据投影点距离 A 点的远近程度对校正点做排序，即产生  $X_1, X_2, \dots, X_i, \dots, X_m$  校正点；

第二步 邻近胞体的建立：

由于可选择校正点较多，解空间太大，并且绝大部分是不可行解，为了减少搜索空间，优化初代基因质量，建立起邻近胞体模型。

遍历所有的待选校正点，计算它与所有后序位的校正点的距离，对于每一个待选校正点建立一个半球邻近胞体，在此胞体内的所有邻近校正点与此胞体中心校正点的距离均在胞体半径以内，其中胞体半径为：

$$r_1 = \max\{\alpha_1, \alpha_2, \beta_1, \beta_2\} \quad (4-14)$$

保存为胞体表格文件，如下图所示：

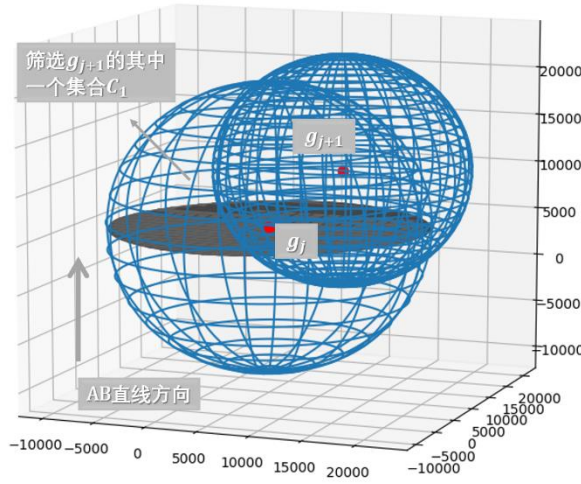


图 4 半球胞体模型

（由于难以画出半球，只能画出球体，实际上待选区域是每个球体内部沿 AB 方向更靠近 B 点的半球内部）

第三步 选取 A 的下一点

把以 A 点为胞体中心的半球胞体内的所有校正点作为待选点，遍历每一个点，根据其 *type\_i* 和到达此点前的水平误差和垂直误差是否满足校正条件，若满足校正条件则将此点作为 A 点的下一个点，作为初代种群中的个体染色体的第一个基因  $g_j$ ，此时  $j = 1$ ，若不满足，则重新选校正点，重复第三步；

第四步 以染色体  $g_j$  对应校正点为胞体中心的半球胞体为搜索空间，作一个半径为

$r_{j+1}'$  和  $r_{j+1}''$  米的球，

$$\begin{cases} e_{v_{j-1}} + \delta * d_j \leq \alpha_1 \\ e_{v_{j-1}} + \delta * d_j \leq \alpha_2 \end{cases} \quad (4-15)$$

$$r_{j+1}' = \sup d_j \quad (4-16)$$

$$\begin{cases} e_{h_{j-1}} + \delta * d_j \leq \beta_1 \\ e_{h_{j-1}} + \delta * d_j \leq \beta_2 \end{cases} \quad (4-17)$$

$$r_{j+1}'' = \sup d_j \quad (4-18)$$

其中,  $d_j$ 为染色体 $g_{j+1}$ 对应校正点与 $g_j$ 点的距离, 为一个中间变量。

将半径为 $r_{j+1}'$ 的球与 $g_j$ 中心半球胞体的交集的校正点作为集合 $C_1$ ; 将半径为 $r_{j+1}''$ 的球与 $g_j$ 中心半球胞体的交集的校正点作为集合 $C_2$ 。在 $C_1 \cup C_2$ 集合中随机选取一个元素作为第二个染色体 $g_{j+1}$ , 此时  $j=1$ ;

第五步 若 $g_j$ 对应校正点不为 B, 即航迹没有延伸到终点 B 点, 转第三步;

设定初始种群规模为 $popSize=500$ , 即循环上述第三、四、五步骤 500 次。

### STEP3 单点交叉

当初代建立以后, 或者选择后产生了 $popSize$ 个子代, 即开始进行交叉、变异操作。

令 $r_i$ 服从  $U(0, 1)$  的均与分布, 这样随机产生  $[0, 1]$  区间内的一个随机数 $r_i$ , 令交叉率 $p_c=0.25$ , 若 $r_i < p_c$ , 则个体 $i$ 被选择; 把 $g_i$ 和 $g_{i+1}$ 两条染色体作为父代进行交叉, 随机选择两条染色体的数字相同的一个位置作为断点, 交换断点右端部分, 交叉示意图如图 2 所示。

交叉前

$g_i$

0	1	1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---

$g_{i+1}$

1	0	1	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---



交叉后

$g_i$

0	1	1	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---

$g_{i+1}$

1	0	1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---

图 5 交叉操作示意图

将产生的两个新的染色体加入到上一代种群中作为交叉个体。

#### STEP4 双重变异

变异的步骤如下：令 $r_k$ 服从 $U(0, 1)$ 的均匀分布，这样随机产生 $[0, 1]$ 区间内的一个随机数 $r_k$ ，取 $k$ 为从1到种群数量 $popSize$ 之间的整数，令变异率 $p_M=0.25$ ，若 $r_k < p_M$ ，选择其对应的染色体进行变异，随机选择要进行变异的变异位，将此变异位取反，即将父代个体中对应的校正点是否在航迹中的状态取反。

变异增加了种群的多样性，但由于解空间太大以及约束条件的存在，我们在上述变异之后还采用了另外一种定向变异的方法，以增加变异个体的可行率，操作的步骤如下：

令 $r_k$ 服从 $U(0, 1)$ 的均匀分布，这样随机产生 $[0, 1]$ 区间内的一个随机数 $r_k$ ，取 $k$ 为从1到种群数量 $popSize$ 之间的整数，令变异率 $p_M=0.25$ ，若 $r_k < p_M$ ，选择其对应的染色体进行变异，随机选择要进行变异的变异位1，在以变异位1对应的半球胞体内内再选一个类型和变异位1相同的变异位2，若变异位2无法选取，即在此半球胞体内没有第二个与变异位1类型相同的校正点，则对变异位1的数取反变异（普通变异），若变异位2可以选取，则将变异位1的数取反，变异位2的数变成变异位1变异之前的状态，如表1所示。

变异位1 变异前的状态	变异位	变异位1 变异后的状态	变异位2 变异后的状态
1		0	1
0		1	0

表格 1 变异操作示意表

#### STEP5 多目标函数加权和评价方法

1988年Gen和Cheng提出适应性权重法，该方法利用在各代种群中所获得正向的理想点，通过调整权重使Pareto最优解靠近理想点来进行解的搜索。

步骤如下：

第一步  $z_k^{max}$ ， $z_k^{min}$ 表示第 $k$ 个目标函数的最大值和最小值， $k=1, 2$ ，如下定义。P为可行域解的集合。

$$z_k^{max} = \max_P \{f_1\} \quad (4-19)$$

$$z_k^{min} = \min_P \{f_1\} \quad (4-20)$$

计算个目标函数值的 $z_k$ 。若染色体 $(g_j)$ 不满足染色体约束条件，则使用如下惩罚函数。

$$z_k = [p_1 f_1(g_j), p_2 f_2(g_j)] \quad (4-21)$$

$p_1, p_2$ 为100到500中的随机数。其中，若满足约束条件，则 $z_k$ 不发生改变。

第二步 某一个染色体 $g_j$ 的权重和由下式求得。

$$eval = \sum_{k=1}^2 w_k (f_k - z_k^{min}) = \sum_{k=1}^2 \frac{f_k - z_k^{min}}{z_k^{max} - z_k^{min}} \quad (4-21)$$

其中第 $k$ 个目标函数的适应性权重 $w_k$ 由下式计算。

$$w_k = \frac{1}{z_k^{max} - z_k^{min}} \quad (4-22)$$

用此公式获得各非劣解的评价函数值 $eval(g_j)$ 。

#### STEP6 轮盘赌选择

从父代染色体和新生成的染色体中，按各染色体适应度的高低，选择留存到下一代的 $popSize$ 数目的染色体。下面计算各染色体的选择率 $p_j$ 以及累计选择概率 $q_j$ 如下。

$$F = \sum_{j=1} \frac{1}{eval(g_j)} \quad (4-23)$$

$$p_j = (\frac{1}{eval(g_j)})/F \quad (4-24)$$

$$q_j = \sum_{j=1} p_j \quad (4-25)$$

生成 $[0, 1]$ 区间内的随机数表 $r_k$ ，若 $g_{j-1} < r_k < g_j$ 则个体  $j$  被选择。

轮盘赌选择模型不仅可以让适应度高的个体有更多的几率存活下来，而且也给了适应度低的个体一定的机会，具有良好的选择性和后代多样性。

#### STEP7 终止条件

若满足终止条件 $maxGen \geq gen$ ，则算法停止，否则返回 STEP3。



## 算法流程:

数据预处理:

- 1、读取数据
- 2、投影到 AB 线段上, 计算出次序和距离
- 3、剔除投影在 AB 线段以外的点, 并取离 AB 距离在内部三分之一的点
- 4、利用剩下的点生成以每个节点编号为键值的字典, 字典内容是以键为中心的半球胞体的校正点编号、类型、离键点的距离

生成遗传初代:

While 循环次数小于种群规模:

- 1、第一个点设置为 A 点, 个体染色体 `population[i]` 全置 0
- 2、遍历以 A 为键的值:
  - If 类型为 1, 并且离 A 点距离  $\text{distance} \leq \min(\alpha_1, \alpha_2)$ :  
加入到待选点集 `node_candidate` 中
  - Else if 类型为 0, 并且离 A 点距离  $\text{distance} \leq \min(\beta_1, \beta_2)$ :  
加入到待选点集 `node_candidate` 中
- 3、在 0 到  $\text{len}(\text{node\_candidate})$  之间的一个随机整数  $r$ , 取 `node_candidate` 中第  $r+1$  个点作为 A 点的下一点, 保存到航迹点列表 `node_list` 中
- 4、水平误差  $\text{error\_level} \leftarrow \text{distance} * \delta$   
垂直误差  $\text{error\_vertical} \leftarrow \text{distance} * \delta$   
当前所在校正点 `temp_node`  $\leftarrow$  选中的校正点信息包括 `type`、`distance`、`num`
- 5、if `type==1`:
  - `error_vertical=0`else:
  - `error_level=0`
- 6、while 当前所在校正点不是 B 点:
  - For 遍历当前所在校正点 `temp_node` 为中心的半球胞体:
    - 判断类型与到达待选点之前的误差是否符合校正要求;
    - 若符合要求, 则将此点加入待选点集合
  - If 待选点集合为空:
    - 从 A 点开始重新找航迹
  - Else if 待选点集合中只有一个元素:
    - 选中此点
  - Else:
    - 在 0 到待选点集合元素数目的范围中生成一个随机数  $r$
    - 选中待选点集合中的第  $r+1$  个校正点
    - 根据 `temp_node` 与选中点之间的距离更新误差
    - 根据选中点的类型校正误差
    - `Temp_node`  $\leftarrow$  选中的点
    - 将 `temp_node` 加入校正点列表 `node_list` 中
- For 遍历 `node_list` 中的节点:
  - 将染色体 `population[i]` 中对应位置置 1

交叉:

For 遍历种群中的每一个个体:

    生成一个[0,1]之间的随机数  $r$

    If  $r < \text{交叉概率 } pc$ :

        生成一个[0,染色体长度-1]之间的随机整数  $t$

        将染色体  $\text{population}[i]$  中第  $r+1$  位之后的部分  $\text{population}[i][r:]$  与下一条染色体中第  $r+1$  位之后的部分  $\text{population}[i][r:]$  进行互换,生成两个交叉个体

        将交叉个体加入到种群当中

变异:

For 遍历种群中的每一个个体:

    生成一个[0,1]之间的随机数  $r$

    If  $r < \text{交叉概率 } pm$ :

        生成一个[0,染色体长度-1]之间的随机整数  $t$

        将染色体  $\text{population}[i]$  中第  $r+1$  位取反

        在该位对应校正点的半球胞体中随机选择一个与之类型相同的校正点,将其置为变异位变异之前的状态,生成三个变异个体

        将变异个体加入种群中

计算适应度:

For 遍历种群中的每一个个体:

    判断变量  $F$  设为 True

    For 遍历每个个体中的每一位:

        If 某一位为 1:

            校正点数目+1

            计算距离、误差

            If 不满足校正条件:

$F \leftarrow \text{False}$

            Else:

                进行误差校正

    If 到达 B 点的误差大于  $\theta$ :

$F \leftarrow \text{False}$

    If  $F == \text{False}$ :

        给总长度  $\text{length}$  和总校正点数目  $\text{num\_node}$  乘一个较大的数目

    计算  $\text{length}$ 、 $\text{num\_node}$  中的最大、最小值

    根据多目标函数加权原理, 计算目标函数值  $z_k$

$Z_k \leftarrow 1/z_k$

求种群  $z_k$  的累积和

For 遍历每个  $z_k$ :

$\text{Fitness} \leftarrow z_k / \text{sum}$

选择:

依次计算  $\text{fitness}$  的累积和

生成[0, 1]范围内的随机数列表  $ms$ , 长度为预先设计的种群规模

For 遍历 ms 中的每个数：  
    判断 ms[i] 处于累积和中的哪个位置  
    取出对应位置处的父代个体放入子代种群中  
后续一些处理：  
每一次迭代形成的子代中选取适应度最大的个体  
计算每一代最优个体的误差、长度等

以上过程循环预先设计的最大迭代次数

画图可视化  
导出数据、表格

### 4.3.2 问题 1 求解结果

基于上述遗传算法、多目标函数加权和评价、邻近胞体、轮序、管道融合模型，我们对问题一进行了求解。

对于数据集 1，选取了 100 个体规模的种群，最大迭代次数设为 500 次，在飞行器在两校正点的路径为直线的假设条件下，综合考虑航迹长度和经过校正点的数目，求解出来的航迹结果如下，其中 3d 图中点的坐标为校正点的空间坐标：

校正点编号	校正前垂直误差	校正前水平误差	校正点类型
0	0	0.000000	0.000000 出发点A
1	303	17.674219	17.674219 01
2	64	22.805201	5.130982 11
3	607	18.455601	23.586583 01
4	170	22.048521	3.592920 11
5	282	7.597400	11.190321 01
6	33	10.067317	2.469917 11
7	315	15.460078	17.929995 01
8	403	23.486067	8.025989 11
9	594	11.029112	19.055101 01
10	501	22.226062	11.196950 11
11	612	8.490015	19.686965 终点B

表格 2 附件 1 结果表

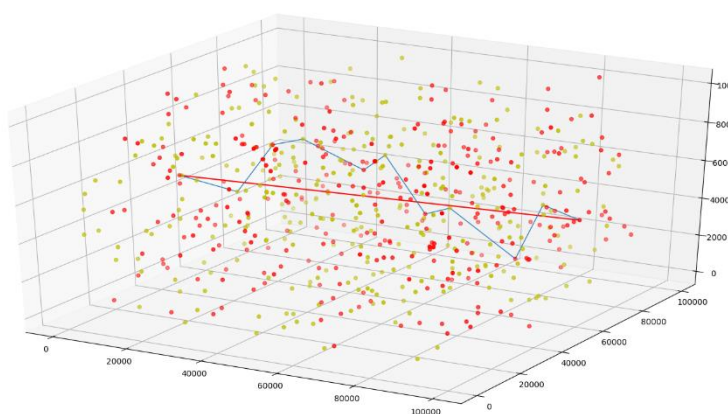


图 6 附件 1 路径结果图

对于数据集 2，选取了 100 个体规模的种群，最大迭代次数设为 500 次，在飞行器在两校正点的路径为直线的假设条件下，综合考虑航迹长度和经过校正点的数目，求解出来的航迹结果如下，其中 3d 图中的每个点的坐标为校正的空间坐标：

校正点编号	校正前垂直误差	校正前水平误差	校正点类型
0	0	0.000000	出发点
1	163	13.287902	0'
2	114	18.622050	1'
3	8	13.921990	0'
4	309	19.446310	1'
5	305	5.968718	0'
6	123	15.173117	1'
7	45	10.006157	0'
8	160	17.491291	1'
9	92	5.776166	0'
10	93	15.260881	1'
11	61	9.834210	0'
12	292	16.388124	1'
13	326	6.960509	终点

表格 3 附件 2 结果表

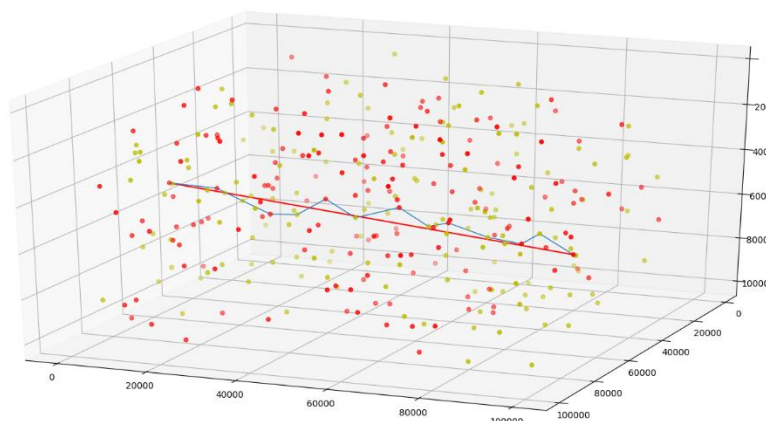


图 7 附件 2 路径结果图

### 4.3.3 问题 1 结果分析

我们对附件一的数据进行分析，进行以下两组的对比，分别生成 10 次最终解，画图如下。

- (1) 种群大小为 100，选择子代的规则采取排序法，与种群大小为 100，选择子代的规则采取轮盘赌选择法；
- (2) 种群大小为 100，选择子代的规则采用排序法，与种群大小为 200，选择子代的规则采用排序法；

