



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 | 中国科学院大学 |
|--------|---------|

| | |
|------|-------------|
| 参赛队号 | 19144300055 |
|------|-------------|

| | |
|--|--------|
| | 1. 候宗府 |
|--|--------|

| | |
|------|--------|
| 队员姓名 | 2. 孙晓伟 |
|------|--------|

| | |
|--|--------|
| | 3. 胡潇尹 |
|--|--------|

“华为杯”第十六届中国研究生 数学建模竞赛

题目 多约束条件下智能飞行器航迹快速规划

摘要：

复杂环境下航迹快速优化是智能飞行器控制的一个重要课题。但是由于系统结构限制，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。为了成功对定位误差进行校正，本文针对题目中所提出的三种情况分别建立了规划模型，并在 Python 语言环境下利用 Gurobi 软件进行求解。

对于问题一，我们采用了一般最短路问题的规划模型并在目标函数中对路径经过的节点总数进行了惩罚，最后通过对题目中所要求的垂直和水平校正误差进行限制，建立了规划模型。在对附件 1 和附件 2 的 Excel 表格中的数据集实际求解时，我们预先根据文中所给的要求，通过计算，对可行边集合进行了剪枝，从而将变量数从 375769 和 106929 减少到了 31266 和 5483。最后利用 Gurobi 对规划模型的求解，得到结果：附件 1-数据集经过校正点数最少的最短路径为 0->503->294->91->607->540->250->340->277->612，路径长为 104861（详见图 1.1 表 1.1）；附件 2-数据集经过校正点数最少的最短路径为 0->163->114->8->309->305->123->45->160->92->93->61->292->326，路径长为 109342（详见图 1.2 表 1.2）。

对于问题二，我们在考虑最小转弯半径为 200m 的条件之后建立了物理模型，采用一段圆弧连接切线段的模式进行飞行器的航迹规划。与问题一不同的是，问题二的模型变量为可行边集合连接的三元组可行边，这使得模型复杂度大大提高，直接对模型求解困难巨大。对此，我们在问题一的基础上另外剪掉了与 AB 反向的可行边。同时，我们还以 AB 为轴线，R 为半径参数建立了剪枝圆柱体，用以减去多余的顶点和相应的边。最后利用 Gurobi 对此问题的规划模型求解，得到附件 1-数据集经过校正点数最少的最短路径为 0->503->294->91->607->540->250->340->277->612，路径长 104874（详见图 2.1 表 2.1）；附件 2-数据集经过校正点数最少的最短路径为 0->163->114->8->309->305->123->45->160->92->93->61->292->326，路径长为 109455（详见图 2.2 表 2.2）。

对于问题三，我们在考虑存在 20% 可能出现校正失败点的情况下建立了规划模型。同样地，我们对目标函数采用了节点总数的惩罚，以便于得到校正次数最少的路径。与此同时，我们通过对校正有问题的点添加了 0-1 变量的限制，在规划模型中引入了校正出现问题的约束条件。在通过与问题一相同的剪枝处理后，我们利用 Gurobi 求解此问题，得到附件 1-数据集经过校正点数最少的最短最短路径为 0->503->69->506->371->183->194->450->286->485->612，路径长 104864，成功到达终点的概率为 100%（详见图 3.1 表 3.1）；附件 2-数据集经过校正点数最少的最短路径为 0->169->322->270->89->236->132->53->112->268->250->243->73->249->274->12->216->16->282->141->291->161->326，路径长为 161650，成功到达终点的概率为 100%（详见图 3.2 表 3.2）。值得注意的是，我们的模型具有非常好的鲁棒性，即倘若问题三所给的数据不存在能够 100% 到达终点的路径，我们依旧可以通过求解这个模型，给出概率最大的最优路径。

关键字： 航迹规划 规划模型 Gurobi 软件 剪枝

目录

| | |
|-----------------------|-----------|
| 1. 问题重述 | 3 |
| 2. 模型假设 | 4 |
| 3. 符号说明 | 5 |
| 4. 模型建立与求解 | 6 |
| 4.1 问题 1 | 6 |
| 4.1.1 问题分析 | 6 |
| 4.1.2 模型建立 | 6 |
| 4.1.3 模型求解 | 8 |
| 4.1.4 结果展示以及有效性和复杂性分析 | 10 |
| 4.2 问题 2 | 11 |
| 4.2.1 问题分析 | 11 |
| 4.2.2 模型建立 | 11 |
| 4.2.3 模型求解 | 13 |
| 4.2.4 结果展示以及有效性和复杂性分析 | 16 |
| 4.3 问题 3 | 17 |
| 4.3.1 问题分析 | 17 |
| 4.3.2 模型建立 | 17 |
| 4.3.3 模型求解 | 18 |
| 4.3.4 结果展示以及鲁棒性分析 | 22 |
| 5. 模型评价 | 22 |
| 5.1 模型优点 | 22 |
| 5.2 模型缺点 | 23 |
| 6. 参考文献 | 23 |
| 附录 A 程序代码 | 24 |
| 1.1 第一题 | 24 |
| 1.2 第二题 | 27 |
| 1.3 第三题 | 32 |

1. 问题重述

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，从而产生定位误差，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和一些给定约束下的航迹快速规划问题。我们借助图 1 来分析该问题。假设飞行器的飞行区域如图 1 所示，其中出发点为 A 点，目的地为 B 点。在飞行过程中，会有一定约束。其航迹约束如下：

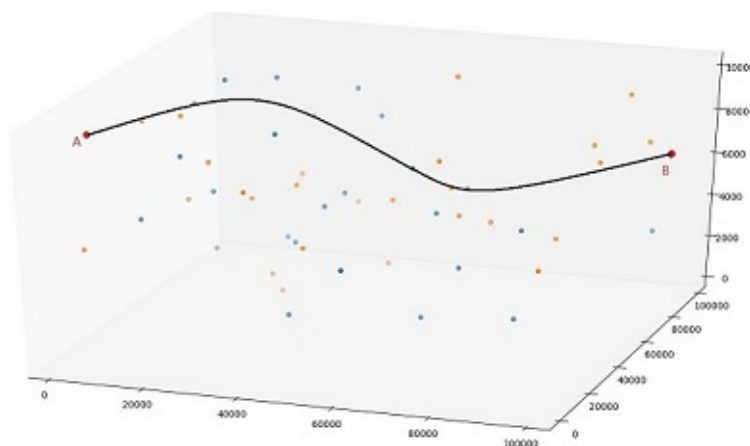


图 1 飞行器航迹规划区域示意图

(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。假设飞行器每飞行 1m，垂直误差和水平误差将各增加 δ 个单位。到达终点时垂直误差和水平误差均应小于 θ 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 θ 个单位时，飞行器仍能够按照规划路径飞行。

(2) 飞行器的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，黑色曲线代表一条航迹）。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 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 α_1 个单位，水平误差不大于 α_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 β_1 个单位，水平误差不大于 β_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。

请为上述智能飞行器建立从 A 点飞到 B 点的航迹规划一般模型和算法并完成以下问题：

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:

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

请绘出两个数据集的航迹规划路径, 并将结果 (即飞行器从起点出发经过的误差校正点编号及校正前误差) 依次填入航迹规划结果表, 放于正文中, 同时将两个数据集的结果填入附件 3 的 Sheet1 和 Sheet2 中。

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。
- 并讨论算法的有效性和复杂度。

请绘出两个数据集的航迹规划路径 (直线用黑色, 圆弧用红色), 并将结果 (即飞行器从起点出发经过的误差校正点编号及校正前误差) 依次填入航迹规划结果表, 放于正文中, 同时将两个数据集的结果填入附件 3 的 Sheet3 和 Sheet4 中。

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 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。

请绘出两个数据集的航迹规划路径, 并将结果 (即飞行器从起点出发经过的误差校正点编号及校正前误差) 依次填入航迹规划结果表, 放于正文中, 同时将两个数据集的结果填入附件 3 的 Sheet5 和 Sheet6 中。

