



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 南京理工大学

参赛队号 19102880019

| | |
|------|--------|
| 队员姓名 | 1. 许哲源 |
| | 2. 孙佳星 |
| | 3. 金霄 |

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 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，以最优化理论为基础，对模型目标进行加权求和构建目标函数，利用进化算法具备的高鲁棒性和广泛适用性的能力来分析问题。针对本题的优化问题，一方面希望航迹长度尽可能小，另一方面希望经过校正区域进行校正的次数尽可能少，显然这是个多目标优化问题。我们采用 **Pareto 解集** 的概念，最终得到每个问题的一组最优解即相应多目标优化问题的非支配解集；在构建好目标函数模型后，我们利用 **A* 算法** 的启发式搜索思想，结合 **模拟退火算法** 的思想引入随机函数对航迹路线进行有效搜索；并且我们采用了十分有效的 **路网稀疏策略**，使用满足题设约束条件的球体去对原始密布在三维空间的点进行有效选择下一个可连接的点，并通过 **预先存储** 无约束条件最短路径和搜索过程中产生的若干不同规模的矩阵，加快算法运行速度。

问题一中，根据两个数据集中的数据规划出满足题设中（1）~（7）条件的飞行器的航迹，使其达到航迹长度尽可能小和经过校正区域进行校正的次数尽可能少的两个优化目标，我们使用数据集中的数据点在三维空间中建立拓扑图模型，根据约束条件提出了一种预处理模型，剔除拓扑图中不满足约束无法通行的边。通过有效地预处理和对所使用算法的部分优化后，我们得出了符合条件的 **Pareto 解集**，并且其中效果最好的航迹距离 104065m 仅略高于起点 A 到终点 B 的直线距离，且进行校正的次数接近理想状态下需要校正的次数。具体的计算结果已经按照规定格式存入附件 3 的 Sheet1 和 Sheet2 中。

问题二中，加上条件（8）的约束，飞行器在转弯时不能像第一问中一样即时转弯按照直线方向飞行，需要通过增加圆弧航迹来驶向下一个误差校正点，目标函数同问题一中不变。针对目标，我们提出一种几何模型，每当航迹方向发生改变时，使用两个半径最小即为题设中限制的 200 米的球去紧紧相切，并且一个球体切于航迹原方向所在平面，另一个球体切于新航迹直线方向所在平面，具体的模型构建原因与示意图见第三节中详细介绍。

通过对该约束条件的处理，我们得到了综合效果最好的航迹路线。具体的计算结果已经按照规定格式存入附件 3 的 Sheet3 和 Sheet4 中。

问题三中，在问题一的基础上增加了一个对校正点的限制，原数据集中的校正点有部分是无法完成理想校正情况的（在下文中统称这些点为问题校正点），飞行器在问题校正点进行误差校正时有 80% 的概率校正成功，20% 的概率校正失败。本问中希望飞行器成功到达终点的概率尽可能大，我们在问题一得到的航迹优解集的基础上进行该目标的处理与实现。通过对航迹中遇到的问题校正点个数及其校正情况分四类讨论：（1）若航迹中的问题校正点个数为 0；（2）若航迹中遇到的问题校正点均成功校正；（3）若航迹中仅遇到一个问题校正点校正失败；（4）若航迹中遇到一次以上问题校正点且均校正失败。对这四种情况分别计算了其对应概率，并最后通过加权求和得到某条特定航线的飞行成功率，并找到最优解。具体的计算结果已经按照规定格式存入附件 3 的 Sheet5 和 Sheet6 中。

关键字： 路径优化策略 A* 算法 路网稀疏策略 进化算法 多目标规划 Pareto 解集

1. 问题重述

1.1 引言

航迹规划自从飞行器被发明开始，就一直在任务规划系统中占有重要地位，并且通常是在战场环境下被研究 [1]。在二十世纪八十年代初期，航迹规划基本上依赖规划人员的手工操作。而随着防空技术和体系的不断发展和完善，人工规划的粗糙航迹已难以达到有效突防的目的。同时，由于卫星、通讯技术的快速发展，航迹规划人员获得的信息也愈来愈多，例如通过卫星，无人机等侦察手段可以得到敌方防空阵地的精确信息，如位置、类型、火力圈范围等；通过数字地形图可以获得相关地区的地形信息，比如高山、峡谷和平原等数据。为了最大限度地减小突防飞行器被敌方雷达发现的概率，航迹规划必须在满足飞行器自身性能等约束的前提下，为飞行器规划出超低空地形威胁回避的飞行航迹。因为其中涉及因素太多，各种因素之间需要进行仔细权衡，传统的手工规划方式难以完成如此复杂的任务。

飞行器航迹规划的目的是要找到一条能够保证飞行器安全突防的飞行航迹 [2]，既要尽量减少被敌防空设施捕获和摧毁的概率，又要降低坠毁的概率，同时还必须满足各种约束条件。例如，无人飞行器在战场环境下，需要选择从起始点到目标点的最优飞行轨迹，这是无人飞行器实现自主飞行的技术保障，如果因为定位误差偏离航线，会极大增加其被敌军发现并击落的概率。当飞行器处于复杂环境中时（包括战场、救援、太空），这时地面一次性全局规划已不能满足要求。在这种环境下，要求智能飞行器在飞行过程中实时处理新出现的威胁，依靠实时探测到的环境信息，经过多次规划来确定每一时刻当前已知信息下的最优飞行航迹。

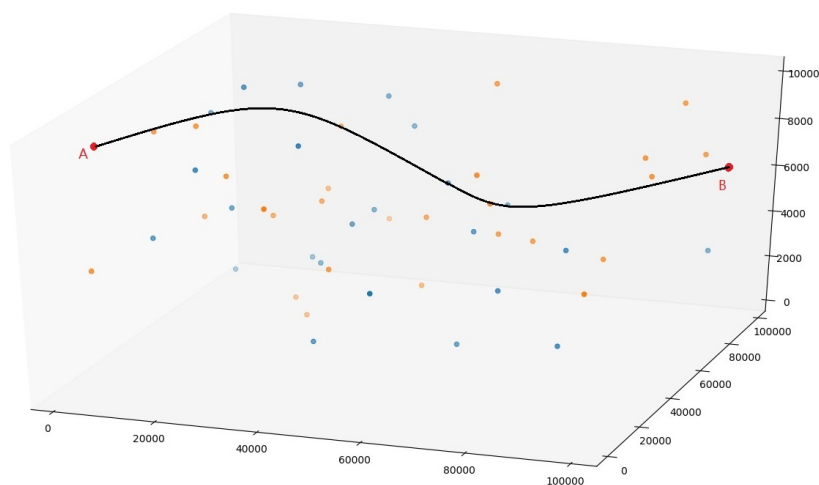


图 1 飞行器航迹规划区域示意图

因此，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。已有的方法可以分为两类 [3]，第一类方法是基于已知信息规划一条初始航迹，然后在碰到预先未知的威胁时局部地修改初始航迹，绕过威胁后重新回到预规划好的初始航迹上；另一类方法是，在发现新威胁时从飞行器当前位置到目标位置之间完全重新规划一条新的最优航迹。这两种方法以优化或计算效率为代价来满足不同的需求。第一种方法只需要少量的计算，因此可以避免实时威胁，但是这种方法不能保证轨道的最佳性。后一种方法显然具有很高的计算成本，因为每次添加新的威胁信息时，都需要重新开始搜索目标位置的完整轨迹。