得到以下图表，其中图 5-8 的点坐标为校正点的空间坐标，图 9，图 11 的纵坐标为校正次数，横坐标为迭代次数，图 10，图 12 的纵坐标为航迹长度，横坐标为迭代次数。

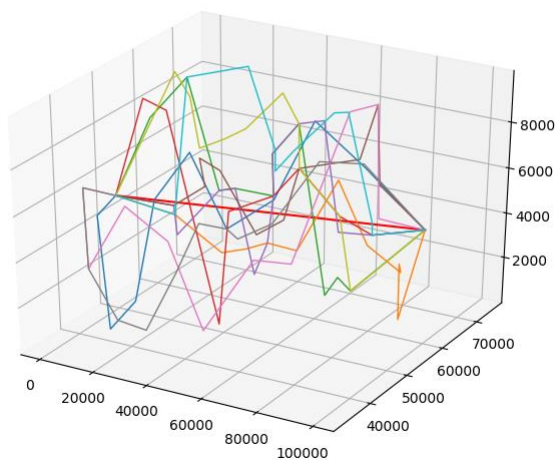


图 8 种群大小为 100，排序法

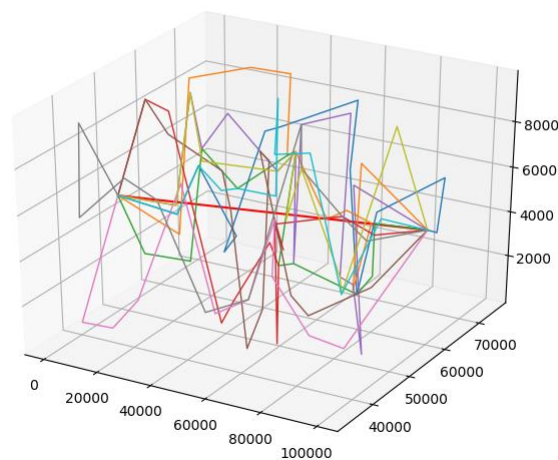


图 9 种群大小为 100，轮盘赌选择法

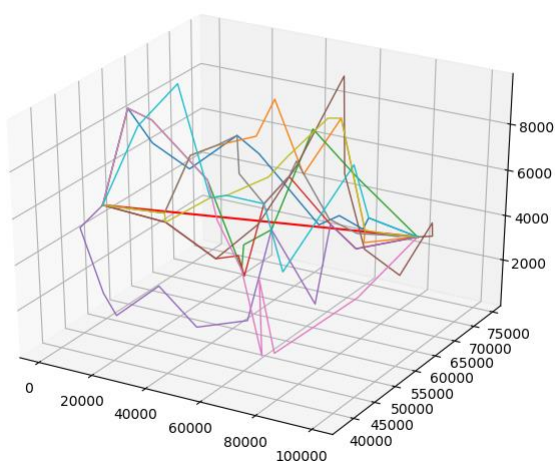


图 10 种群大小为 200，排序法

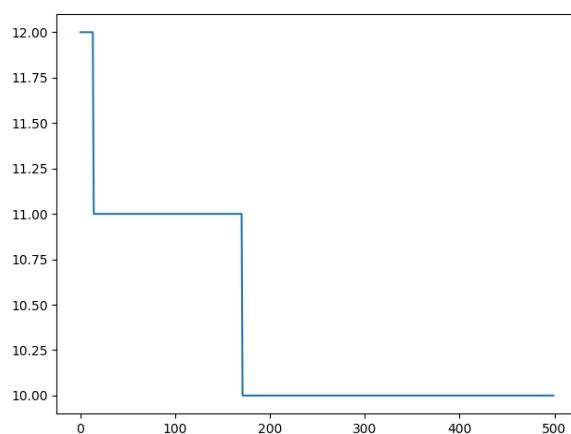


图 11 种群大小为 100，排序法的典型情况，图中纵坐标为校正次数

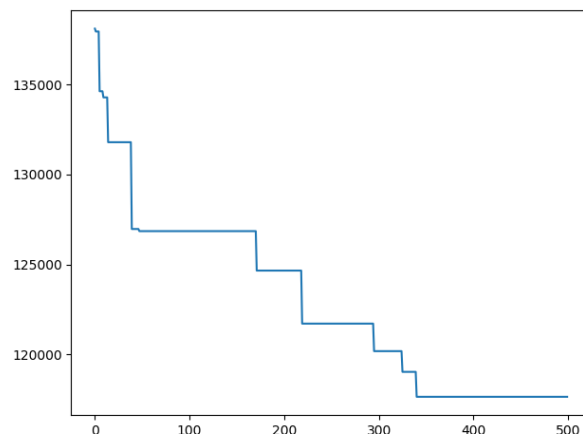


图 12 种群大小为 100，排序法的典型情况，图中纵坐标为航迹长度

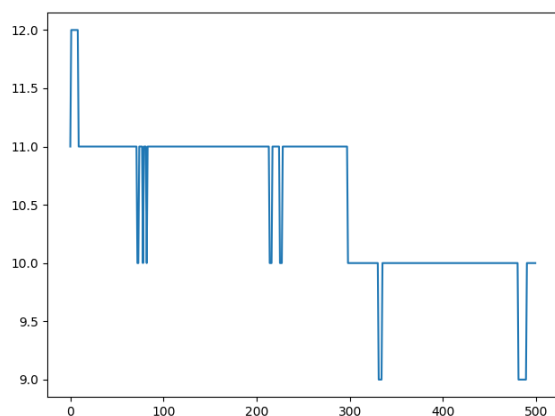


图 13 种群大小为 100，轮盘赌选择法的典型情况，图中纵坐标为校正次数

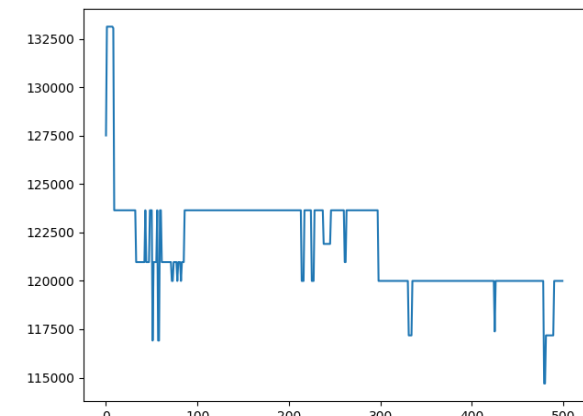


图 14 种群大小为 100，轮盘赌选择法的典型情况，图中纵坐标为航迹长度

根据图 4-5 的数据散点图不难发现，水平校正点与垂直校正点随机分布在空间中，且数量相近，由此，满足条件的最优解不一定限制在一条航迹，多种差异较大的航迹均有较小的目标函数值。图 6-8 的结果验证了该想法。

有效性分析：

对于第一种比较, 可以看到, 总体上讲, 排序法得到的最短航迹长度小于轮盘赌选择法得到的最短航迹长度。轮盘赌选择法是遗传算法中常用的选择方法, 按照一条染色体适应函数的大小占有所有染色体的适应函数的和的方式选择子代, 而根据该课题的实际情况, 每个个体的基因总数远大于航迹经历的校正点, 变异与交叉产生的子代往往不满足约束条件。这容易引起结果的早熟, 即适应函数更优的父代有较强的遗传的能力, 导致子代有大量的类似于该父代的染色体, 甚至是若干条与该父代相同的染色体, 大大减小了多样性。实验表明, 在迭代 50 次之后, 互不相同的染色体仅占 1/3。排序法克服了这一缺陷, 通过排序将尽可能多差异较大的染色体选入下一代, 每个染色体仅可能被选择一次, 弱化了适应函数优异的父代的遗传能力, 避免早熟, 陷入局部最优。

图 9-10 为排序法选择的典型情况, 随着迭代次数的增加, 两个目标函数单调趋于最优解。图 11-12 为轮盘赌选择的典型情况, 迭代过程中, 通过变异交叉生成了更优的染色体, 但是由于已有较为优异的父代的强大的遗传能力, 最优的染色体一定概率未能被选入子代中进行遗传, 表现为图像中的曲线上升情况。

对于第二种比较, 可以看到, 种群规模增大与否, 结果没有明显的优劣关系。说明种群规模对最优结果影响较小, 或者说, 最优结果对种群规模收敛较快。这一结果主要是因为, 满足条件的最优解不一定限制在一条航迹, 多种差异较大的航迹均有较小的目标函数值。增加种群数量, 在初代生成以及子代遗传选择中增加了多样性, 但是较小种群数量的情况下, 配合排序法做子代选择, 依旧能找到最优解。同样的, 当迭代次数大于一定量时 (比如 500), 迭代次数的增加, 对结果影响较小。这说明了结果能较快的收敛。