2. 模型假设

根据题意, 提出下列模型假设

- 假设飞行器按规划航迹飞行时不受其他因素影响, 能正常沿既定航迹到达下一校正点。
- 假设飞行器的水平误差和垂直误差均小于 θ 个单位时, 飞行器仍能按照规划路径飞行。
- 在考虑问题一和问题二时, 假设若飞行器的水平和垂直误差不超过问题中所给的限度, 飞行器在校正点总是可以成功校正。
- 在考虑问题二时, 假设航迹上经过每个校正点时飞机的方向近似用前一校正点与当前校正点的连线方向代替。

- 在考虑问题三时，假设飞行器在到达该校正点时就可以知道是否校正成功，但不论校正成功与否，均不能改变规划路径。

3. 符号说明

| | |
|-------------|--|
| A | 飞行起点 |
| B | 飞行终点 |
| E | 所有有向边的集合 |
| H | 所有水平误差校正点的集合 |
| V | 所有垂直误差校正点的集合 |
| M | 给定的很大的正数 |
| K | 顶点个数的惩罚系数 |
| C | 20% 失败点在最优路径中所占数目的惩罚系数 |
| R | 校正点分布图中以 AB 为轴包含最优路径的圆柱体半径 |
| I_i | 飞行器到达 i 点时进行误差校正的示性变量 (1, 垂直误差校正; 0, 水平误差校正) |
| Λ_i | 在 i 点校正是否存在问题的示性变量 (1, 存在问题; 0, 正常校正) |
| Ψ_i | i 点是否影响飞行器成功到终点的示性变量 (1, 成功到达终点; 0, 未成功到达终点) |
| δ | 飞行器每飞行 1m, 垂直或水平误差增加的单位 |
| α_1 | 垂直误差校正时的垂直误差上限 |
| α_2 | 垂直误差校正时的水平误差上限 |
| β_1 | 水平误差校正时的垂直误差上限 |
| β_2 | 水平误差校正时的水平误差上限 |
| x_{ij} | 由点 i 到点 j 的航迹的示性变量 (1, 当飞行器从点 i 飞到点 j ; 0, 其它) |
| w_{ij} | 点 i 到点 j 的欧氏距离 |
| h_i | 飞行器到达 i 点时误差校正前的水平偏移量 |
| v_i | 飞行器到达 i 点时误差校正前的垂直偏移量 |
| r | 飞行器最小转弯半径 |
| \vec{v} | 飞行器初速度方向 |
| θ | 飞行器的方向变化角 |
| φ | 在第二题中飞行器由 i 点经飞行路径到 j 点时的速度方向与 i, j 之间的夹角 |
| u_{kij} | 飞行器由 k 出发经过 i 到 j 的路径所得到的 i, j 之间的路径长度 |

4. 模型建立与求解

4.1 问题1

4.1.1 问题分析

根据附件 1 和附件 2 的数据集，我们分别可以画出校正点的三维散点图。下面是附件 1 校正点的分布图。

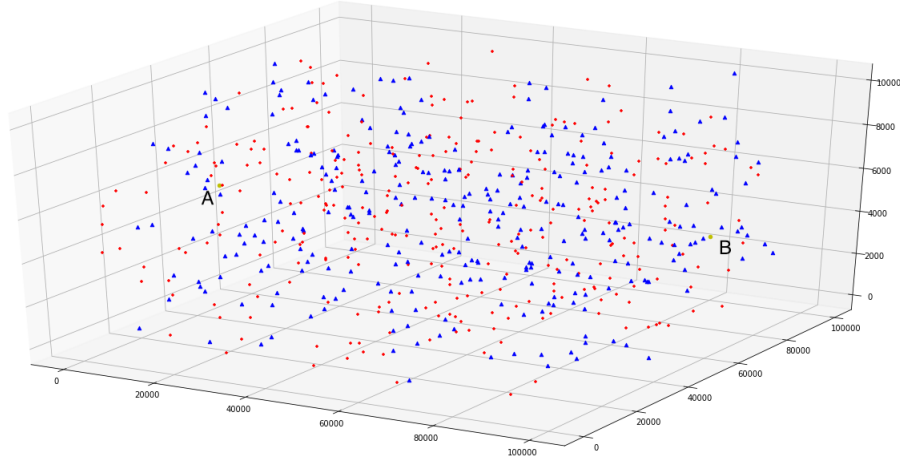


图 2 附件 1 校正点分布图

其中 A 点为出发点，B 点为终点，红色点表示垂直校正点，蓝色三角点表示水平校正点。

由图 2 可知，校正点在空间中的分布较为稠密。我们要解决的问题是对智能飞行器建立从 A 点飞到 B 点的航迹规划的模型和算法，其中规划要满足一些约束条件，并且要使得航迹长度尽可能小，经过的校正次数也要尽可能少。直观来讲，从出发点 A 到目的地 B，我们的任务是从三维散点图中筛选尽可能少的边，在满足一定的约束条件下，要保证这些边的长度总和尽可能小。因此，我们可以从一般的最短路问题入手，给出其目标函数和约束条件，然后根据该问题的要求构建改进的目标函数，并补充相应的约束条件，最后对模型进行求解。

4.1.2 模型建立

问题 1 中，要求对附件 1 和附件 2 中的数据分别规划满足一些约束条件时飞行器的航迹，并且使得航迹长度尽可能小，且经过校正区域进行校正的次数尽可能少。我们可以考虑最短路问题。在一般最短路问题中，目标函数为：

$$\min \sum_{(i,j) \in E} w_{ij} x_{ij}$$

约束条件有：

(1) 对于航迹图中除去飞行起点和终点的其它点有约束：

$$\sum_{(i,j) \in E} x_{ij} = \sum_{(j,k) \in E} x_{jk}, \forall j \neq A, B \quad (4.1.1)$$

其中 A 点为飞行起点，B 点为飞行终点。

(2) 对飞行起点 A 有约束：

$$\sum_{(A,j) \in E} x_{Aj} = 1, \quad \sum_{(j,A) \in E} x_{jA} = 0 \quad (4.1.2)$$

(3) 对飞行起点 B 有约束：

$$\sum_{(j,B) \in E} x_{jB} = 1, \quad \sum_{(B,j) \in E} x_{Bj} = 0 \quad (4.1.3)$$

在该问题中，除了考虑最短路问题的三个约束条件，还需加入几个约束条件：

(4) 飞行器从点 i 飞到点 j，水平误差应满足约束：

$$I_i h_i + \delta w_{ij} - h_j \leq (1 - x_{ij})M \quad (4.1.4)$$

其中 M 给定的很大的正数，下同。

(5) 飞行器从点 i 飞到点 j，垂直误差应满足约束：

$$(1 - I_i)v_i + \delta w_{ij} - v_j \leq (1 - x_{ij})M \quad (4.1.5)$$

(6) 由该问题描述可知有约束：

$$v_i \leq \alpha_1, \quad h_i \leq \alpha_2, \quad i \in V \quad (4.1.6)$$

(7) 由该问题描述可知有约束：

$$v_i \leq \beta_1, \quad h_i \leq \beta_2, \quad i \in H \quad (4.1.7)$$

此外，注意到该题不仅要使航迹长度尽可能小，还要使得经过校正区域进行校正的次数尽可能少。因此，我们考虑的目标函数变为：

$$\min \sum_{(i,j) \in E} w_{ij} x_{ij} + K \sum_{(i,j) \in E} x_{ij}$$

其中 K 为顶点个数的惩罚系数。

综合上述公式，对问题 1 我们可以建立如下模型：

$$\begin{aligned} & \min \sum_{(i,j) \in E} w_{ij} x_{ij} + K \sum_{(i,j) \in E} x_{ij} \\ & s.t. \begin{cases} \sum_{(i,j) \in E} x_{ij} = \sum_{(j,k) \in E} x_{jk}, \forall j \neq A, B \\ \sum_{(A,j) \in E} x_{Aj} = 1, \quad \sum_{(j,A) \in E} x_{jA} = 0 \\ \sum_{(j,B) \in E} x_{jB} = 1, \quad \sum_{(B,j) \in E} x_{Bj} = 0 \\ I_i h_i + \delta w_{ij} - h_j \leq (1 - x_{ij})M, \quad (i, j) \in E \\ (1 - I_i)v_i + \delta w_{ij} - v_j \leq (1 - x_{ij})M, \quad (i, j) \in E \\ v_i \leq \alpha_1, \quad h_i \leq \alpha_2, \quad i \in V \\ v_i \leq \beta_1, \quad h_i \leq \beta_2, \quad i \in H \end{cases} \end{aligned}$$