本文讨论的问题针对于航迹规划中的一种特殊情景。由于一类飞行器的结构限制，在飞行过程中无法准确定位自身，一旦定位误差超过某一限制，那么很可能导致任务失败。

1.2 问题的提出

围绕智能飞行器从 A 点飞到 B 点的航迹规划，本文依次提出如下问题：

- (1) 针对附件 1 和附件 2 中的数据分别规划满足题目条件 (1) ~ (7) 时飞行器的航迹，并且综合考虑以下优化目标：
 - a) 轨迹长度尽可能小；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。
- (2) 针对附件 1 和附件 2 中的数据分别规划满足题目条件 (1) ~ (8) 时飞行器的航迹，并且综合考虑以下优化目标：
 - a) 轨迹长度尽可能小；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。
- (3) 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。

2. 模型的假设

2.1 题目给出的假设

- (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 δ 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 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。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 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) 针对问题二，假设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。
- (9) 针对问题三，假设存在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。

2.2 为简化模型求解追加的假设

- (1) 飞行器在飞行过程中视为质点。
- (2) 飞行过程中不考虑油耗及其他题目条件外的限制因素，不会因为燃料不足等因素而任务失败。
- (3) 为减少航迹距离，飞行器在飞行时从一个校正点径直飞向下一个目标点。
- (4) 不考虑飞行器的最大爬升/俯冲角，即考虑飞行器在垂直平面内上升和下滑的最大角度。
- (5) 在飞行航迹中，不会重复经过同一校正点。
- (6) 为了实现经过的校正点尽可能少，在飞行航线规划中，同类型的校正点不会相邻。
- (7) 为了满足最优目标，在航迹规划中需要保证经过的所有校正点都满足校正条件。

3. 符号说明

| 符号 | 意义 |
|-------------|-----------------------------|
| V | 全部点构成的集合 |
| E | 全部点连线构成有向边的集合 |
| $G(V, E)$ | 由点集 V 和边集 E 构成的拓扑图 |
| $v(i)$ | 编号为 i 的点 |
| $e(i, j)$ | 从编号为 i 的点指向编号为 j 的点的有向边 |
| $vec(i, j)$ | 从编号为 i 的点指向编号为 j 的点的向量 |
| $\ vec\ $ | 向量 vec 的二范数 |
| v_{error} | 垂直误差 (单位 δ) |
| h_{error} | 水平误差 (单位 δ) |
| v_{error} | 垂直误差 (单位 δ) |
| h_{error} | 水平误差 (单位 δ) |

4. 问题一

4.1 问题一分析

问题 1 规划满足题目条件飞行器的航迹，并且综合考虑以下优化目标：

$a)$ 轨迹长度尽可能小； $b)$ 经过校正区域进行校正的次数尽可能少。

我们对问题一进行分析，问题一优化目标是从 A 到 B 航迹最短和停留的校正点数最少。这是一个多目标优化问题，根据常识，这两个目标呈现高度的正相关性。我们猜想在目标一得到优化同时，目标二也会改善。因此我们首先考虑如何解决从 A 到 B 航迹最短规划。

根据拓扑学知识我们知道，最短路径问题是一个 NP 问题。对于这一类问题，我们首先需要建立**拓扑图结构模型**，通过**路径优化策略 Highway Hierarchies** 对图模型进行稀疏化，然后考虑结合最短路径搜索算法 **A* 算法** 和**进化算法** 的思想来对该问题进行求解。

我们注意到问题一与通常最短路径问题相比，增加了对定位误差的约束条件。在问题背景中，提出了三条约束。

(1) 在垂直校正点处，只有满足垂直误差不大于 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。

(2) 在水平校正点处，只有满足垂直误差不大于 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。

(3) 航行过程中，两类误差都不能超过 θ 。

我们注意到这些约束可以用来简化拓扑图的结构模型，将所有不可能满足约束的边从拓扑模型中剔除。

4.2 问题一模型的建立

如分析所述，我们结合问题一的假设建立了拓扑图模型 $G = (V, E)$ ，其中 V 是由所有数据点构成的集合，而 E 是这些点间的有向边集。

我们需要找到一条路径 $route$ ，我们假设这条路径包含 n 个校正点，那么自然地这条航线中共经过 $n + 2$ 个节点，包含 $n + 1$ 条边。我们规定该路径经过的第 i 个点的编号记为 r_i ，其中 r_1 固定为出发点 A 的编号， r_{n+2} 为终点 B 的编号。我们将编号为 j 的校正点类型记为 $type(j)$ 。为与题目所给数据表格一致，我们规定垂直校正点 $type = 1$ ，水平校正点 $type = 0$ 。为区别终点 B 与校正点，规定 $type(B) = -1$ 。第 i 段航迹长度记为 $dist(i)$ ，在问题一中航迹距离的计算公式为

$$dist(i) = \|e(r_i, r_{i+1})\| \quad (1)$$

在每一点处的校正前定位误差计算公式为：

$$errorV(r_{i+1}) = \begin{cases} dist(i) * \delta & type(r_i) = 1, \\ dist(i) * \delta + errorV(r_i) & type(r_i) = 0. \end{cases} \quad (2)$$

$$errorH(r_{i+1}) = \begin{cases} dist(i) * \delta + errorH(r_i) & type(r_i) = 1, \\ dist(i) * \delta & type(r_i) = 0. \end{cases} \quad (3)$$

那么我们需要优化的目标函数便可以形式化的表示为：

$$obj : \sigma_1 \sum_{i=1}^{n+1} dist(i) + \sigma_2(n + 2) \quad (4)$$

需满足的约束条件为：

$$errorV(r_i) \leq \begin{cases} \alpha_1 & type(r_i) = 1, \\ \beta_1 & type(r_i) = 0, \\ \theta & type(r_i) = -1. \end{cases} \quad (5)$$

$$errorH(r_i) \leq \begin{cases} \alpha_2 & type(r_i) = 1, \\ \beta_2 & type(r_i) = 0, \\ \theta & type(r_i) = -1. \end{cases} \quad (6)$$

其中 $errorV(r_i)$ 和 $errorH(r_i)$ 是编号为 r_i 在点校正前的定位误差。

4.3 问题一模型的求解

4.3.1 问题一模型的简化

根据路径优化策略 Highway Hierarchies[4] 算法的思想，我们对路网进行稀疏化，目的是将两类边从图 G 中去除。第一种边，两点间距离超过对应类型可行距离上确界的边，这种边可以视为在图 G 中不连通。第二种是对于最优化目标没有做出贡献的边。

根据校正点的不同类型，我们对约束条件进行讨论，在问题一中，因为不需要考虑飞行器转弯角度等限制，相邻节点 (i, j) 的航迹长度即为 $\|e(i, j)\|$ 。