第一种比较, 体现了排序法选择优于轮盘赌选择法, 由此我们最终采用更优的排序法。第二种比较, 体现了最优解能在较小种群规模下较快的收敛, 但是找到的很大概率上并不是局部最优解。原因如下:

- (1) 由数据得, AB 点之间距离约为 100465m, 最终解的航迹长度一般在 115000m 以下校正次数在 11 以下。这在一定程度体现了最终解的优越性;
- (2) 变异算法采用双重变异法, 第一重变异为遗传算法中针对 0-1 染色体常用的反向变异, 即取反。由于基因中参与航迹的校正点占基因总数较小, 取反变异中的大部分为将 0 变为 1 的变异, 生成满足约束条件的可能性较小。由此, 该种变异虽然增强了多样性, 但是生成的子代被选入下一代的概率较小。由此, 我们引入了第二重变异, 以一定概率选取航迹中的校正点, 在其邻域中选择同种类型的校正点进行变异。该变异方法由于子代与父代差异较小, 容易生成满足约束条件的更优解。该方法类似于混合遗传算法中的爬山法。不同的是, 混合遗传算法中, 爬山法在选择前, 即交叉变异结束后, 对每一个新生成的子代进行局部最优的计算, 以增加所有可行的解的遗传能力, 尤其希望找到全局最优解。而本次变异采用的第二重变异方法, 在遗传变异中进行计算, 且仅以一定概率增加了生成子代的可行性, 而不是直接找到局部最优解。这是因为本课题中最优解不是单一的, 在选择子代前进行局部最优解的寻找意义不大, 算法本身能够较快的收敛, 并且有效的找到最优解。另外, 本课题中不涉及梯度, 利用爬山法进行局部优化时间复杂度较大。

## 5 问题 2 模型求解及算法设计

### 5.1 问题分析

问题 2 是在问题 1 的基础上增加了一个飞行器在转弯时受到结构和控制系统的限制，即飞行器无法完成即时转弯(飞行器前进方向无法突然改变)，假设飞行器的最小转弯半径为 200m 这样的约束条件，目标与问题 1 相同，是使飞行器的航迹长度尽可能小，并且在此基础上使飞行器经过校正区域进行校正的次数尽可能少，即以最小化总校正次数和最小化飞行器轨迹长度为目标函数，求解问题 2。问题 2 首先也要满足问题 1 的五个约束条件：飞行器按规划路径飞行条件、飞行器误差增加方式、在 A 点的条件、飞行器进行误差校正的方式、飞行器进行校正的条件。

### 5.2 模型的建立与准备

#### 5.2.1 数据定义与初始化

(1) 变量：

$v(vx, vy, vz)$ ——飞行器航行的方向

$x1, y1, z1$ ——已知航迹末端的空间坐标

$x2, y2, z2$ ——待选校正点的空间坐标

(2) 常量

$r$ ——飞行器最小转弯半径

#### 5.2.2 模型建立

问题 2 所满足的约束条件同问题 1：

(1) 飞行器按规划路径飞行条件：

$$e_{v_i} < \theta, e_{h_i} < \theta, i = 1, 2, \dots, m$$

(2) 飞行器误差增加方式：

$$e_{v_i} = e_{v_{i-1}} + \delta * \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2},$$

$$e_{h_i} = e_{h_{i-1}} + \delta * \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}, i = 1, 2, \dots, m$$

(3) 在 A 点的条件：在出发地 A 点，飞行器的垂直和水平误差均为 0：

$$e_{v_0} = 0, e_{h_0} = 0$$

(4) 飞行器进行误差校正的方式：经过垂直误差校正后，

$$e_{v_i} = 0, e_{h_i} = e_{h_{i-1}}, i = 1, 2, \dots, m$$

经过水平误差校正后，

$$e_{h_i} = 0, e_{v_i} = e_{v_{i-1}}, i = 1, 2, \dots, m$$

(5) 飞行器进行校正的条件：

进行垂直误差校正：

$$e_{v_i} \leq \alpha_1, e_{h_i} \leq \alpha_2, i = 1, 2, \dots, m$$

进行水平误差校正：

$$e_{v_i} \leq \beta_1, e_{h_i} \leq \beta_2, i = 1, 2, \dots, m$$

(6) 飞行器转弯限制：飞行器在转弯时受到结构和控制系统的限制，无法完成即时转弯（飞行器前进方向无法突然改变），假设飞行器的最小转弯半径为 200m：

$$r \geq 200$$

问题 2 目标函数与问题 1 相同：

以经过校正区域进行校正次数和飞行器轨迹长度组成的一个向量为目标函数，目标函数如下：

$$\vec{Z} = (f_1(c), f_2(x, y, z))$$

$$f_1(c) = \sum_{i=1}^m c_i$$

$$f_2(x, y, z) = \sum_{i=1}^m \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}$$

综上，可以建立问题 2 的数学模型，见 MOD2

MOD2

$$\min z_1 = f_1(c), z_2 = f_2(x, y, z)$$



$$s.t. \begin{cases} e_{v_i} < \theta, e_{h_i} < \theta, i = 1, 2, \dots, m \\ e_{v_0} = 0, e_{h_0} = 0 \\ e_{v_i} = 0, e_{v_i} = e_{v_{i-1}} \\ e_{h_i} = 0, e_{v_i} = e_{v_{i-1}} \\ e_{v_i} \leq \alpha_1, e_{h_i} \leq \alpha_2 \\ e_{v_i} \leq \beta_1, e_{h_i} \leq \beta_2 \\ r \geq 200 \end{cases}$$

### 环形管道模型:

由于飞行器有最小转弯半径  $r$  的限制，在问题一的基础上，可行解搜索空间进一步减小。

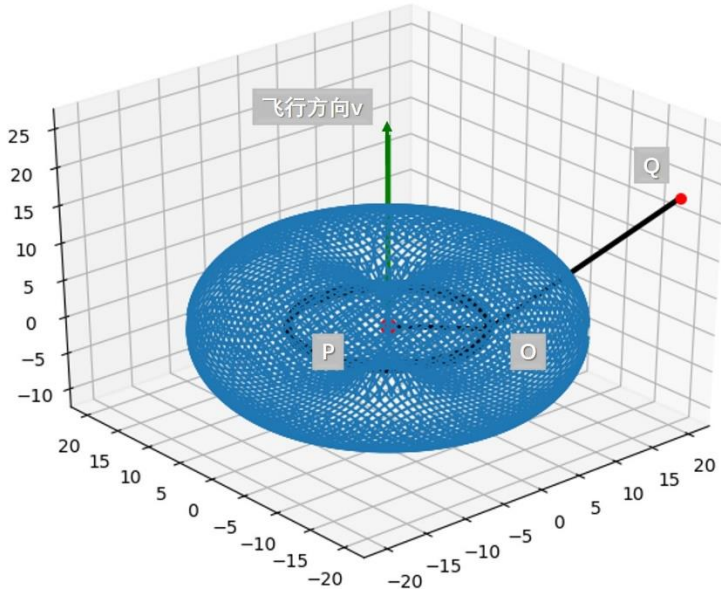


图 15 环形管道模型判断是否合格点

如图所示，P 点是已知航迹段的末端， $v$  为飞行器在 P 点的飞行方向，以 P 点为中心，最小转弯半径  $r$  为半径作一个圆 C，以圆 C 为中心，作一个半径为  $r$  的环形管道，由于转弯半径的限制，下一航迹经过的校正点 Q 应在环形管道以外，若 Q 点在环形管道内部，则即使飞行器以最小转弯半径转弯飞行，也无法到达 Q 点。

在判断 Q 点是否在管道以外时，我们可以利用圆 C，当 Q 点到圆 C 的最近距离大于  $r$  时，Q 点才在环形管道以外，才是满足转弯半径约束条件的可行校正点：

$$d_{QO} \geq r$$

求解  $d_{QO}$ 。

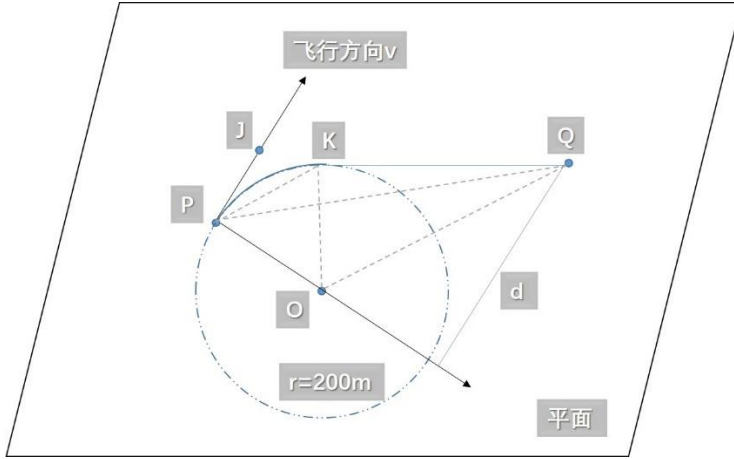


图 16 求解路径、运动方向示意图

如图所示，飞行方向  $v$  与  $P$ 、 $Q$  构成的平面，过  $P$  点作半径为  $r$  的圆  $O$ ，圆心为  $O$ ， $QO$  的长度即是我们待求的  $Q$  点到上面圆  $C$  的最短距离。