4.1.3 模型求解

首先，根据附件 1 和附件 2 的数据集，如果我们直接把所有边直接带入模型中求解，经计算得出附件 1 的变量数为 374544，附件 2 的变量数为 106929。可以看出，如果我们不对数据做预处理，模型中的变量数是巨大的，这会影响求解的速度。因此，我们要对变量（边）进行“剪枝”。假设飞行器从 i 点飞到 j 点，其中 i 点坐标为 $(0, 0)$ ，可知 j 点坐标为 $(\delta w_{i,j}, \delta w_{i,j})$ 。若 $j \in V$ ，则要满足垂直校正条件 $\delta w_{i,j} \leq \min(\alpha_1, \alpha_2)$ ；若 $j \in H$ ，则要满足水平校正条件 $\delta w_{i,j} \leq \min(\beta_1, \beta_2)$ 。这意味着飞机每飞过一定距离必须要经过一个校正点，因此垂直或水平校正点之间的路线距离不得超过一定范围。比如，对附件 1 的数据来讲，

$$\min(\alpha_1, \alpha_2) = 15, \delta = 0.001.$$

因此，垂直修正点之间边的长度不得超过 $15/0.001 = 15000\text{m}$ 。同样，经计算水平修正点之间边的长度不得超过 20000m 。此外，到达终点时垂直误差和水平误差均应小于 $\theta = 30$ 个单位，因此，在距离终点 30000m 以内的校正点校正结束之后，可直接飞向 B 点。由这些计算结果，在附件 1 数据集中，我们可以把垂直校正点之间的超过 15000m 的边剪掉，把水平校正点之间的超过 20000m 的边剪掉，并且把以 B 点为终点的超过 30000m 的边剪掉。

经过上述变量“剪枝”操作，附件 1 的变量数降为 31266，而未进行预处理的变量数为 375769。可以看出，“剪枝”预处理大大减少了变量个数，从而加快了模型求解。同样地，我们也对附件 2 的数据集进行了“剪枝”操作，计算方法和附件 1 类似，在此就不再赘述。经过“剪枝”操作，附件 2 的变量数降为 5483，而未进行预处理的变量数为 106929。可以看出，“剪枝”预处理大大减少了附件 2 的变量个数，从而加快了模型求解。

在数据预处理结束之后，我们把附件 1 和附件 2 “剪枝”后的变量带入模型中，利用 Gurobi 优化软件进行求解，可以分别得到两个数据集的航迹规划结果。其中我们把顶点个数的惩罚系数 K 取为 0 和 10000。当 K 取 0 时，表示距离优先的情况；当 K 取 10000 时，表示顶点个数优先的情况。对于附件 1 的数据，通过采用不同的 K 值，我们可以得到如下表格：

| K | 校正次数 | 航迹长度 (m) |
|-------|------|----------|
| 10000 | 8 | 104861 |
| 0 | 9 | 103517 |

根据附件 1 的数据，我们取 K 为 10000，从而保证模型能够兼顾航迹长度和校正点个数。

对于附件 2 的数据，当 K 取 0 和 K 取 10000 时，我们得到的结果是一样的，因此我们可取 K 为 0 或 10000。

我们利用 Gurobi 优化软件求解，下面我们分别展示了对附件 1 和附件 2 数据集的求解结果，gap 为 0，即我们所得解为最优解，且附件 1 和附件 2 的求解时间分别为 15.11 秒和 0.63 秒。

```
Explored 3418 nodes (51510 simplex iterations) in 15.11 seconds
Thread count was 8 (of 8 available processors)

Solution count 3: 194861 194936 224264

Optimal solution found (tolerance 1.00e-04)
Best objective 1.948610649820e+05, best bound 1.948610649820e+05, gap 0.0000%
```

```

Explored 1 nodes (702 simplex iterations) in 0.59 seconds
Thread count was 8 (of 8 available processors)

Solution count 3: 109342 111957 117294

Optimal solution found (tolerance 1.00e-04)
Best objective 1.093422806027e+05, best bound 1.093422806027e+05, gap 0.0000%

```

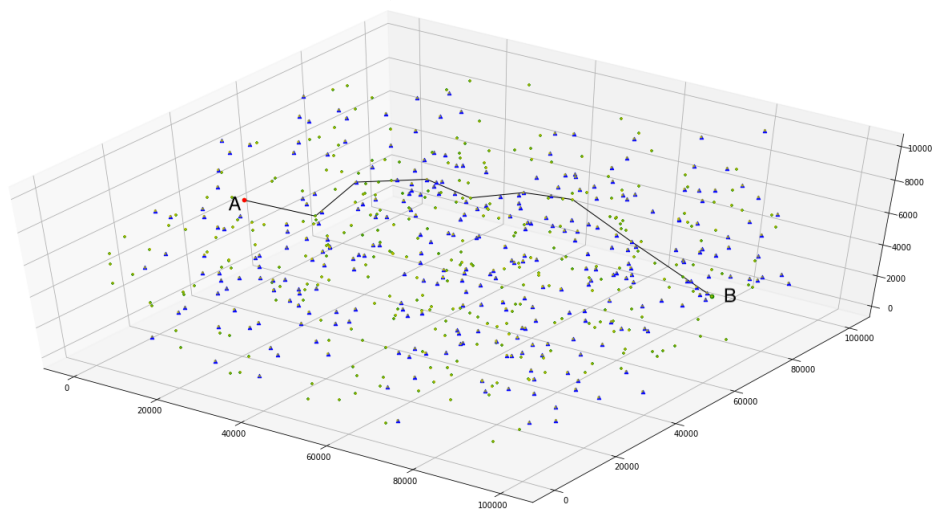
下面我们给出附件 1 的航迹规划结果表 1.1 和航迹规划路径图 1.1：

表 1.1 航迹规划结果表

| 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 | 0.0000 | 0.0000 | 出发点 A |
| 503 | 13.38791985 | 13.38791985 | 11 |
| 294 | 10.18081072 | 23.56873058 | 01 |
| 91 | 17.53551211 | 7.354701386 | 11 |
| 607 | 8.353018096 | 15.70771948 | 01 |
| 540 | 21.38590334 | 13.03288524 | 11 |
| 250 | 11.42932512 | 24.46221036 | 01 |
| 340 | 24.19844923 | 12.76912411 | 11 |
| 277 | 12.00237598 | 24.77150009 | 01 |
| 612 | 28.35328046 | 16.35090448 | 终点 B |