(1) 当 $type(j) = 1$ 时，即下一个抵达点为垂直校正点时，需满足约束 (1)。这段抵达点为垂

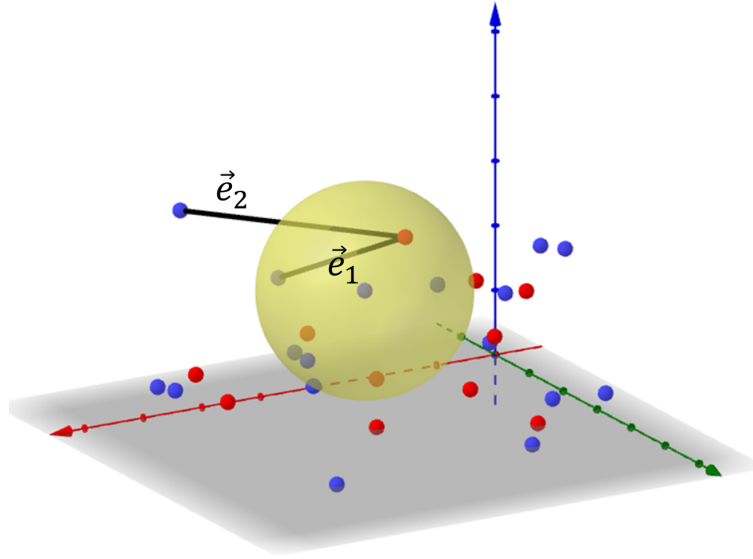


图 2 航迹半径示意图

直校正点的子航程距离的上确界 $sup(dist_{nextV})$ 计算公式为:

$$sup(dist_{nextV}) = \min(\alpha_1, \alpha_2) / \delta \quad (7)$$

这便意味着抵达垂直校正点的所有边中，超过该距离上确界的边都是无法通行的。如图 2 所示，可以抵达垂直校正点 j 的点被包含在以点 j 为球心， $sup(dist_{nextV})$ 为半径所构成的球中。

(2) 当 $type(j) = 0$ 时，即下一个抵达点为水平校正点时，需满足约束 (2)。这段抵达点为水

平校正点的子航程距离的上确界 $\sup(dist_{nextH})$ 计算公式为:

$$\sup(dist_{nextH}) = \min(\beta_1, \beta_2)/\delta \quad (8)$$

同样的, 在抵达水平校正点的所有边中, 超过该距离上确界的边都是无法通行的。可以抵达垂直校正点 j 的点 i 被包含在以点 j 为球心, $\sup(dist_{nextH})$ 为半径所构成的球中。

(3) 当 $type(j) = -1$ 时, 即下一个抵达点为目标点 B 时, 需满足约束 (3)

$$\sup(dist_{nextB}) = \theta/\delta \quad (9)$$

我们计算最后一段抵达点 B 的子航程距离的上确界 $dist_{nextB} = \theta/\delta$, 这便意味着这段抵达 B 的子航程中, 所有出发点被包含在以点 B 为球心, $dist_{nextB}$ 为半径所构成的球中。

根据上述对不同抵达点类型的航程长度上确界分析, 我们可以将不满足不等式 (10) 的边 $e(i, j)$ 从 E 中去除。缩减后的图 G 仍然包含所有可能飞行的航迹路径, 这是一个无损的模型简化步骤。

$$e(i, j) \leq \sup(dist_{type}) \quad (10)$$

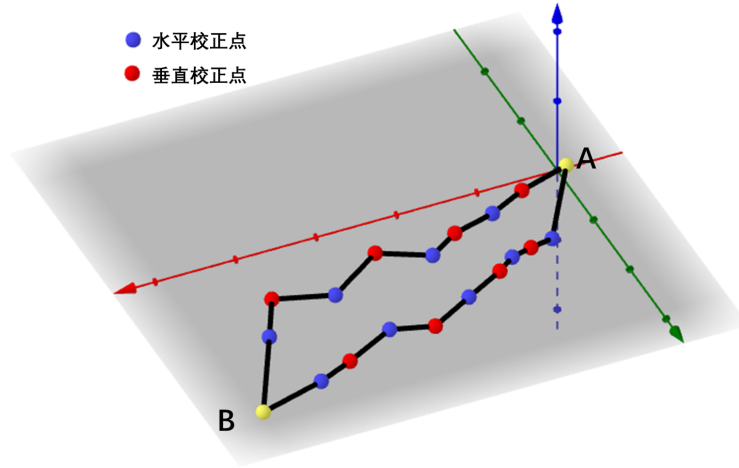


图 3 假设路径中同类型的校正点不会相邻

如图4所示, 观察到校正点在三维空间内分布密集, 根据路网稀疏化策略, 我们追加了假设

假设 1 为了实现经过的校正点尽可能少, 在飞行航线规划中, 同类型的校正点不会相邻。

假设 2 在飞行航线规划中, 每一段航程向量 $vec(i, j)$ 与向量 $vec(j, B)$ 的夹角余弦值要大于 0.5。

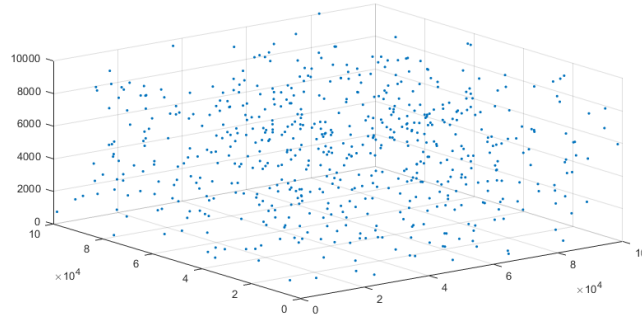


图 4 数据散点图

根据假设增加对边的约束

$$e(i, j) \notin E, \text{if}(\text{type}(i) == \text{type}(j)) \quad (11)$$

$$\cos(e(i, j), \text{vec}(j, B)) > 0 \quad (12)$$

以数据集 sheet1 为例说明简化模型所带来的效果，这样我们将图 G 中边集 E 的元素数量从 613*613 降到了 8272，降为原来边规模的 **2.2%**。从图 G 中所去除的这些边都是对我们所要优化的目标函数式 (24) 没有贡献或贡献很小的边，所以不用担心因为简化模型而带来的结果质量下降。