求解过程如下：

设  $v$  坐标  $(vx, vy, vz)$ ， $P(x1, y1, z1)$ ， $Q(x2, y2, z2)$ 。

过  $P$  点，以  $v$  为法向量的平面  $S$  为：

$$vx * (x - x1) + vy * (y - y1) + vz * (z - z1) = 0$$

过  $Q$  点作  $PO$  延长线的垂线， $Q$  点到  $PO$  的距离记为  $d$ ，即为  $Q$  点到平面  $S$  的距离：

$$d = \frac{|vx * (x2 - x1) + vy * (y2 - y1) + vz * (z2 - z1)|}{\sqrt{vx^2 + vy^2 + vz^2}}$$

$PQ$  距离为：

$$d_{PQ} = \sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2}$$

在平面  $PQO$  上建立二维坐标系：

以  $P$  点为坐标原点，则  $P$  点坐标变为  $(0, 0)$ ， $Q$  点坐标变为：

$$(\sqrt{d_{PQ}^2 - d^2}, d)$$

$O$  点坐标为  $(r, 0)$ ，即可求出  $QO$  距离：

$$d_{QO} = \sqrt{(\sqrt{d_{PQ}^2 - d^2} - r)^2 + d^2}$$

若  $d_{QO} \geq r$ ，则  $Q$  点是满足最小转弯半径的校正点。

**求解最短路径：**

由于飞行器无法即时转弯，故当  $v$  不指向  $Q$  时，飞行器的最短路径应当是从  $P$  沿着圆  $O$  直到  $v$  指向  $Q$ ，即过  $Q$  点作圆  $O$  的切线，切点为  $K$ ，飞行器从  $P$  点到  $Q$  点的最短路径应当是圆弧段  $PK$  加线段  $KQ$ 。

求路径  $PKQ$  长度：

$$d_{KQ} = \sqrt{d_{QO}^2 - r^2}$$

设  $\angle KOQ = \theta_2$ ， $\angle POK = \theta_1$ ， $\angle QOd = \theta_3$ ，

$$\theta 2 = \arccos \left( \frac{r}{d_{QO}} \right)$$

$$\theta 3 = \arcsin \left( \frac{d}{d_{QO}} \right)$$

$$\theta 1 = \pi - \theta 2 - \theta 3$$

$$PK = \theta 1 * r$$

则从校正点 P 到校正点 Q 的最短路径长度为：

$$l_{PQ} = PK + d_{KQ}$$

### 更新运动方向：

飞行器到达 Q 点的运动方向是 KQ 方向，更新运动方向，即是求解 KQ 方向，Q 点坐标已知，需要求解 K 点坐标。

可以联想到用四元数法去求解空间点的坐标：

Step1: 由 v 和 P、Q 坐标求平面 POQ 的方程；

Step2: 由平面方程得到垂直于此平面的垂线 W 向量；

Step3: 在 v 方向上取一点 J，使得 PJ 和 PK 长度相等，PK 长度可以通过解三角形 PKO 得到，从而得到 J 点坐标；

Step4:  $\angle JPK = \theta 1 \frac{1}{2}$ , 可以把 K 点看成是 J 点绕空间向量 W 旋转  $\alpha = \angle JPK$  角度得到的点；

Step5: 建立四元数模型，四元数  $Q \left( \cos \left( \frac{\alpha}{2} \right), \sin \left( \frac{\alpha}{2} \right) \cdot (u, v, w) \right)$ , 其中  $(u, v, w)$  是空间向量 W 的坐标，令  $Q = (q_0, (q_1, q_2, q_3))$ , 则旋转后四元数坐标为  $P' = QJQ^{-1}$ , 于是可求得 K 点坐标为：

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_0q_2 + 2q_1q_3 \\ 2q_0q_3 + 2q_1q_2 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_0q_1 + 2q_2q_3 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

其中  $(x', y', z')$  是待求点 K 点坐标， $(x, y, z)$  代表 J 点坐标。

于是到达 Q 点时的运动方向为  $v = (x' - x_2, y' - y_2, z' - z_2)$

## 5.3 模型的求解及分析

问题 2 的数学模型与问题 1 类似，问题 2 仅在约束条件中增加了飞行器最小转弯半径条件，问题 2 仍然采用遗传算法来解决。

### 5.3.1 算法设计

问题 2 同样采用遗传算法求解，算法流程与问题 1 相似，分为个体编码设计、初始种群生成、单点交叉、变异算子、多目标函数加权和评价方法、轮盘赌选择。

所不同的是在初代求解中考虑到 Q 点的可行性，更新误差时求解路径不再是算欧拉距离，而是求的 PKQ 曲线长度，求解方法在 5.2 节中已经详细说明，在求 Q 点的可行性和求最短路径更新误差时，都不免需要用到飞行器的运动方向，更新运动方向的方法也已经在上节中详细说明。

另外在计算适应度的时候，也需要考虑引入运动方向和路径计算，因为误差更新和路径长度和这些都息息相关。

## 算法流程:

数据预处理:

5、读取数据

6、投影到 AB 线段上, 计算出次序和距离

7、剔除投影在 AB 线段以外的点, 并取离 AB 距离在内部三分之一的点

8、利用剩下的点生成以每个节点编号为键值的字典, 字典内容是以键为中心的半球胞体的校正点编号、类型、离键点的距离

生成遗传初代:

While 循环次数小于种群规模:

1、第一个点设置为 A 点, 个体染色体 population[i]全置 0

2、遍历以 A 为键的值:

If 类型为 1, 调用判断函数, 判断点是否在环形管道内部, 若不在:

加入到待选点集 node\_candidate 中

Else if 类型为 0, 调用判断函数, 判断点是否在环形管道内部, 若不在:

加入到待选点集 node\_candidate 中

3、在 0 到 len(node\_candidate)之间的一个随机整数 r, 取 node\_candidate 中第 r+1 个点作为 A 点的下一点, 保存到航迹点列表 node\_list 中

4、水平误差 error\_level $\leftarrow$ distance\*delta

垂直误差 error\_vertical $\leftarrow$ distance\*delta

当前所在校正点 temp\_node $\leftarrow$ 选中的校正点信息包括 type、distance、num

5、if type==1:

error\_vertical=0

else:

error\_level=0

6、while 当前所在校正点不是 B 点:

For 遍历当前所在校正点 temp\_node 为中心的半球胞体:

判断是否符合环形管道准则

判断类型与到达待选点之前的误差是否符合校正要求;

若符合要求, 则将此点加入待选点集合

If 待选点集合为空:

从 A 点开始重新找航迹

Else if 待选点集合中只有一个元素:

选中此点

Else:

在 0 到待选点集合元素数目的范围中生成一个随机数 r

选中待选点集合中的第 r+1 个校正点

根据 temp\_node 与选中点之间的距离更新误差

更新方向

根据选中点的类型校正误差

Temp\_node $\leftarrow$ 选中的点

将 temp\_node 加入校正点列表 node\_list 中

For 遍历 node\_list 中的节点:

将染色体 population[i]中对应位置置 1

交叉:

For 遍历种群中的每一个个体:

    生成一个[0,1]之间的随机数  $r$

    If  $r <$  交叉概率  $pc$ :

        生成一个[0,染色体长度-1]之间的随机整数  $t$

        将染色体  $population[i]$  中第  $r+1$  位之后的部分  $population[i][r:]$  与下一条染色体中第  $r+1$  位之后的部分  $population[i][r:]$  进行互换,生成两个交叉个体

        将交叉个体加入到种群当中

变异:

For 遍历种群中的每一个个体:

    生成一个[0,1]之间的随机数  $r$

    If  $r <$  交叉概率  $pm$ :

        生成一个[0,染色体长度-1]之间的随机整数  $t$

        将染色体  $population[i]$  中第  $r+1$  位取反

        在该位对应校正点的半球胞体中随机选择一个与之类型相同的校正点,将其置为变异位变异之前的状态,生成三个变异个体

        将变异个体加入种群中

计算适应度:

For 遍历种群中的每一个个体:

    判断变量  $F$  设为 True

    For 遍历每个个体中的每一位:

        If 某一位为 1:

            校正点数目+1

            计算路径长度、误差

            If 不满足校正条件:

$F \leftarrow$  False

            Else:

                进行误差校正

    If 到达 B 点的误差大于  $\theta$ :