其中，11 表示垂直误差校正成功，01 表示水平误差校正成功。

图 1.1 航迹规划路径图



其中 A 点为出发点，B 点为终点，绿色点表示垂直校正点，蓝色三角点表示水平校正点，黑线表示飞行航迹。

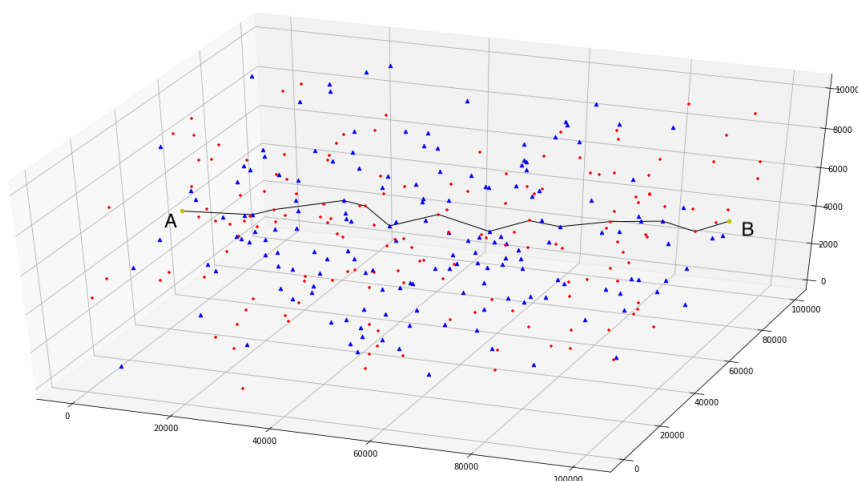
接下来我们给出附件 2 的航迹规划结果表 1.2 和航迹规划路径图 1.2：

表 1.2 航迹规划结果表

| 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 | 0.0000 | 0.0000 | 出发点 A |
| 163 | 13.2879 | 13.2879 | 01 |
| 114 | 18.6221 | 5.3342 | 11 |
| 8 | 13.9220 | 19.2561 | 01 |
| 309 | 19.4463 | 5.5243 | 11 |
| 305 | 5.9687 | 11.4930 | 01 |
| 123 | 15.1731 | 9.2044 | 11 |
| 45 | 10.0062 | 19.2106 | 01 |
| 160 | 17.4913 | 7.4851 | 11 |
| 92 | 5.7762 | 13.2613 | 01 |
| 93 | 15.2609 | 9.4847 | 11 |
| 61 | 9.8342 | 19.3189 | 01 |
| 292 | 16.3881 | 6.5539 | 11 |
| 326 | 6.9605 | 13.5144 | 终点 B |

其中，11 表示垂直误差校正成功，01 表示水平误差校正成功。

图 1.2 航迹规划路径图



其中 A 点为出发点，B 点为终点，红色点表示垂直校正点，蓝色三角点表示水平校正点，黑线表示飞行航迹。

4.1.4 结果展示以及有效性和复杂性分析

经过求解，得到附件 1 数据集经过校正点数最少的最短路径为 0->503->294->91->607->540->250->340->277->612，路径长为 104861m，校正次数为 8，有 4 个垂直校正点，4 个水平校正点；

附件 2 数据集经过校正点数最少的最短路径为 0->163->114->8->309->305->123->45->160->92->93->61->292->326，路径长为 109342m，校正次数为 12，有 6 个

垂直校正点，6 个水平校正点。

本题采用 0-1 混合整数规划求解，设 N 为数据集所给的点的数量，那么理论上问题的求解变量规模为 N^2 。然而通过分析，我们可以发现，通过适当的剪枝，可以大大的降低求解难度。例如在数值求解里面，我们通过将第一个数据集 612^2 的边集变量规模降到了 31266，以及第二个数据集的 326^2 的边集变量规模降到了 5483，将原问题的边集变量规模降到了 5%-8% 的水平。而 h, j 的变量个数只跟点的数量相关，因此两个数据集总的变量个数分别为 31266 个 0-1 变量、1224 个连续变量（附件 1）和 5483 个 0-1 变量、652 个连续变量。总的来说，变量规模控制在 $N^2/10$ 个 0-1 变量和 $2N$ 个连续变量以内。

而在问题约束上，我们的约束数量的为 $O(|E|)$ ， E 为前面的边集变量。并且在这个问题上，通过预估我们可以发现，边集解是稀疏的，也就是说虽然变量规模很大，但是在最优解中大部分的 0-1 变量为 0，只有不超过 20 个变量为 1，因此问题求解相对简单。在这个 0-1 混合整数规划的求解上，我们调用了 Gurobi 求解，附件 1 数据只需 15.11 秒即可得出精确解，附件 2 数据只需 0.3 秒即可得到精确解。

综上，在本题中，我们通过求解数学规划，利用 Gurobi 求解，能够在极短的时间内分别得到两个数据集的最优路径。

4.2 问题 2

4.2.1 问题分析

该题加入了条件 (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 r 为 200m。因此在该题中我们要考虑转弯对航迹的影响。

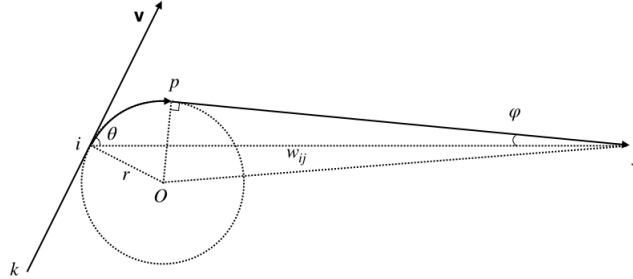


图 3 飞行器转弯示意图

如图 3 所示，飞行器要从 i 点飞到 j 点， i 点初速度方向 \vec{v} 与 i, j 之间的连线夹角为 θ ，此时飞行器的飞行轨迹为圆 O 的 \widehat{ip} 再加上 p, j 之间的连线，其中 p, j 之间的连线为圆 O 的切线， p 点为切点，显然这是图中飞行器的最优飞行轨迹。我们要在这种情况下给出飞行器的航迹使得航迹尽可能短和校正次数尽可能少。由附件 1 和附件 2 的数据可看出， k 点到 i 点的距离一般很大，因此我们可以假定角 φ 为 0。此时我们可得 $\widehat{ip} = \theta r$ ，且 \vec{v} 与 \vec{ki} 之间的方向相同，因此角 θ 可由 \vec{ki} 和 \vec{ij} 求得。设 u_{kij} 为飞行器由 k 出发经过 i 到 j 的路径所得到的 i, j 之间的路径长度，我们可以预先求得所有的 u_{kij} ，并通过一定的“剪枝”算法处理，利用数学规划求得最优值。

4.2.2 模型建立

由余弦定理，我们可得：

$$Oj^2 = r^2 + w_{ij}^2 - 2rw_{ij} \sin \theta.$$

由勾股定理可得：

$$pj^2 = Oj^2 - r^2.$$

此外还有：

$$\widehat{ip} = r(\theta + \varphi).$$

由于 $w_{ij} \gg r$ ，故有 $\varphi \rightarrow 0$ 。因此我们可得

$$\begin{aligned} dis(i, j) &= \widehat{ip} + pj = r\theta + (Oj^2 - r^2)^{1/2} \\ &= r\theta + (r^2 + w_{ij}^2 - 2rw_{ij} \sin \theta - r^2)^{1/2} \\ &= r\theta + (w_{ij}^2 - 2rw_{ij} \sin \theta)^{1/2} \end{aligned}$$

其中 $dis(i, j)$ 表示该题中 i 点到 j 点之间的距离。

由前面问题分析可知

$$\cos \theta = \frac{\langle \vec{ki}, \vec{ij} \rangle}{|\vec{ki}| |\vec{ij}|}$$

因此，我们可得：

$$dis(i, j) = r \arccos \frac{\langle \vec{ki}, \vec{ij} \rangle}{|\vec{ki}| |\vec{ij}|} + \left(w_{ij}^2 - 2rw_{ij} \sin \arccos \frac{\langle \vec{ki}, \vec{ij} \rangle}{|\vec{ki}| |\vec{ij}|} \right)^{1/2}$$

由该公式我们可以预先求得所有的 u_{kij} 。

问题 1 中的约束条件 (4.1.1), (4.1.2), (4.1.3), (4.1.6) 和 (4.1.7) 在该问题中仍然需要，因此此处不再赘述。根据问题 2 提出的要求，增加更新的约束条件如下：