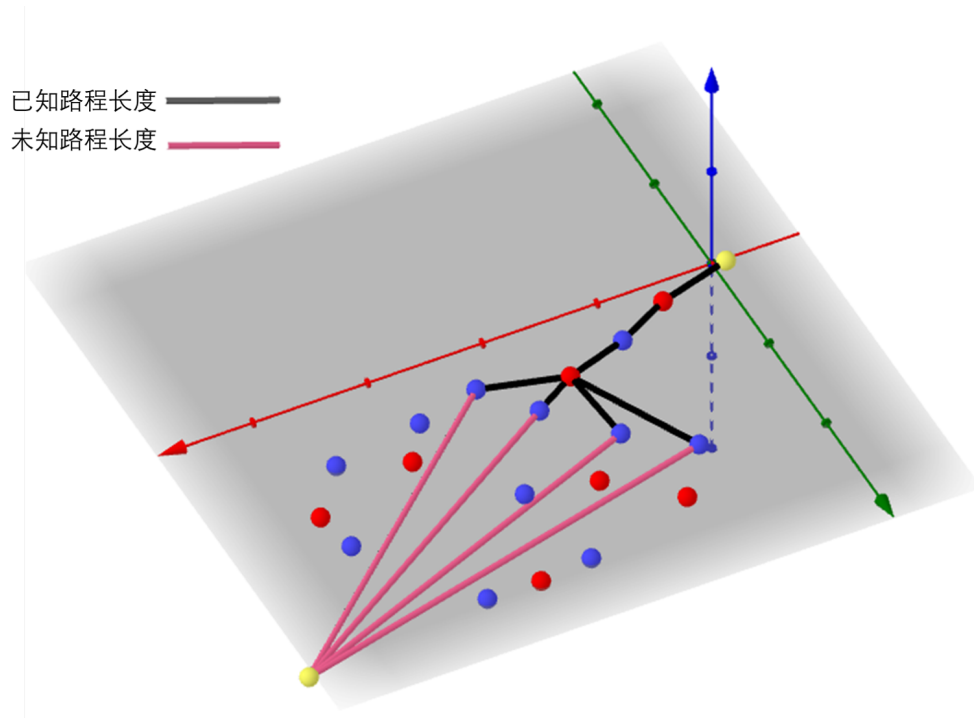


图 5 A* 算法示意图

4.3.2 问题一模型的求解

因为本问题为多目标最优化问题，所以我们希望能够针对不同权重系数 σ_1, σ_2 输出一个 **Pareto 解集**，我们通过粒子群优化模型实现这一目标。我们结合 A* 算法和进化算法这两种策略设计了问题一的模型求解算法。A* 算法是在 Dijkstra 算法的基础上对其进行改进，使得在每一步搜索中优先选择能够指向目标点方向的边，如图??所示。在本问题中，并非所有连通的边都能够满足约束，每一步搜索都需要对约束条件进行判断。在没有穷举所有状态之前，图的连通情况是不确定的，每走一步都会使图的拓扑情况发生改变。这种不确定性，使得每次搜索到终点 B 的解可能仅仅是一个局部最优解，而不是全局最优解。为了避免陷入局部最优解，在每次搜寻中按照一定概率对不同可行路径作出选择。同时我们还应用了进化算法中的思想，每次搜寻作为一个个体根据目前自身的信息和全部个体的最优信息作出判断，预先将每个个体在无约束条件下到达点 B 的最短路径记录。具体

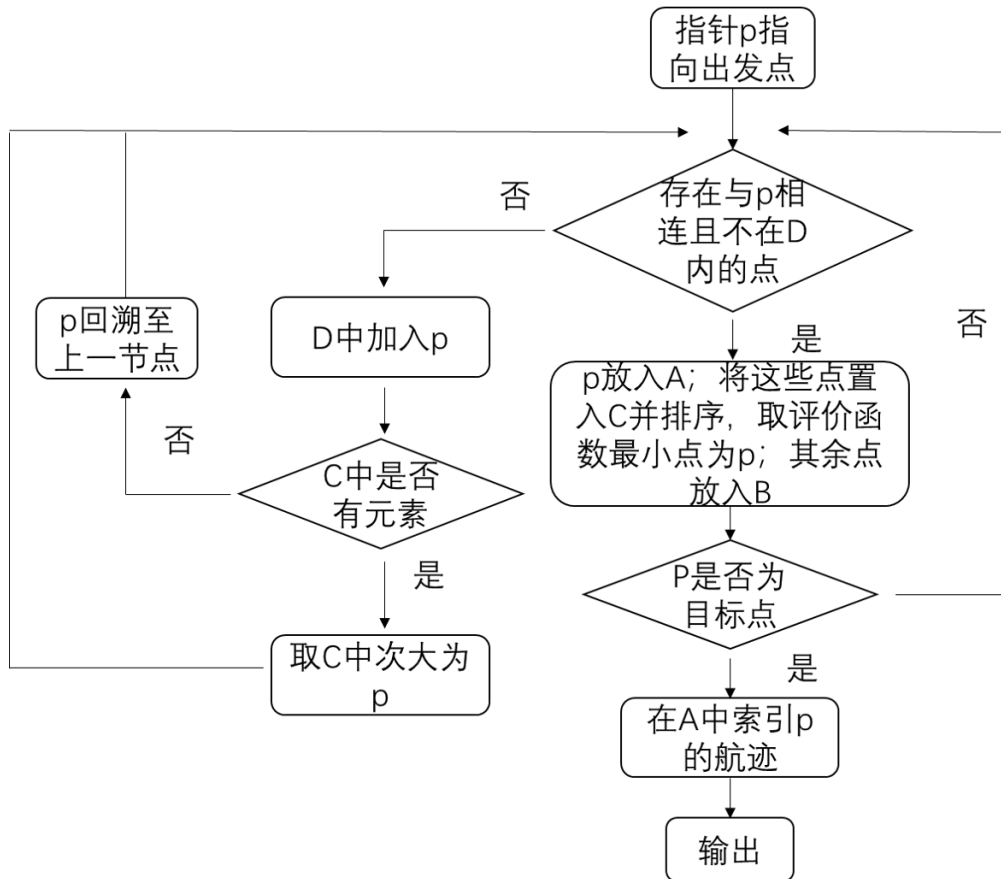


图 6 算法流程图

的算法是，我们设置一个状态指针 p ，每一状态 p 包括 [当前点编号，上一个编号，前一段航迹长度，校正前 $errorV$, 校正前 $errorH$, 评估函数, 总航迹长度]，初始化 $p=[1,1,0,0,0,0,0]$ 。我们分别设置了 4 个状态栈 A,B,C,D。其中

- 状态栈 A 存储已搜索的状态集

- 状态栈 B 存储待搜索状态集
- 状态栈 C 存储下一步可行状态集
- 状态栈 D 存储废弃状态集

当 p 被更新至终结状态时，便可以根据栈 A 存储的状态可以还原整条路径。

更新所用的评估函数为式 (13)，其中 $distSum$ 为当前已走航迹总长度， $shortestDistToB$ 为目标点 v_j 点 B 在无约束的情况下的最短路径，预先将其存储以减少运算复杂度。

$$Fun = distSum + shortestDistToB(j) \quad (13)$$

我们给出单次搜索的伪代码算法：

表 1 问题一算法

| 步骤 | 说明 |
|----|---|
| 1 | 初始化当前状态 p ，状态栈 A,B,C,D |
| 2 | while(p 没有达到终止条件) |
| 3 | do { |
| 4 | 搜寻 p 的可行状态域，与 D 中废弃状态做差集运算，将结果放入栈 C |
| 5 | 若 C 非空， p 入 A，选择 C 中可行状态更新 p ，把没有被选择的状态 C- p 入栈 B |
| 6 | 若 C 为空，状态 p 入废弃栈 D，选择 B 中可行状态更新 p |
| 7 | 若 B 也为空，则无可行路径，退出 |
| 8 | } |
| 9 | 根据栈 A 还原路径 route |

4.4 问题一结果

4.4.1 sheet1 问题一结果

我们通过该算法可以得到表2为问题一 sheet1 数据上 **9 校正点**的最短路径方案，航迹长度为 **104065.88 米**。表3为问题一 sheet1 数据上 **8 校正点**的最短路径方案，航迹长度为 **104861.06 米**。这两解分别对应着目标函数中对不同目标的重视程度 σ_1 和 σ_2 不一样的情况，两解构成问题一 sheet1 数据上多目标规划的一个 **Pareto 解集**。