$F \leftarrow$  False

    If  $F ==$  False:

        给总长度  $length$  和总校正点数目  $num\_node$  乘一个较大的数目

    计算  $length$ 、 $num\_node$  中的最大、最小值

    根据多目标函数加权原理, 计算目标函数值  $z_k$

$Z_k \leftarrow 1/z_k$

求种群  $z_k$  的累积和

For 遍历每个  $z_k$ :

$Fitness \leftarrow z_k / sum$

选择:

依次计算  $fitness$  的累积和

生成[0, 1]范围内的随机数列表  $ms$ , 长度为预先设计的种群规模

```

For 遍历 ms 中的每个数：
    判断 ms[i] 处于累积和中的哪个位置
    取出对应位置处的父代个体放入子代种群中
后续一些处理：
每一次迭代形成的子代中选取适应度最大的个体
计算每一代最优个体的误差、长度等

```

以上过程循环预先设计的最大迭代次数

画图可视化

导出数据、表格

## 5.4 问题 2 求解结果及分析

### 求解结果：

在问题一算法的基础上，生成初代时候考虑环形管道原则，计算路径长度时考虑新的路径、运动方向计算方法。

对于数据集 1，选取了 100 个体规模的种群，最大迭代次数设为 500 次，在飞行器在两校正点的路径为圆弧加直线的假设条件下，综合考虑航迹长度和经过校正点的数目，求解出来的航迹结果如下：

	校正点编号	校正前垂直误差	校正前水平误差	校正点类型
0	0	0.000000	0.000000	出发点A
1	503	13.387920	13.387920	11
2	69	8.808412	22.196332	01
3	237	21.308271	12.499858	11
4	233	10.827686	23.327544	01
5	33	16.002388	5.174702	11
6	194	10.505069	15.679771	01
7	450	16.491469	5.986401	11
8	594	18.061461	24.047862	01
9	397	21.126483	3.065022	11
10	612	17.003445	20.068467	终点B

图 17 附件 1 结果表

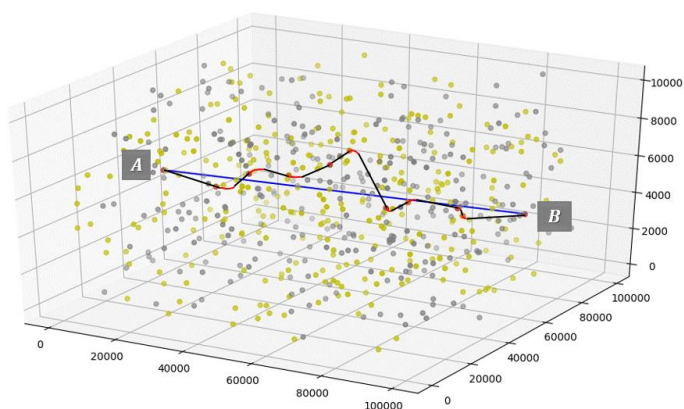


图 18 路径结果图

对于数据集 2，选取了 100 个体规模的种群，最大迭代次数设为 500 次，在飞行器在两校正点的路径为直线的假设条件下，综合考虑航迹长度和经过校正点的数目，求解出来的航迹结果如下：

校正点编号	校正前垂直误差	校正前水平误差	校正点类型	
0	0	0.000000	0.000000	出发点A
1	163	13.287902	13.287902	01
2	114	18.629452	5.341550	11
3	234	4.556869	9.898419	01
4	222	11.839790	7.282921	11
5	227	8.167972	15.450892	01
6	309	13.640552	5.472580	11
7	305	5.978931	11.451511	01
8	123	15.183462	9.204531	11
9	45	10.011281	19.215812	01
10	160	17.510473	7.499192	11
11	92	5.785733	13.284925	01
12	93	15.275843	9.490110	11
13	61	9.837364	19.327475	01
14	292	16.394806	6.557442	11
15	326	7.007259	13.564701	终点B

图 19 附件 1 结果表



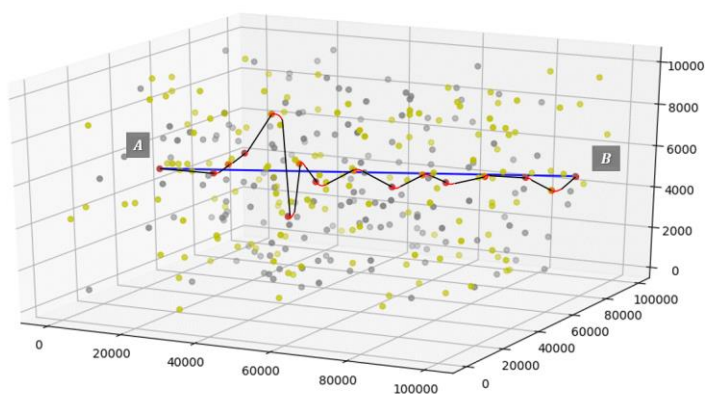


图 20 路径结果图

### 问题 2 结果分析：

在对飞行器最短路径进行建模时，我们将沿运动方向的圆弧一直飞到目标点的方案和上述先 200m 半径圆弧后直线的方案进行了比较，不仅在几何理论上证明了后者方案路径更短，而且从优化数据中也可以看出最短路径应当是先圆弧再直线的方案。

将问题 2 和问题 1 的优化结果进行比较后发现，问题 2 的可行解更少，在遗传算法中进化的更慢，这是因为有很多在问题 1 中满足条件的点在问题 2 中不满足转弯半径限制，而且由于问题 2 中飞行器的路径不再是直线，而是比直线更长的曲线，导致校正点与校正点之间的误差增量更大，也会让一部分方案失效，最终的结果和问题 1 相比也是距离更长一点。

并且把数据集 2 和数据集 1 比较，在空间散点图中可以看出数据集 2 的点分布更稀疏，导致可选的方案更少，故优化结果包括结点数目和路径长度比数据集 1 稍差一些。

## 6 问题 3 模型求解及算法设计

### 6.1 问题分析

问题 3 是在问题 1 的基础上增加了概率约束，我们采用在问题 1 的遗传算法上增加机会约束规划模型。问题三的目标函数相对于问题 1 有所变化，增加了到达终点概率尽可能大这一目标，我们按照优化目标重要程度对目标进行排序，表示如下：

- (1) 最小化总校正次数和最小化飞行器轨迹长度；
- (2) 最大化到达终点概率。

### 6.2 模型建立与准备

#### 6.3.1 数据定义与初始化

$pk$ ——一个体中第  $k$  种情况发生的概率

$P$ ——单个个体成功飞行到  $Q$  点且满足误差条件的概率

$\alpha$ ——置信水平

$p1, p2$ ——惩罚系数

#### 6.3.2 问题 3 模型建立

问题 3 需满足的约束条件前 5 个同问题 1：

- (1) 飞行器按规划路径飞行条件：

$$e_{v_i} < \theta, e_{h_i} < \theta, i = 1, 2, \dots, m$$

- (2) 飞行器误差增加方式：

$$e_{v_i} = e_{v_{i-1}} + \delta * \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2},$$

$$e_{h_i} = e_{h_{i-1}} + \delta * \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}, i = 1, 2, \dots, m$$

- (3) 在  $A$  点的条件：在出发地  $A$  点，飞行器的垂直和水平误差均为 0：

$$e_{v_0} = 0, e_{h_0} = 0$$

- (4) 飞行器进行误差校正的方式：经过垂直误差校正后，

$$e_{v_i} = 0, e_{h_i} = e_{h_{i-1}}, i = 1, 2, \dots, m$$

经过水平误差校正后，

$$e_{h_i} = 0, e_{v_i} = e_{v_{i-1}}, i = 1, 2, \dots, m$$

(5) 飞行器进行校正的条件：

进行垂直误差校正：

$$e_{v_i} \leq \alpha_1, e_{h_i} \leq \alpha_2, i = 1, 2, \dots, m$$

进行水平误差校正：

$$e_{v_i} \leq \beta_1, e_{h_i} \leq \beta_2, i = 1, 2, \dots, m$$

问题 3 与问题 1 不同的约束条件：

(6) 飞行器在可能出现问题的校正点能够成功将某个误差校正为 0 的概率为 80%，

如果校正失败，则校正后的剩余误差为  $\min(\text{error}, 5)$  个单位（其中  $\text{error}$  为校正前误差， $\min$  为取小函数）：

$$p\{e_{v_i}(\zeta_i) < \theta, e_{h_i}(\zeta_i) < \theta\} \geq \alpha, i = 1, 2, \dots, m \quad (6-1)$$