(1) 飞行器从点 i 飞到点 j，水平误差应满足约束：

$$I_i h_i + \delta u_{kij} - h_j \leq (2 - x_{ki} - x_{ij})M, \quad (4.2.1)$$

其中 M 为给定的很大的正数，下同。

(2) 飞行器从点 i 飞到点 j，垂直误差应满足约束：

$$(1 - I_i)v_i + \delta u_{kij} - v_j \leq (2 - x_{ki} - x_{ij})M \quad (4.2.2)$$

(3) 对飞行起点 A 有如下约束：

$$\delta w_{Aj} - h_j \leq (1 - x_{Aj})M \quad (4.2.3)$$

$$\delta w_{Aj} - v_j \leq (1 - x_{Aj})M \quad (4.2.4)$$

在该题中，更新后的目标函数为

$$\min \sum_{\substack{(k,i) \in E \\ (i,j) \in E}} u_{kij}(x_{ki} \wedge x_{ij}) + \sum_{(A,i) \in E} w_{Ai}x_{Ai} + K \sum_{(i,j) \in E} x_{ij}$$

其中 $x_{ki} \wedge x_{ij} = 1$ 当且仅当 $x_{ki} = 1$ 和 $x_{ij} = 1$ ，此外 K 为顶点个数的惩罚系数。

综合上述公式，对问题 2 我们可以建立如下模型：

$$\min \sum_{\substack{(k,i) \in E \\ (i,j) \in E}} u_{kij}(x_{ki} \wedge x_{ij}) + \sum_{(A,i) \in E} w_{Ai}x_{Ai} + K \sum_{(i,j) \in E} x_{ij}$$

$$s.t. \begin{cases} \sum_{(i,j) \in E} x_{ij} = \sum_{(j,k) \in E} x_{jk}, \forall j \neq A, B \\ \sum_{(A,j) \in E} x_{Aj} = 1, \sum_{(j,A) \in E} x_{jA} = 0 \\ \sum_{(j,B) \in E} x_{jB} = 1, \sum_{(B,j) \in E} x_{Bj} = 0 \\ I_i h_i + \delta u_{kij} - h_j \leq (2 - x_{ki} - x_{ij})M \\ (1 - I_i)v_i + \delta u_{kij} - v_j \leq (2 - x_{ki} - x_{ij})M \\ \delta w_{Aj} - h_j \leq (1 - x_{Aj})M \\ \delta w_{Aj} - v_j \leq (1 - x_{Aj})M \\ v_i \leq \alpha_1, \quad h_i \leq \alpha_2, \quad i \in V \\ v_i \leq \beta_1, \quad h_i \leq \beta_2, \quad i \in H \end{cases}$$

4.2.3 模型求解

首先，我们对附件 1 和附件 2 的数据集做“剪枝”预处理。通过观察问题 1 的结果，我们发现飞行器在从 A 点飞到 B 点的过程中，最优路径较为平缓，而且并不会走“回头路”。记以 A 为起点，B 为终点的向量为 \overrightarrow{AB} 。也就是说，在数据集中，若某条有向边和 \overrightarrow{AB} 做内积之后，得到的数小于 0，那我们就可以把这条边剪掉。此外，我们要求路径长度尽可能短，而且要较为平缓，可以考虑所求路径被以 AB 为轴，半径为 R 的圆柱体包住，在圆柱体外面的点都被剪掉。因为在第一题中，最优路径上的点距离直线 AB 的最大值处于 7000 到 7500 之间，这里我们不妨取 R 为 10000。如图 4 所示：

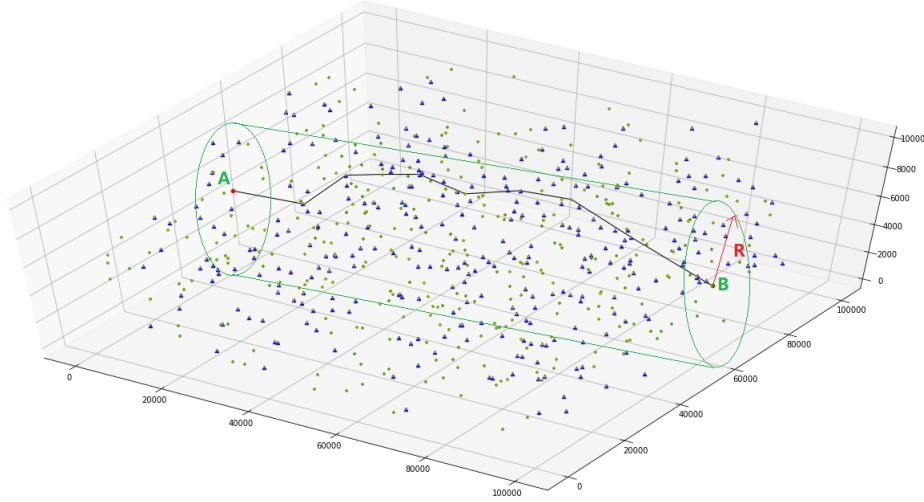


图 4 数据预处理示意图

经过“剪枝”操作，附件 1 的变量 x_{ij} 的数目由未处理的 31266 降为 2488， u_{kij} 的数目由未处理的 1972536 降为 44578；附件 2 的变量 x_{ij} 的数目由未处理的 5483 降为 447， u_{kij} 的数目由未处理的 191670 降为 2625；可以看出，“剪枝”预处理大大减少了附件 1 和附件 2 的约束和变量个数，从而加快了模型求解。

在数据预处理结束之后，我们把附件 1 和附件 2 “剪枝”后的变量带入模型中，利用 Gurobi 优化软件进行求解，可以分别得到两个数据集的航迹规划结果。其中我们把顶点个数的惩罚系数 K 取为 0 和 10000。当 K 取 0 时，表示距离优先的情况；当 K 取 10000 时，表示顶点个数优先的情况。对于附件 1 的数据，通

过采用不同的 K 值，我们可以得到如下表格：

| K | 校正次数 | 航迹长度 (m) |
|-------|------|----------|
| 10000 | 8 | 104874 |
| 0 | 9 | 103536 |

根据附件 1 的数据，我们取 K 为 10000，从而保证模型的解在最小化航迹长度的同时，兼顾校正点个数。

对于附件 2 的数据，当 K 取 0 和 K 取 10000 时，我们得到的结果是一样的，因此我们可取 K 为 0 或 10000。

我们利用 Gurobi 优化软件求解，下面我们分别展示了对附件 1 和附件 2 数据集的求解结果，gap 为 0，即我们所得解为最优解，且附件 2 的求解时间仅为 0.43 秒。

```
Explored 37266 nodes (5475927 simplex iterations) in 2413.72 seconds
Thread count was 8 (of 8 available processors)

Solution count 10: 194874 195725 196484 ... 230989

Optimal solution found (tolerance 1.00e-04)
Best objective 1.948742532602e+05, best bound 1.948742532602e+05, gap 0.0000%
```

```
Explored 374 nodes (3979 simplex iterations) in 0.43 seconds
Thread count was 8 (of 8 available processors)

Solution count 9: 109455 110790 111105 ... 128424

Optimal solution found (tolerance 1.00e-04)
Best objective 1.094549968861e+05, best bound 1.094549968861e+05, gap 0.0000%
```