具体结果见附录表2，表3。

航迹图见图7, 图8。

4.4.2 sheet2 问题一结果

校正点数为 12, 航线距离为 109342 米。具体结果见附录表4, 航迹图见图9。

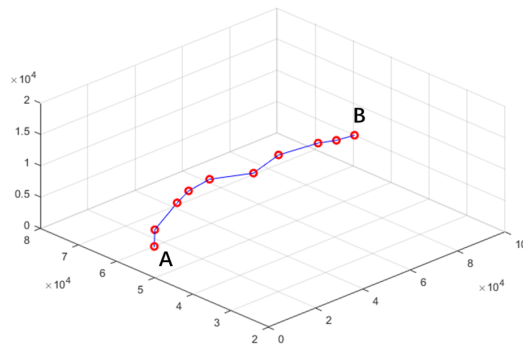


图 7 sheet1 问题一 8 校正点航迹规划路径

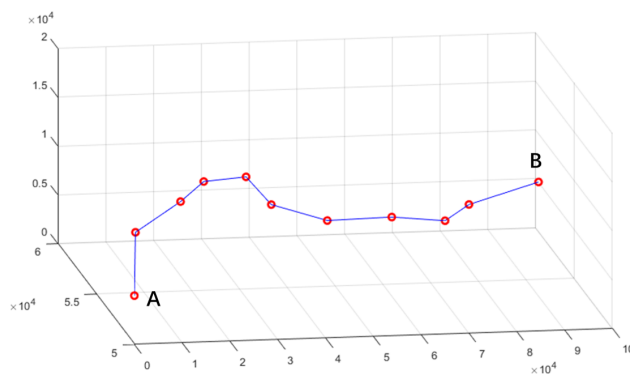


图 8 sheet1 问题一 9 校正点航迹规划路径

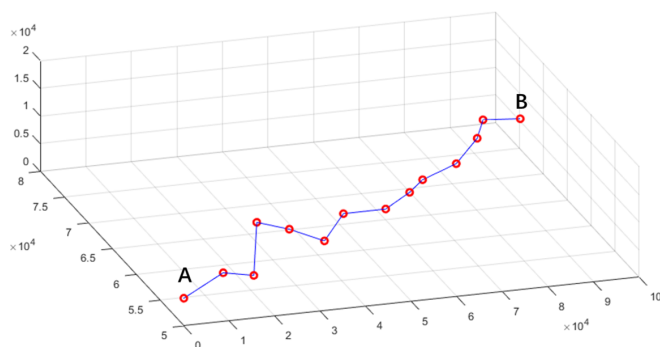


图 9 sheet2 问题一 12 校正点航迹规划路径

5.2 问题二模型的建立和求解

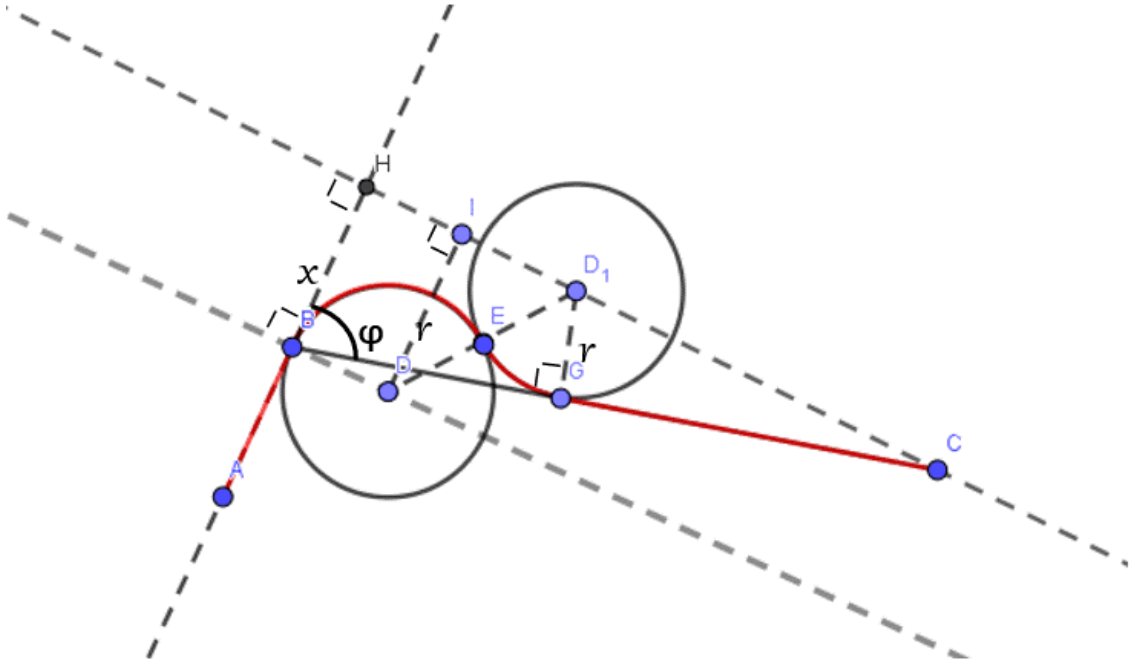


图 11 问题二转弯模型示意图

图11 对上述模型添加所需辅助线及对应标点，fig:simple 假设 $BH = x$ ， $D_1H = y$ ， $\angle HBC = \phi$ ，圆 D 和 D_1 的半径均为 $r = 200$ 。求解目标

$$obj : \min(\widehat{BE} + \widehat{EG} - BG) \quad (14)$$

求解过程：由 $\triangle BHC \sim \triangle D_1GC$ 得：

$$x^2 + (y - r)^2 = (2r)^2 \quad (15)$$

由勾股定理可得：

$$y = x \tan \phi - \frac{r}{\cos \phi} \quad (16)$$

联立式 (15)(16) 可得下列一元二次型方程：

$$\left(\frac{x}{r}\right)^2 (1 + \tan^2 \phi) - 2(\tan \phi) * \left(\frac{x}{r}\right) * \left(\frac{1}{\cos \phi} + 1\right) + \left(\frac{1}{\cos \phi} + 1\right)^2 - 4 = 0 \quad (17)$$

其解为

$$\frac{x}{r} = \sqrt{\cos^2 \phi * \left(2 - 2 * \cos \phi + \sin^2 \phi\right) + \sin \phi(1 + \cos \phi)} \quad (18)$$

根据初等几何知识：

$$\widehat{BE} = \left(\frac{\pi}{2} + \arccos \frac{2r}{x}\right) * r \quad (19)$$

$$\widehat{EG} = \left(\pi - \varphi - \arcsin \frac{2\pi}{x} \right) * r \quad (20)$$

$$BG = \frac{x}{\cos \varphi} - r \tan \varphi \quad (21)$$

将上述所得解式 (18) 代入式 (19)(20)(21) 可得到

$$dist_{add} = \widehat{BE} + \widehat{EG} - BG \quad (22)$$

在问题二中航迹距离的计算公式为

$$dist(i) = \|e(r_i, r_{i+1})\| + dist_{add}(i) \quad (23)$$

问题二的目标函数:

$$obj : \sigma_1 \sum_{i=1}^{n+1} dist(i) + \sigma_2(n+2) \quad (24)$$

需满足的约束条件为:

$$errorV(r_i) \leq \begin{cases} \alpha_1 & type(r_i) = 1, \\ \beta_1 & type(r_i) = 0, \\ \theta & type(r_i) = -1. \end{cases} \quad (25)$$

$$errorH(r_i) \leq \begin{cases} \alpha_2 & type(r_i) = 1, \\ \beta_2 & type(r_i) = 0, \\ \theta & type(r_i) = -1. \end{cases} \quad (26)$$

其中 $errorV(r_i)$ 和 $errorH(r_i)$ 是编号为 r_i 在点校正前的定位误差。

其中 $dist_{add}$ 即为该问题中因新增假设而增加的飞行器航迹路程。这样我们便可以更新航迹长度，将问题一中的可行解集中的路径用新的航迹长度代入检验，进而寻找到问题二中的 Pareto 解集。

5.3 问题二结果

5.3.1 sheet1 问题二结果

给出的解集中包含两解。

- 1 校正点数为 9，航迹长度为 104917 米，具体结果见附录表5。航迹图见图12
- 2 校正点数为 8，航迹长度为 105075 米, 附录表6中。航迹图见图13