其中  $\zeta_i = \begin{cases} 1, & \text{校正成功} \\ 0, & \text{校正失败} \end{cases}$

(7) 飞行器到达该校正点时即可知道在该点处是否能够校正成功，但不论校正成功与否，均不能改变规划路径。

综上，可以建立问题 3 的数学模型，见 MOD3：

MOD3

$$\min z_1 = f_1(c), z_2 = f_2(x, y, z)$$

$$s. t. \begin{cases} e_{v_i} < \theta, e_{h_i} < \theta, i = 1, 2, \dots, m \\ e_{v_i} = e_{v_{i-1}} + \delta * \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \\ e_{h_i} = e_{h_{i-1}} + \delta * \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \\ e_{v_0} = 0, e_{h_0} = 0 \\ e_{v_i} = 0, e_{v_i} = e_{v_{i-1}}, i = 1, 2, \dots, m \\ e_{h_i} = 0, e_{v_i} = e_{v_{i-1}}, i = 1, 2, \dots, m \\ e_{v_i} \leq \alpha_1, e_{h_i} \leq \alpha_2, i = 1, 2, \dots, m \\ e_{v_i} \leq \beta_1, e_{h_i} \leq \beta_2, i = 1, 2, \dots, m \\ p\{e_{v_i}(\zeta_i) < \theta, e_{h_i}(\zeta_i) < \theta\} \geq \alpha, i = 1, 2, \dots, m \end{cases}$$

## 6.3 模型的求解与分析

### 6.3.1 算法设计

问题 3 算法同样采用遗传算法求解，算法的流程与问题 1 类似。

**多目标函数加权和评价方法融合机会约束规划模型：**

第一步 求出飞行器能够飞到 B 点的概率：

$$p\{e_{v_i}(\zeta_i) < \theta, e_{h_i}(\zeta_i) < \theta\} = \sum_{i=1}^m F * p_i \quad (6-2)$$

其中， $p_i$  为成功校正的概率， $p_i = 0.8$ 。

$$F = \begin{cases} 1, & \text{成功到达B点} \\ 0, & \text{没法成功到达B点} \end{cases} \quad (6-3)$$

第二步  $z_k^{max}$ ， $z_k^{min}$  表示第  $k$  个目标函数的最大值和最小值， $k=1、2$ ，如下定义。P 为可行域解的集合。

$$z_k^{max} = \max_P \{f_1\} \quad (6-4)$$

$$z_k^{min} = \min_P \{f_1\} \quad (6-5)$$

计算各个目标函数值的  $z_k$ 。

第三步 判断，若  $p\{e_{v_i}(\zeta_i) < \theta, e_{h_i}(\zeta_i) < \theta\} \geq \alpha$ ，则  $z_k$  不发生变化，否则，使用如下惩罚函数，

$$z_k = [p_1 f_1(g_j), p_2 f_2(g_j)] \quad (6-6)$$

$p_1, p_2$  为 100 到 500 中的随机数。其中，若满足约束条件，则  $z_k$  不发生改变。

第二步 某一个染色体  $g_j$  的权重和由下式求得。

$$\text{eval} = \sum_{k=1}^2 w_k (f_k - z_k^{min}) = \sum_{k=1}^2 \frac{f_k - z_k^{min}}{z_k^{max} - z_k^{min}} \quad (6-7)$$

其中第  $k$  个目标函数的适应性权重  $w_k$  由下式计算。

$$w_k = \frac{1}{z_k^{max} - z_k^{min}} \quad (6-8)$$

用此公式获得各非劣解的评价函数值  $\text{eval}(g_j)$ 。

## 算法流程:

数据预处理:

读取数据

投影到 AB 线段上, 计算出次序和距离

剔除投影在 AB 线段以外的点, 并取离 AB 距离在内部三分之一的点

利用剩下的点生成以每个节点编号为键值的字典, 字典内容是以键为中心的半球胞体的校正点编号、类型、离键点的距离

生成遗传初代:

While 循环次数小于种群规模:

1、第一个点设置为 A 点, 个体染色体 `population[i]` 全置 0

2、遍历以 A 为键的值:

    If 类型为 1, 并且离 A 点距离  $\text{distance} \leq \min(\alpha_1, \alpha_2)$ :

        加入到待选点集 `node_candidate` 中

    Else if 类型为 0, 并且离 A 点距离  $\text{distance} \leq \min(\beta_1, \beta_2)$ :

        加入到待选点集 `node_candidate` 中

3、在 0 到  $\text{len}(\text{node\_candidate})$  之间的一个随机整数  $r$ , 取 `node_candidate` 中第  $r+1$  个点作为 A 点的下一点, 保存到航迹点列表 `node_list` 中

4、水平误差  $\text{error\_level} \leftarrow \text{distance} * \delta$

    垂直误差  $\text{error\_vertical} \leftarrow \text{distance} * \delta$

    当前所在校正点 `temp_node`  $\leftarrow$  选中的校正点信息包括 `type`、`distance`、`num`

5、if `type==1`:

    If 是问题校正点:

`error_vertical=1`

    else:

`error_vertical=0`

else:

    If 是问题校正点:

`error_vertical=1`

    else:

`error_vertical=0`

6、while 当前所在校正点不是 B 点:

    For 遍历当前所在校正点 `temp_node` 为中心的半球胞体:

        判断类型与到达待选点之前的误差是否符合校正要求:

        若符合要求, 则将此点加入待选点集合

    If 待选点集合为空:

        从 A 点开始重新找航迹

    Else if 待选点集合中只有一个元素:

        选中此点

    Else:

        在 0 到待选点集合元素数目的范围中生成一个随机数  $r$

        选中待选点集合中的第  $r+1$  个校正点

```

        根据 temp_node 与选中点之间的距离更新误差
        根据选中点的类型校正误差
        Temp_node ← 选中的点
        将 temp_node 加入校正点列表 node_list 中
    For 遍历 node_list 中的节点:
        将染色体 population[i] 中对应位置置 1
交叉:
For 遍历种群中的每一个个体:
    生成一个[0,1]之间的随机数 r
    If r < 交叉概率 pc:
        生成一个[0,染色体长度-1]之间的随机整数 t
        将染色体 population[i] 中第 r+1 位之后的部分 population[i][r:] 与下一条染色体中第 r+1 位
        之后的部分 population[i][r:] 进行互换,生成两个交叉个体
        将交叉个体加入到种群当中
变异:
For 遍历种群中的每一个个体:
    生成一个[0,1]之间的随机数 r
    If r < 交叉概率 pm:
        生成一个[0,染色体长度-1]之间的随机整数 t
    将染色体 population[i] 中第 r+1 位取反
    在该位对应校正点的半球胞体中随机选择一个与之类型相同的校正点,将其置为变异位变异之前的
    状态,生成三个变异个体
    将变异个体加入种群中
计算适应度:
For 遍历种群中的每一个个体:
    判断变量 F 设为 True
    设此个体能到达 B 点且满足约束条件的概率为 P
    For 遍历问题点每一种组合情况:
        得到此种情况的概率 pk
        For 遍历每个个体中的每一位:
            根据类型、问题点是否正常校正:
                校正点数目+1
                计算距离、误差
                If 不满足校正条件:
                    F ← False
                Else:
                    进行误差校正
        If 到达 B 点的误差大于 theta:
            F ← False
        If F == False:
            给总长度 length 和总校正点数目 num_node 乘一个较大的数目
    Else:
        P += pk
    计算 length、num_node 中的最大、最小值

```

根据多目标函数加权原理，计算目标函数值  $z_k$

$Z_k \leftarrow 1/z_k$

求种群  $z_k$  的累积和

For 遍历每个  $z_k$ :

$Fitness \leftarrow z_k/sum$

选择:

依次计算  $fitness$  的累积和

生成 $[0,1]$ 范围内的随机数列表  $ms$ ，长度为预先设计的种群规模

For 遍历  $ms$  中的每个数:

判断  $ms[i]$  处于累积和中的哪个位置

取出对应位置处的父代个体放入子代种群中

后续一些处理:

每一次迭代形成的子代中选取适应度最大的个体

计算每一代最优个体的误差、长度等

以上过程循环预先设计的最大迭代次数

### 6.3.2 问题 3 求解结果:

关于问题三结果表格中类型一列的说明:

由于该题中校正成功与否是概率性的，最终输出的只是其中的一种结果，假如加上概率因素，可能还无法成功到达 B 点，此种情况下，只考虑最好情况即把所有问题点都当做正常校正点去校正误差。假如说是否校正成功，只能说下面给出的结果的每个点都是校正成功的。

对于数据集 1:

校正点编号	校正前垂直误差	校正前水平误差	校正点类型
0	0	0.000000	出发点A
1	346	11.257594	1
2	69	11.814998	0
3	123	18.160946	1
4	75	12.092893	0
5	61	21.060080	1
6	282	3.622216	0
7	288	17.609267	1
8	122	4.854497	0
9	113	12.914779	0
10	406	16.791803	1
11	400	10.766322	1
12	425	7.300148	0
13	612	23.553260	16.253111 终点B