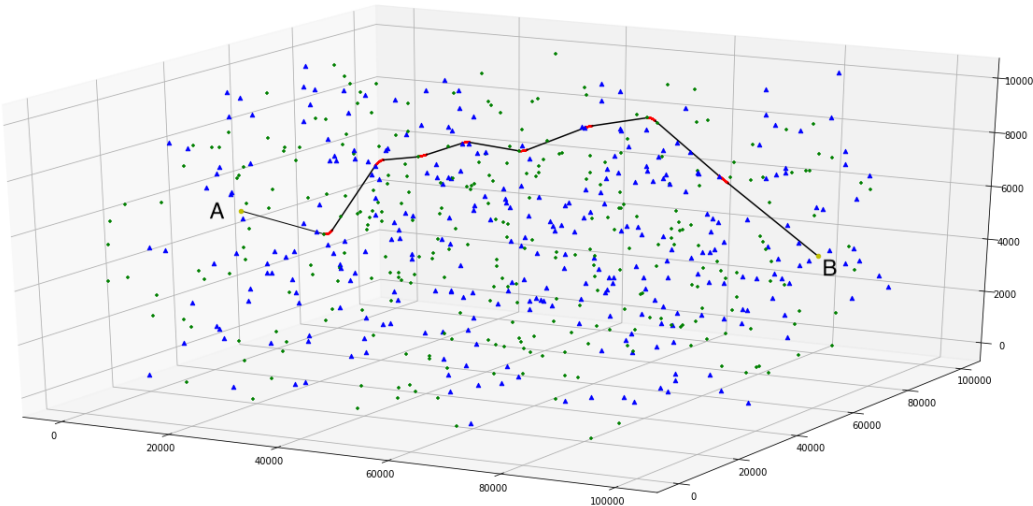
下面我们给出附件 1 的航迹规划结果表 2.1 和航迹规划路径图 2.1：

表 2.1 航迹规划结果表

| 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 | 0.0000 | 0.0000 | 出发点 A |
| 503 | 13.3879 | 13.3879 | 11 |
| 294 | 10.1880 | 23.5759 | 01 |
| 91 | 17.5432 | 7.3552 | 11 |
| 607 | 8.3533 | 15.7085 | 01 |
| 540 | 21.3867 | 13.0334 | 11 |
| 250 | 11.4305 | 24.4639 | 01 |
| 340 | 24.2001 | 12.7696 | 11 |
| 277 | 12.0054 | 24.7749 | 01 |
| 612 | 28.3563 | 16.3509 | 终点 B |

其中，11 表示垂直误差校正成功，01 表示水平误差校正成功。

图 2.1 航迹规划路径图



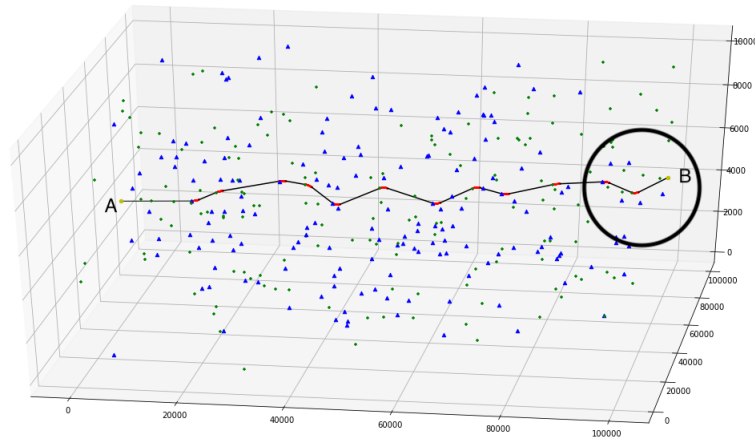
其中 A 点为出发点，B 点为终点，绿色点表示垂直校正点，蓝色三角点表示水平校正点，黑线表示飞行直线，红线表示飞行圆弧。

表 2.2 航迹规划结果表

| 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 | 0.0000 | 0.0000 | 出发点 A |
| 163 | 13.28789761 | 13.28789761 | 01 |
| 114 | 18.62808847 | 5.340190855 | 11 |
| 8 | 13.955761 | 19.29595185 | 01 |
| 309 | 19.51577274 | 5.560011738 | 11 |
| 305 | 5.968723685 | 11.52873542 | 01 |
| 123 | 15.18928879 | 9.220565106 | 11 |
| 45 | 10.00792793 | 19.22849303 | 01 |
| 160 | 17.4933707 | 7.485442778 | 11 |
| 92 | 5.77632763 | 13.26177041 | 01 |
| 93 | 15.26138823 | 9.485060597 | 11 |
| 61 | 9.834574257 | 19.31963485 | 01 |
| 292 | 16.38868469 | 6.554110432 | 11 |
| 326 | 6.978403274 | 13.53251371 | 终点 B |

其中，11 表示垂直误差校正成功，01 表示水平误差校正成功。

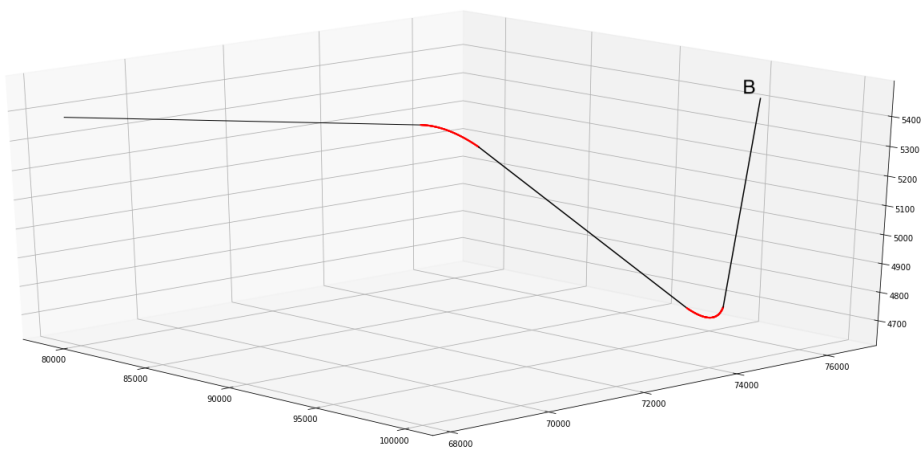
图 2.2 航迹规划路径图



其中 A 点为出发点，B 点为终点，绿色点表示垂直校正点，蓝色三角点表示水平校正点，黑线表示飞行直线，红线表示飞行圆弧。

因为在该图中红色圆弧不是十分明显，所以我们选取其中一段进行放大以便查看。如下图所示：

图 2.3 航迹规划路径图（放大）



4.2.4 结果展示以及有效性和复杂性分析

经过求解，得到附件 1 数据集经过校正点数最少的最短路径为 0->503->294->91->607->540->250->340->277->612，路径长为 104874m，校正次数为 8，有 4 个垂直校正点，4 个水平校正点；

附件 2 数据集经过校正点数最少的最短路径为 0->163->114->8->309->305->123->45->160->92->93->61->292->326，路径长为 109455m，校正次数为 12，有 6 个垂直校正点，6 个水平校正点。

延续第一题的模型框架，本题依旧采用 0-1 混合整数规划求解，与第一题不同的是，我们在建模中使用的变量数量远大于第一题，倘若不采取适当的剪枝策略，将使得结果非常的难以求解。因此在第二题中，我们根据第一题的结果采用了合理的剪枝方案。下面我们提到的剪枝对比是将问题二剪枝结果和问题一

的剪枝结果进行对比。我们将附件 1 的变量 x_{ij} 的数目由 31266 降到 2488, u_{kij} 的数目由 1972536 降到 44578; 附件 2 的变量 x_{ij} 的数目由未处理的 5483 降到 447, u_{kij} 的数目由未处理的 191670 降到 2658。因此, 两个数据集总的变量个数分别为 2488 个 0-1 变量、1224 个连续变量 (附件 1) 和 447 个 0-1 变量、652 个连续变量。约束数量大致为 2 倍的 u_{kij} 数量。与第一题类似的是, 在这个问题里, 最优解中大部分的 0-1 变量依旧为 0, 因此我们得到的依旧是稀疏解, 因此问题求解依旧不难。尤其是 Excel2, 由于点的个数相对较少, 我们在 0.43 秒就得到了精确解 (Gurobi 的 gap 为 0)。

不过由于在本题建模时, 我们对模型做了近似, 因此在具体的模型路径长度和飞行角度计算上, 仍存在一定的误差。但是从我们的解来看, 任意两个校正点之间的距离最小不低于 4000m, 远大于转弯圆弧半径, 因此我们的简化依旧是有意义的。