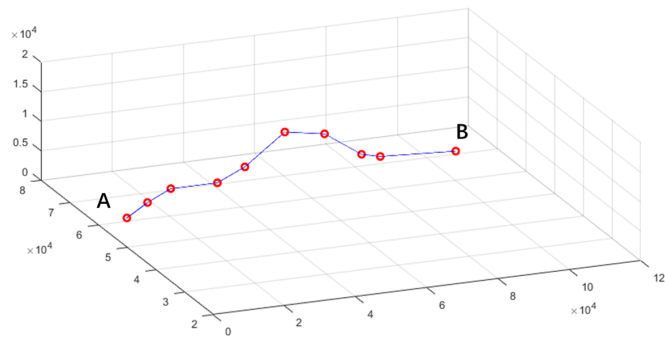


图 12 sheet1 问题二 8 校正点航迹规划路径图

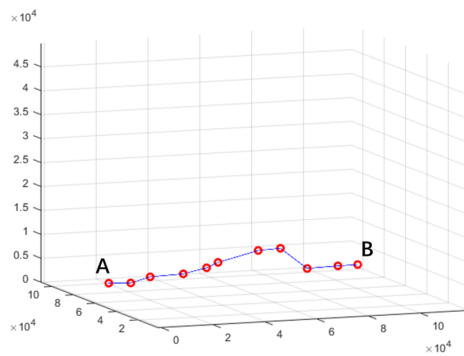


图 13 sheet1 问题二 9 校正点航迹规划路径图

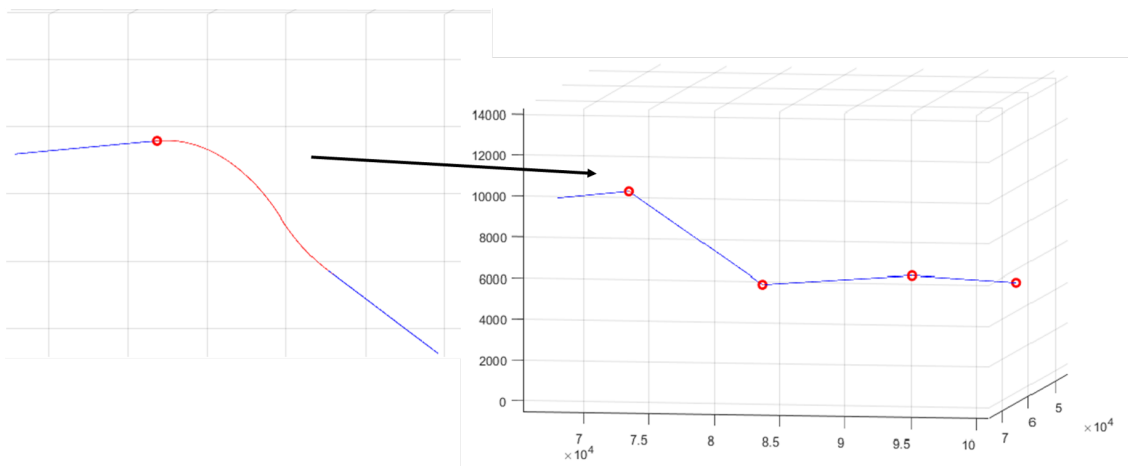


图 14 sheet1 问题二 9 校正点航迹规划局部放大图

5.3.2 sheet2 问题二结果

textbf 校正点数为 12，航迹长度为 109547 米，具体结果在附录表7。

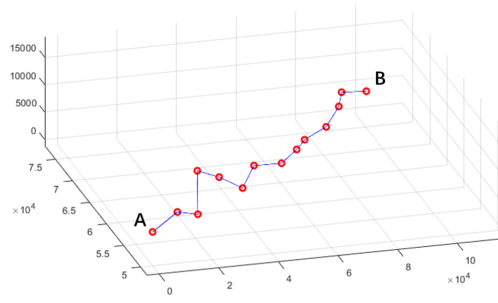


图 15 sheet2 问题二 12 校正点航迹规划路径图

6. 问题三

6.1 问题三分析

问题 3 在问题一的基础上增加假设：飞行器在问题校正点能够成功将某个误差精确校正为 0 的概率是 80%；如果校正失败，则校正后的剩余误差为 $\min(error, 5)$ 个单位。该问题基于问题一重新规划其要求的航迹，要求成功到达终点的概率 $P_{success}$ 尽可能大。

对于问题三，根据题设，对部分校正点增加校正成功的概率作为该点的一个新属性，目标是希望尽可能增加航迹的成功率；为了简化问题，我们在问题一求解给出的解集的基础上考虑航迹到达终点的成功率 $P_{success}$ ，从而得出最优解。

6.2 问题三模型的建立和求解

为了建立问题三模型，我们分四种情况进行讨论：

- (1) 若航迹中的问题校正点个数为 0，则 $P_{success} = 1$ 。
- (2) 若航迹中的问题校正点个数不为 0，但问题校正点均成功校正，则这种情况下

$$P_{success}(1) = power(0.8, n) \quad (27)$$

其中 $power(x, y)$ 表示数 x 的 y 次方， n 表示某一条从起点 A 到终点 B 的航迹中存在的问题校正点的个数。

- (3) 若航迹中仅有一个问题校正点校正失败，根据排列组合的知识可推得这一个校正失败问题点的位置会有 n 种可能，我们记录在这 n 种可能的情况中成功的次数为 m ，那么这种情况下

$$P_{success}(2) = C_{numOfProblem}^1 * power(0.8, numOfProblem - 1) * 0.2 * m/n \quad (28)$$

- (4) 若航迹中有两个或两个以上校正点校正失败，为了简化问题模型，我们给出假设：

假设 4 若航迹中有两个或两个以上校正点校正失败，此时由经验知识可知这种情况下的航迹飞行成功率十分低，视为失败。

$$P_{success}(3) = 0 \quad (29)$$

最后通过将上述情况的成功概率相加得到某条特定航线的总飞行成功率 $P_{success}$ 。

$$P_{success} = P_{success}(1) + P_{success}(2) + P_{success}(3) \quad (30)$$

问题三的目标函数为：

$$obj : \sigma_1 \sum_{i=1}^{n+1} dist(i) + \sigma_2(n+2) + \sigma_3 P_{success} \quad (31)$$

需满足的约束条件为：

$$errorV(r_i) \leq \begin{cases} \alpha_1 & type(r_i) = 1, \\ \beta_1 & type(r_i) = 0, \\ \theta & type(r_i) = -1. \end{cases} \quad (32)$$

$$errorH(r_i) \leq \begin{cases} \alpha_2 & type(r_i) = 1, \\ \beta_2 & type(r_i) = 0, \\ \theta & type(r_i) = -1. \end{cases} \quad (33)$$

其中 $errorV(r_i)$ 和 $errorH(r_i)$ 是编号为 r_i 在点校正前的定位误差。

这样我们便可以利用问题一中的模型和算法对该问题进行求解，得到问题三的 Pareto 解集。

6.3 问题三结果

6.3.1 sheet1 问题三结果

对于 sheet1 所提供的的数据我们给出一条路径，该路线校正点为 8，航迹长度为 106640，成功率 0.8192。该路径有 5 种成功情况，分别对应附录表8, 表9, 表10, 表11, 表12。

6.3.2 sheet2 问题三结果

对于 sheet2 所提供的的数据我们给出一条路径，该路线校正点数为 13，航迹长度为 114066，成功率为 0.1342。该路径有 2 种成功情况，分别对应附录表13, 表14。