图 21 附件 1 结果表

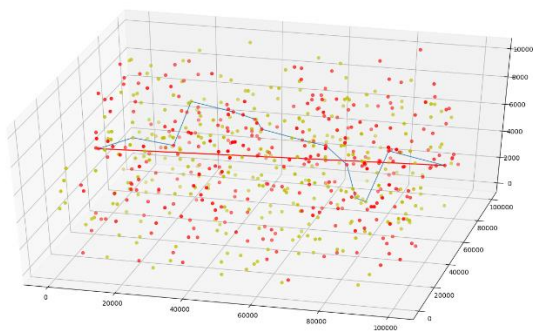


图 22 路径结果图

对于数据集 2:



校正点编号	校正前垂直误差	校正前水平误差	校正点类型
0	0	0.000000	出发点A
1	260	8.348976	1
2	226	11.256952	0
3	180	16.411977	1
4	158	7.500158	0
5	58	11.333732	1
6	173	3.490014	1
7	8	7.706712	0
8	309	13.231032	1
9	305	5.968718	0
10	121	11.806069	0
11	123	17.157672	1
12	49	11.790194	0
13	160	18.303997	1
14	269	6.446384	0
15	324	12.506359	0
16	19	17.151523	1
17	244	10.449399	0
18	88	18.230682	1
19	38	11.609628	0
20	287	15.574903	1
21	161	10.117929	0
22	59	16.634045	1
23	135	3.987457	0
24	326	5.182241	1.194784 终点B

图 23 附件 1 结果表

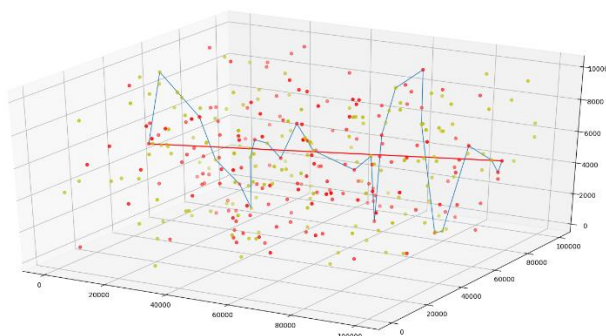


图 24 路径结果图

### 6.3.3 数据结果分析

在初代生成阶段，我们将是否是有问题校正点加上了一个惩罚措施，就是在校正的时候，虽然无法提前知道能否校正成功，但是根据 0.2 的失败概率，以及矫正后的误差  $\min(\text{error}, 5)$ ，通过前面两题的经验来看，error 基本上是在 5 以上，故可把有问题点校正后的误差期望记作  $5 \times 0.2 = 1$ ，故在校正的时候，正常校正点依然校正为 0，有问题校正点校正为 1，这样在一定程度上体现了概率问题点的特性，使之比正常点的误差产生更大，就有更低的概率在遗传进化中传递下去。

在适应度计算过程中，我们把概率性和多目标加权的惩罚系数联系起来，在我们的模型中，我们对于每一个个体我们找出所有的问题点，问题点的分布类似于伯努利分布，

我们把每个问题校正点正常校正的情况记作 1，校正出现问题的情况记作 0，穷举搜索所有的可能情况，并且计算出此种情况下飞行器能否在满足误差限制的条件下来到达 B 点，若能到达 B，则利用伯努利概率公式  $p^i(1-p)^{n-i}$  求出此种情况的概率  $p_k$ ，累积所有能成功到达 B 点的概率，最后的结果就是该个体能成功到达 B 点的概率  $P$ ，我们还设置了置信度  $\alpha$ ，即约束条件成立的可靠度，当  $P \geq \alpha$  时，将此校正点当做普通无问题校正点看待，反之当做问题一和问题二中不符合误差约束条件的点考虑，给总路径长度和总结点数目乘上一定的惩罚系数，使之被选中的概率下降，从而有更小的几率遗传给下一代。

此种模型计算精确度很高，但是时间复杂度也很高，因为一旦初代选中的个体中有较多的问题校正点的时候，之后遍历状态空间的时候，会成指数倍增长，因为对于一个具有  $n$  个校正点的个体，它的校正点状态的可能数目为  $2^n$  个。故在有些情况下计算所消耗时间太长，对于数据集一，我们采取的种群数目为 100，最大迭代次数为 500，在 i5 九代 CPU，2.4GHz 笔记本上运行了一个半小时得出结果，从路径长度、路径图像、校正点成功到达 B 点的概率来看，效果还是不错的，但是对于数据集 2 来说，由于校正点分布地较散，导致有些可行解中的问题校正点数目较多，从而导致计算时间复杂度特别大，我们只能取较小的种群规模和较小的最大迭代次数，得到了次优可行解。

## 7 模型的总结与评价

### 7.1 模型总结

三个模型均为多目标多约束模型，三个模型的目标函数都相同，但约束条件不相同。

(1) 问题 1 模型是以最小化总校正次数和最小化飞行器轨迹长度为目标函数的双目标多约束模型，采用遗传算法设计。本文中遗传算法的亮点是初始种群生成的设计，建立起以校正点为中心、AB 方向为前进方向的半球邻近胞体模型，并在数据经验的基础上，只取较靠近 AB 线段的区域点，大大减少了搜索空间，并以待选区域为基因位链，成功建立起初代种群，在适应度计算上，我们采用了多目标函数加权和评价方法，当约束条件在某种程度上不满足时，给每个目标函数加上不同的惩罚系数，以综合多目标多约束条件，在变异的过程中，考虑到解空间的复杂性，我们改进了传统的变异方法，构建出一种小范围变异的思路，以尽量加大子代中可行解的数量，最后采用轮盘赌从父代中选择子代；

(2) 问题 2 模型是在问题 1 的基础上增加了约束条件，在构建初代的过程中，我们采用了环形管道规则引擎，当目标点在原始点的环形管道以外则不符合条件重新搜索，另外在误差更新的时候，我们具体化了飞行器的飞行路径，以求出飞行器的路径长度，在此过程中用到了三维坐标降二维坐标技术、空间几何和平面几何知识，在求解飞行器的运动路径的时候，我们巧妙地把问题转换为空间点绕空间向量旋转后求坐标的问题，并成功运用超复数四元数理论求解此问题；

(3) 问题 3 在问题 1 的基础上增加了机会约束，我们参考了机会约束规划模型，在置信度以外赋予对应点相应的惩罚系数，巧妙地将概率约束与目标函数求解联系起来

### 7.2 模型的优点

本文采用的数学模型能够清晰简明的反映我们的目标与各种约束，它主要包含以下优点：

- (1) 决策变量能够针对校正点选择问题，模型思路清晰，约束方程简单明了，有利于模型的求解；
- (2) 模型的可移植性高，成功应用于遗传性算法；
- (3) 模型的可扩展性强，可再继续添加规则，减小时间、空间复杂度；
- (4) 在模型建立的过程中融合了多种模型和规则，取各自的优点，针对于具体问题、数据，有的放矢，灵活构建规则、融合模型。

### 7.3 模型的待改进地方：

- (1) 在初代生成的阶段，虽然针对于问题 1 很有效，计算速度也很快，但是可以进

一步改进，针对于问题 2 的初代生成，则因为几何模型、逻辑较为复杂，使得生成初代较慢，针对于问题 3 则是因为约束的较多，初代生成也较慢，都有待于进一步优化算法；

(2) 在进化的过程中，我们发现进化的模式是阶梯型进化，即在连续若干代里都没有更新最优解，还可以进一步改进选择、变异、交叉思路以加快进化速度，扩大种群多样性；

(3) 在问题 3 的求解过程中，利用求解每个点成功到达 B 的概率 P 来惩罚目标函数的思路是正确的，但是针对于数据集 2，计算 P 的时间复杂度较高，可以进一步优化求解 P 的方法和思路。

## 8 参考文献

- [1]玄光男. 林林 (M) . 网络模型与多目标遗传算法. 北京: 清华大学出版社, 2017.
- [2]庞碧君 (M) . 线性规划与随机线性规划. 郑州: 郑州大学出版社, 2009.
- [3]孙小玲. 白小迪. 郑小金. 概率约束最优化问题 (J) . 运筹学学报, 2012 (3) : 65-74
- [4] 刁训娣. 基于多目标遗传算法的项目调度及其仿真研究[D]. 上海交通大学, 2010.
- [5] Cao Y J , Wu Q H . Optimal reactive power dispatch using an adaptive genetic algorithm[C]// Genetic Algorithms in Engineering Systems: Innovations and Applications, 1997. GALESIA 97. Second International Conference On (Conf. Publ. No. 446). IET, 1997.
- [6] Yun Y , Gen M . Performance Analysis of Adaptive Genetic Algorithms with Fuzzy Logic and Heuristics[J]. Fuzzy Optimization and Decision Making, 2003, 2(2):161-175.
- [7]Kim K W , Yun Y S , Yoon J M , et al. Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling (J) . Computers in Industry, 2005, 56(2):143-160.
- [8] Zbigniew Michalewicz . Genetic Algorithms+ Data Structures= Evolution Programs (J) . New York: Spring-Verlag, 1992