综上, 在本题中, 我们通过对模型做巧妙的近似, 继而做合理的剪枝, 求解数学规划, 利用 Gurobi 求解, 能够在较短的时间内分别得到两个数据集的最优路径。

4.3 问题 3

4.3.1 问题分析

和问题 1 不同的是, 在该问题中, 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), 因此, 我们需要规划出一条可行路径, 使得由 A 点出发成功到达 B 点的概率尽可能大。首先, 我们来分析飞行器飞行路径可能出现的情况。在飞行器的飞行路径上一共存在三种点: 第一种是正常校正点; 第二种是校正可能会出现问题, 但我们可以通过缩短该点到后面点的距离来保证飞行器能正常飞行的点, 即使该点校正失败, 我们依然能够保证飞行器能够顺利到达终点; 第三种点是 20% 失败点, 即校正可能会出现问题的点, 并且倘若出现问题, 飞行器便无法到达终点。而我们的目标是使飞行器成功到达终点的概率尽可能大, 因此, 在飞行路径上, 第三种点出现的个数要尽可能少。因此, 我们可以在问题 1 模型的基础上对模型进行更新。其中相应的约束条件要做改变, 最后对我们更新的模型进行求解。

4.3.2 模型建立

问题 1 中的约束条件 (4.1.1), (4.1.2), (4.1.3), (4.1.6) 和 (4.1.7) 在该问题中仍然需要, 因此此处不再赘述。根据问题 3 提出的要求, 增加更新的约束条件如下:

飞行器从点 i 飞到点 j , 在问题 3 的分析中所提到的第一种点水平和垂直误差应分别满足约束 (4.1.4) 和 (4.1.5):

$$I_i h_i + \delta w_{ij} - h_j \leq (1 - x_{ij})M$$

$$(1 - I_i)v_i + \delta w_{ij} - v_j \leq (1 - x_{ij})M$$

第二种点水平和垂直误差应分别满足约束:

$$(1 - I_i) \min(h_i, 5) + I_i h_i + \delta w_{ij} - h_j \leq (1 - x_{ij})M$$

$$I_i \min(v_i, 5) + (1 - I_i)v_i + \delta w_{ij} - v_j \leq (1 - x_{ij})M$$

第三种点水平和垂直误差应分别满足约束:

$$I_i h_i + \delta w_{ij} - h_j \leq (1 - x_{ij})M$$

$$(1 - I_i)v_i + \delta w_{ij} - v_j \leq (1 - x_{ij})M$$

综合这三种点的约束我们得到如下约束表达式：

$$\Lambda_i(1 - I_i) \min(h_i, 5) - \Psi_i \Lambda_i(1 - I_i) \min(h_i, 5) + I_i h_i + \delta w_{ij} - h_j \leq (1 - x_{ij})M \quad (4.3.1)$$

$$\Lambda_i I_i \min(v_i, 5) - \Psi_i \Lambda_i I_i \min(v_i, 5) + (1 - I_i)v_i + \delta w_{ij} - v_j \leq (1 - x_{ij})M \quad (4.3.2)$$

其中 Λ_i 为在 i 点校正是否存在问题的示性变量（1，存在问题；0，正常校正）， Ψ_i 为 i 点是否是第三种点的示性变量（1，是第三种点；0，不是第三种点）。

在该题中，更新后的目标函数为

$$\min \sum_{(i,j) \in E} w_{ij} x_{ij} + K \sum_{(i,j) \in E} x_{ij} + C \sum_i \Psi_i$$

其中 C 为第三种点在最优路径中所占数目的惩罚系数。

综合上述公式，对问题 3 我们可以建立如下模型：

$$\begin{aligned} \min \quad & \sum_{(i,j) \in E} w_{ij} x_{ij} + K \sum_{(i,j) \in E} x_{ij} + C \sum_i \Psi_i \\ \text{s.t.} \quad & \begin{cases} \sum_{(i,j) \in E} x_{ij} = \sum_{(j,k) \in E} x_{jk}, \forall j \neq A, B \\ \sum_{(A,j) \in E} x_{Aj} = 1, \sum_{(j,A) \in E} x_{jA} = 0 \\ \sum_{(j,B) \in E} x_{jB} = 1, \sum_{(B,j) \in E} x_{Bj} = 0 \\ (1 - \Psi_i) \Lambda_i(1 - I_i) \min(h_i, 5) + I_i h_i + \delta w_{ij} - h_j \leq (1 - x_{ij})M, \quad (i, j) \in E \\ (1 - \Psi_i) \Lambda_i I_i \min(v_i, 5) + (1 - I_i)v_i + \delta w_{ij} - v_j \leq (1 - x_{ij})M, \quad (i, j) \in E \\ v_i \leq \alpha_1, \quad h_i \leq \alpha_2, \quad i \in V \\ v_i \leq \beta_1, \quad h_i \leq \beta_2, \quad i \in H \end{cases} \end{aligned}$$

4.3.3 模型求解

在该题中，我们对附件 1 和附件 2 的数据集做与问题 1 模型求解时的相同的变量“剪枝”处理。在数据预处理结束之后，我们把附件 1 和附件 2 “剪枝”后的变量带入模型中，利用 Gurobi 优化软件进行求解，可以分别得到两个数据集的航迹规划结果。首先在目标函数中我们把顶点个数的惩罚系数 K 取为 0 和 10000。当 K 取 0 时，表示距离优先的情况；当 K 取 10000 时，表示顶点个数优先的情况。对于附件 1 的数据，通过采用不同的 K 值，我们可以得到如下表格：

| K | 校正次数 | 航迹长度 (m) |
|-------|------|----------|
| 10000 | 9 | 104864 |
| 0 | 10 | 104827 |

根据附件 1 的数据，我们取 K 为 10000，从而保证模型的解能够兼顾最小化航迹长度和校正点个数最小化两个要求。

对于附件 2 的数据，当 K 取 0 和 K 取 10000 时，我们得到的结果是一样的，因此我们可取 K 为 0 或 10000。因此对附件 1 和附件 2 的数据，我们把目标函数中的 K 取为 10000， C 取为 100000。这样可保证目标函数为线性，避免了幂次运算。

我们利用 Gurobi 优化软件求解，下面我们分别展示了对附件 1 和附件 2 数据集的求解结果，可看出 gap 为 0，即我们所得为最优解，且附件 1 和附件 2 的求解时间分别为 11.13 秒和 0.99 秒。从最优解我们可以看出，这两条规划的路线，由 A 到 B 的概率都是 100%。

```
Explored 5385 nodes (67481 simplex iterations) in 11.13 seconds
Thread count was 8 (of 8 available processors)

Solution count 9: 204864 204864 205874 ... 407128

Optimal solution found (tolerance 1.00e-04)
Best objective 2.048643839390e+05, best bound 2.048643839390e+05, gap 0.00000%
```

```
Explored 256 nodes (3809 simplex iterations) in 0.99 seconds
Thread count was 8 (of 8 available processors)

Solution count 7: 161650 162199 170686 ... 358957

Optimal solution found (tolerance 1.00e-04)
Best objective 1.616504102164e+05, best bound 1.616504102164e+05, gap 0.00000%
```