7. 模型的分析与评价

7.1 复杂度分析

对于我们所采用的类 A^* ，考虑到算法性能，外循环中每次从栈 C 取一个元素，共取了 n 次（共 n 个结点），每次展开一个结点的后续结点时，需 $O(n)$ 次，同时再对栈 C 做一次排序，栈 C 大小是 $O(n)$ 量级的，若用快排就是 $O(n\log n)$ ，乘以外循环总的复杂度是 $O(n^2\log n)$ ，如果每次不是对栈 C 进行排序，因为总是不断地有新的结点添加进来，所以不用进行排序，而是每次从栈 C 中求一个最小的，那只需要 $O(n)$ 的复杂度，所以总的复杂度为 $O(n * n)$ ，

7.2 模型的评价

本题的求解通过建立有约束的拓扑图模型，利用 A^* 算法的思想构建启发函数，并按照启发函数值低的方向进行搜索，寻找满足约束条件的飞行器航迹。

针对问题一，在满足约束条件 (1)-(7) 的情况下，从最优化的角度出发，我们通过预先处理数据集中的点，剔除不可通过的边。因为 A 星算法的下限是当决定向哪个方向或者在哪个区域内进行搜索，直到所有可以到达的点均通过 open list 或者 close list 得到了处理，显然这样会浪费大量的 CPU 时间，因此预处理步骤不但可以提前在拓扑图中剔除搜索算法不可到达的边，而且加快了算法运行的速度。

针对问题二，增加条件 (8) 的约束，我们提出一种优化求解模型，使得在飞行器航向发生改变时经过最短的圆弧飞行后就能重新回到直线轨迹上，这不仅利于后续前进方向的确定，还使得飞行器在转弯时额外增加的航迹里程尽可能的小，这一模型的建立也使得 A 星搜索算法在进行搜索时的速度与效率得到了提升。

针对问题三，对误差校正点增加了一个属性，即可能存在问题误差点使得误差校正不能精确到 0，并且相应的对应一个概率，目标希望得到的航迹成功飞到终点的概率尽可能高，我们通过对四种不同情况的讨论，最终建立加权求和的目标函数来得到满足题设的最优飞行器航迹路线。

针对该类有约束的条件下的多目标路径规划问题， A^* 算法是一种静态路网中求解最短路径最有效的方法，而算法中对估价函数的选取显得尤为重要，估价函数值的设定与实际值越接近越好，也能保证搜索方向始终朝着终点的方向进行；约束条件的不同，构建的模型不同，采用基于 A^* 算法的搜索策略来进行该类问题的求解会具有较好的效果。

参考文献

- [1] 郑昌文, 严平, 丁明跃, 等, 飞行器航迹规划研究现状与趋势, 宇航学报, 28(6):1441-1446, 2007.

- [2] 严平, 丁明跃, 周成平, 等, 飞行器多任务在线实时航迹规划, 宇航学报, 25(5):485-489, 2004.
- [3] 李春华, 周成平, 丁明跃, 等, 动态环境中的飞行器实时三维航迹规划方法研究, 宇航学报, 24(1):38-42, 2003.
- [4] Sanders P, Schultes D, Highway hierarchies hasten exact shortest path queries, European Symposium on Algorithms, Springer, 568-579, 2005.

附录 A MATLAB 源程序

1.1 图模型稀疏算法

```

Dist = zeros(length(V(:,1)));           % 计算v的邻接矩阵
for i = 1 : length(V(:,1))-1           % i为出发点的编号
for j = 2 : length(V(:,1))             % j为抵达点的编号
Dist(i,j) = vecnorm(V(i,1:3)-V(j,1:3));
end
end
distA_B = Dist(1,length(V(:,1)));      %
    当所有校正点都在A至B连线上理想位置时的最短航程, 点A到点B的直线距离
minTimes = (distA_B-lastRadius)/distMax * 2 + 1 ; %
    该公式仅对数据一有效!! 当所有校正点都在A至B连线上理想位置时的最少校正次数。

Edge = zeros(length(V(:,1)));          % Edge为有向边
for i = 1 : length(V(:,1))-1           % i为出发点的编号
for j = 2 : length(V(:,1))             % j为抵达点的编号
if V(j,4)==1 && Dist(i,j)>= radiusNextV %
    若下一个目标点为垂直校正点,距离超过可达航程,将该边E(i,j)设为不可达;
Edge(i,j) = 0;
elseif V(j,4)==0 && Dist(i,j)>= radiusNextH %
    若下一个目标点为水平校正点,距离超过可达航程,将该边E(i,j)设为不可达;
Edge(i,j) = 0;
elseif V(j,4)==-1 && Dist(i,j)>= lastRadius %
    若下一个目标点为B,距离超过可达航程,将该边E(i,j)设为不可达;
Edge(i,j) = 0;
else
Edge(i,j) = Dist(i,j);
end
end

```

```

end
%
    计算vector(i,j)与vector(i,B)夹角的余弦值, Degree(i,j)值为vector(i,j)与vector(i,B)夹角
Cos = zeros(length(V(:,1))); %
    Cos(i,j)值为vector(i,j)与vector(i,B)夹角的余弦值
Degree = zeros(length(V(:,1))); %
    Degree(i,j)值为vector(i,j)与vector(i,B)夹角
for i = 1 : length(V(:,1))-1 % i为出发点的编号
for j = 2 : length(V(:,1)) % j为抵达点的编号
if Edge(i,j) ~= 0
v1 = V(j,1:3)-V(i,1:3);
v2 = V(length(V(:,1)),1:3)-V(i,1:3);
if j == length(V(:,1))
Cos(i,j)=1;
else
Cos(i,j) = dot(v1,v2)/norm(v1,2)/norm(v2,2);
end
Degree(i,j) = real(rad2deg(acos(Cos(i,j))));
end
end
end

for i = 1 : length(V(:,1))-1 % i为出发点的编号
for j = 2 : length(V(:,1)) % j为抵达点的编号
if Edge(i,j) ~= 0
if V(i,4)==V(j,4) % 增加约束出发点和目标点不能为同一类型
Edge(i,j) = 0;
end
end
end
end

for i = 1 : length(V(:,1))-1 % i为出发点的编号
for j = 2 : length(V(:,1)) % j为抵达点的编号
if Edge(i,j) ~= 0
if Cos(i,j) <= 0.5 % 要求向量夹角余弦值大于0.5
Edge(i,j) = 0;
end
end
end

```

```

end
end

Number = find(Edge);    %构建拓扑有向图G
count = length(Number);
w = zeros(count,1);
s = mod(Number,length(V(:,1)));
t = (Number-s)/length(V(:,1))+1;
for i = 1:count
w(i) = Edge(Number(i));
end
G = graph(s,t,w);

shortestDistToB = zeros(length(V(:,1)),1);
for i = 1:length(V(:,1))
[~,shortestDistToB(i)] = shortestpath(G,i,length(V(:,1)));
end

```

1.2 问题一 A* 进化算法

```

stackA = [];
stackB = [];
stackC = [];
stackD = [];
p = [1,1,0,0,0,0,0];
count = 1;
iterNum = 10000;
route = cell(iterNum,3);

while count<=iterNum
fesibleB = intersect(find( Edge( p(1),length(V(:,1)) ) ),find( Edge(
    p(1),length(V(:,1)) )<= theta/delta -p(3) ));
if ~isempty(fesibleB)
stackA = [stackA;p];
p = [length(V(:,1)),p(1),Edge( p(1),length(V(:,1))),p(4)+Edge(
    p(1),length(V(:,1)))*delta,p(5)+
Edge(
    p(1),length(V(:,1)))*delta,func(p(1),length(V(:,1)),V,shortestDistToB),p(7)+Edge(
    p(1),length(V(:,1)))]]; %最后状态