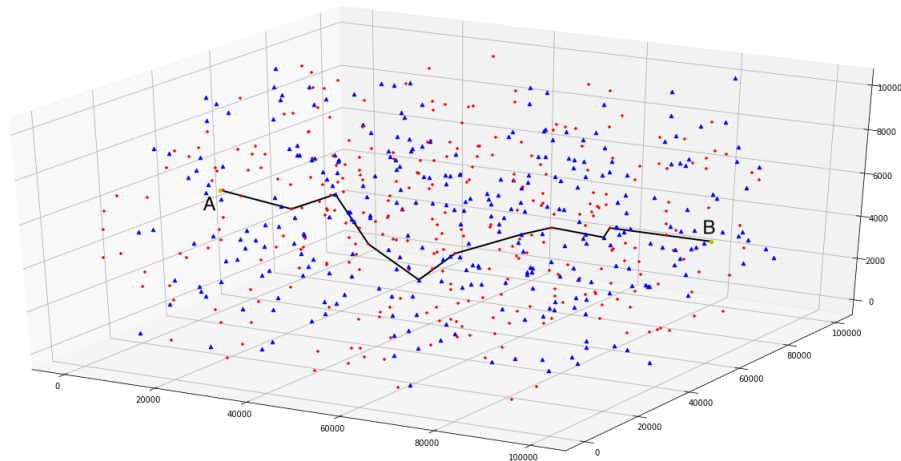
下面我们给出附件 1 的航迹规划结果表 3.1 和航迹规划路径图 3.1：

表 3.1 航迹规划结果表

| 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 | 0.0000 | 0.0000 | 出发点 A |
| 503 | 13.3879 | 13.3879 | 12 |
| 69 | 13.8073 | 22.1953 | 02 |
| 506 | 21.6756 | 12.8682 | 12 |
| 371 | 15.6155 | 23.4837 | 01 |
| 183 | 22.6538 | 7.0383 | 11 |
| 194 | 13.6115 | 20.6498 | 01 |
| 450 | 19.5886 | 5.9771 | 11 |
| 286 | 12.7713 | 18.7485 | 01 |
| 485 | 13.9875 | 1.2162 | 11 |
| 612 | 23.5709 | 24.7871 | 终点 B |

其中，11 表示垂直误差校正成功，01 表示水平误差校正成功，12 表示垂直误差校正不成功，22 表示水平误差校正不成功。

图 3.1 航迹规划路径图



其中 A 点为出发点，B 点为终点，红色点表示垂直校正点，蓝色三角点表示水平校正点，黑线表示飞行航迹。

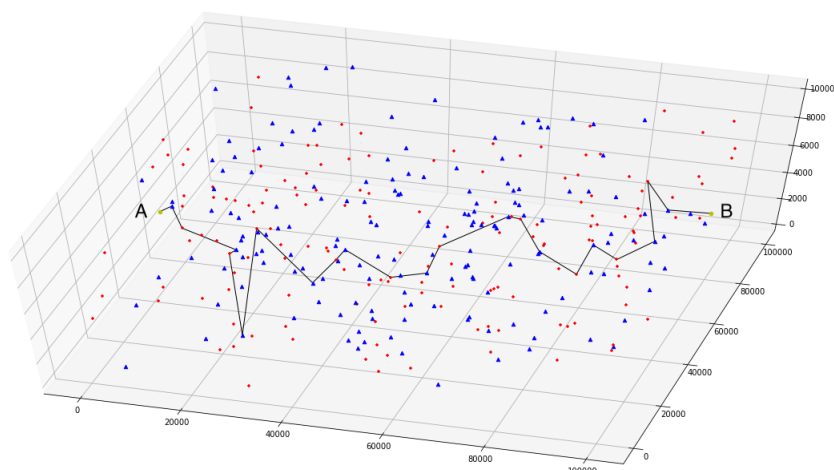
接下来我们给出附件 2 的航迹规划结果表 3.2 和航迹规划路径图 3.2：

表 3.2 航迹规划结果表

| 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 | 0.0000 | 0.0000 | 出发点 A |
| 169 | 9.2705 | 9.2705 | 01 |
| 322 | 13.4186 | 4.1481 | 11 |
| 270 | 11.3408 | 15.4889 | 01 |
| 89 | 18.8916 | 7.5508 | 11 |
| 236 | 10.1274 | 17.6782 | 01 |
| 132 | 19.8316 | 9.7042 | 11 |
| 53 | 10.2592 | 19.9634 | 01 |
| 112 | 15.4621 | 5.2029 | 11 |
| 268 | 2.1572 | 7.3601 | 01 |
| 250 | 11.7639 | 9.6067 | 11 |
| 243 | 6.9589 | 16.5656 | 01 |
| 73 | 10.5018 | 3.5428 | 11 |
| 249 | 12.8462 | 16.3890 | 01 |
| 274 | 15.6869 | 2.8407 | 11 |
| 12 | 6.4367 | 9.2774 | 01 |
| 216 | 14.2391 | 7.8024 | 11 |
| 16 | 4.2163 | 12.0186 | 01 |
| 282 | 11.6561 | 7.4398 | 11 |
| 141 | 8.1003 | 15.5401 | 01 |
| 291 | 13.5862 | 5.4859 | 11 |
| 161 | 6.4661 | 11.9520 | 02 |
| 326 | 16.6126 | 15.1464 | 终点 B |

其中，11 表示垂直误差校正成功，01 表示水平误差校正成功，12 表示垂直误差校正不成功，22 表示水平误差校正不成功。

图 3.2 航迹规划路径图



其中 A 点为出发点，B 点为终点，红色点表示垂直校正点，蓝色三角点表示水平校正点，黑线表示飞行航迹。

4.3.4 结果展示以及鲁棒性分析

经过求解，附件 1 的成功概率为 100%，得到附件 1 数据集经过校正点数最少的最短路径为 0->503->69->506->371->183->194->450->286->485->612，路径长为 104864m，校正次数为 9，有 5 个垂直校正点，4 个水平校正点；

附件 2 的成功概率为 100%，数据集经过校正点数最少的最短路径为 0->169->322->270->89->236->132->53->112->268->250->243->73->249->274->12->216->16->282->141->291->161->326，路径长为 161650m，校正次数为 21，有 10 个垂直校正点，11 个水平校正点。

在这个模型的框架下，倘若存在数据集，使得该飞行器不存在 100% 成功的可能，那么我们依然可以通过解这个 0-1 混合整数规划，得到所需的最优解。因此模型具有较好的鲁棒性，可以应对该问题框架下，各种不同类型的数据集。并且在这个模型里，问题变量和约束规模相对较小，因此该模型可以在很短的时间内，求得问题最优解。

5. 模型评价

5.1 模型优点

本文在正确、清楚地分析了题意的基础上，建立了合理、科学的最优路径计算模型，为获得最优解提供了条件。

对于模型的建立困难有三个方面的：一是如何合理的建立线性模型，由于 Gurobi 解线性模型和非线性模型的时间差距很大，因此合理的建立线性模型有利于迅速求解；二是如何合理的剪枝，降低求解难度。在第二个问题的模型建立中，如果我们不做合理的剪枝，数据集一将无法使用 Gurobi 求解，因此我们引入了合理的剪枝方案，提供了求解的可能性。三是如何合理的使用惩罚系数，使得模型能够在 0-1 混合整数规划的框架下完成求解，不需要借助任何的启发式算法求解。

总的来说，我们的第一问中求得的是题中所需的精确的最优解，而第二问通过合理的近似与剪枝，我们得到了近似后的最优解，在第三问中，我们给出了

具有普适性的方法，也就是说，即便不存在满足百分百能到达终点的路，我们依然能够根据我们所建立的模型，给出概率最大且满足题一所需的最优路径。

5.2 模型缺点

1. 在第二问中引入过多的变量，导致了维数灾难，虽然通过剪枝算法求得最优解，但仍无法在极短的时间内求得数据集 1 的最优解，使得第二问的数据集一的求解较其他问题相对缓慢。

2. 在第二问中对夹角做了一定的近似，倘若将圆弧半径增大一定的倍数，或者最优解中存在距离较小的校正点，将无法使用这个近似模型，降低了模型的普适性。

6. 参考文献

- [1] 卓金武 王鸿钧 MATLAB 数学建模方法与实践（第三版） 北京航空航天大学出版社:104-113,2017.
- [2] 孙小玲 李端 整数规划 科学出版社:1-25,2010.
- [3] 司守奎 孙玺菁 数学建模算法与应用 国防工业出版社:121-156,2011.
- [4] Wes McKinney Python For Data Analysis:Data Wrangling with Pandas,Numpy and IPython O' RILEY:420-432,2017.
- [5] Gurobi 软件内置参考手册:462-463,2019.