```



```

stackA = [stackA;p]; %#ok<*AGROW>
route{count,1}=[length(V(:,1)),p(2)];
i=1;
while p(2)~=1
route{count,1}(i)=p(2);
temp = find(stackA(:,1)==p(2));
p(2)=stackA(temp(end),2);
i=i+1;
end
route{count,1} = fliplr( [length(V(:,1)),route{count,1},1]); %
    将路径正向显示
[route{count,2},route{count,3}] =
    judgement(route{count,1},Edge,distMax,theta/delta);

p = [1,1,0,0,0,0,0];
stackA = [];
stackB = [];
stackC = [];
stackD = [];
count = count+1;
continue;
else % 如果没有终止, 搜寻p的可行点域
fesibleP =intersect( find(Edge( p(1),: )) , find(Edge( p(1),:
    )<=distMax-p(3)));
if ~isempty(fesibleP) % 若可行点域不为空

stackC = [];
if V(p(1),4)==1 % 判断当前点类型
for i = 1:length(fesibleP)
stackC =[stackC;
    [fesibleP(i),p(1),Edge(p(1),fesibleP(i)),Edge(p(1),fesibleP(i))*delta,Edge(p(1),
fesibleP(i))*delta+p(5),func(p(1),fesibleP(i),V,shortestDistToB),p(7)+Edge(p(1),fesibleP(i))
end
else
for i = 1:length(fesibleP)
stackC =[stackC;
    [fesibleP(i),p(1),Edge(p(1),fesibleP(i)),Edge(p(1),fesibleP(i))*delta+p(4),Edge(p(1),
fesibleP(i))*delta,func(p(1),fesibleP(i),V,shortestDistToB),p(7)+Edge(p(1),fesibleP(i))
end
end

```

```

end
if ~isempty(stackC) && ~isempty(stackD)
stackC = setdiff(stackC, stackD, 'row');
end
if ~isempty(stackC) % 从栈C中更新当前状态
stackA = [stackA;p];
stackC = sortrows(stackC,6);
randTemp = randN(stackC);
p = stackC(randTemp,:);
stackC(randTemp,:) = [];
stackB = [stackB;stackC]; % 将没用过的状态栈C并入状态栈B
else
if ~isempty(stackB)
stackB = sortrows(stackB,6); % 栈若栈C与栈D差集为空，从栈B中取出更新当前状态
randTemp = randN(stackB);
p = stackB(randTemp,:);
stackB(randTemp,:)=[];
else
false
break
end
end
else

stackD = [stackD;p];
if ~isempty(stackC)
p = stackC(randN(stackC),:);
stackC(1,:) = [];
for i = length(stackB(:,1)):-1:1
if stackB(i,:)== p
stackB(i,:)=[];
break;
end
end
else % 如果stackC也为空，从栈B中取出状态
if ~isempty(stackB)
stackB = sortrows(stackB,6);
randTemp = randN(stackB);
p = stackB(randTemp,:);

```

```

stackB(randTemp,:)=[];
else
false           % 标记算法出错或问题无解
break
end
end
end
end
end
end

```

1.3 问题二增加长度计算

```

function [bonusLenth] = bonusLength(phi)
r=200.0;
a=1+tan(phi)^2;
b=-2*(1+1/cos(phi))*tan(phi);
c=(1+1/cos(phi))^2-4;
x=r*(-b+sqrt(b^2-4*a*c))/(2*a);
if phi<=2*atan(1/2)
bonusLenth=(-phi+2*asin(x/(2*r)))*r - x/cos(phi)+r*tan(phi);
else
bonusLenth=(2*pi-phi-2*asin(x/(2*r)))*r - x/cos(phi)+r*tan(phi);
end
end

```

1.4 问题三成功率判断

```

numOfProblem = 0;
problem = [];
for i = 2:length(route)
if V(route(i),5)==1
numOfProblem = numOfProblem + 1 ;
problem(numOfProblem) = i; %#ok<AGROW>
end
end
if numOfProblem == 0
successPossibilitySum = 1;
else %如果问题点数目不为0,进行讨论
possibility = zeros(3,1); % 没有遇到校正失败则成功

```

```

possibility(1) = power(0.8,numOfProblem); % 仅考虑任务中遇到一次校正失败的情况
possibility(2) = numOfProblem * power(0.8,numOfProblem - 1)*0.2;
success = 0;
for k = 1:numOfProblem
errorV = zeros(length(route),1);
errorH = zeros(length(route),1);
errorV(1) = 0;
errorH(1) = 0;
tag = 1; %成功则输出1, 失败为-1
for i = 2:length(route)
errorV(i) = errorV(i-1)+Edge(route(i-1),route(i))*delta; %
    先更新到达路径点i校正前的误差
errorH(i) = errorH(i-1)+Edge(route(i-1),route(i))*delta;
if V(route(i),4)== 1 %
    判断是否满足各类型点要求, 若不满足, 判定任务大概率失败
if errorV(i)>alpha1 || errorH(i)>alpha2
tag = -1;
end
elseif V(route(i),4)== 0
if errorV(i)>beta1 || errorH(i)>beta2
tag = -1;
end
elseif V(route(i),4)== -1
if errorV(i)>theta || errorH(i)>theta
tag = -1;
end
end
if V(route(i),5)==0 || problem(k) ~= i % 该校正点正常校正
if V(route(i),4)== 1
errorV(i)=0;
elseif V(route(i),4)== 0
errorH(i)=0;
end
elseif V(route(i),5)==1 && problem(k) == i %
    该校正点为问题校正点且出现问题, 只能取得部分校正
if V(route(i),4)== 1
errorV(i)= min(errorV(i),5);
elseif V(route(i),4)== 0
errorH(i)= min(errorH(i),5);

```

```

end
end
end
if tag ==1
success = success+1;
end
end
% 假设每次任务遇到一次以上校正失败则失败
possibility(3) = 1 - possibility(1) - possibility(2);
successPossbilitySum = possibility(1) + possibility(2) *
    success/numOfProblem;
end
else
successPossbilitySum = 0; % tag ==0时不予判断，一定失败
end

```

附录 B 数据表

表 2 sheet1 问题一 9 校正点路径

| 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 | 出发点 A |
| 503.00 | 13.39 | 13.39 | 1 |
| 294.00 | 10.18 | 23.57 | 0 |
| 91.00 | 17.54 | 7.35 | 1 |
| 607.00 | 8.35 | 15.71 | 0 |
| 170.00 | 11.95 | 3.59 | 1 |
| 278.00 | 10.46 | 14.05 | 0 |
| 369.00 | 21.89 | 11.44 | 1 |
| 214.00 | 13.31 | 24.75 | 0 |
| 397.00 | 22.33 | 9.02 | 1 |
| 612.00 | 16.97 | 25.99 | 终点 B |

表 3 sheet1 问题一 8 校正点路径

| 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 | 出发点 A |
| 503 | 13.38791985 | 13.38791985 | 1 |
| 294 | 10.18081072 | 23.56873058 | 0 |
| 91 | 17.53551211 | 7.354701386 | 1 |
| 607 | 8.353018096 | 15.70771948 | 0 .. |
| 540 | 21.38590334 | 13.03288524 | 1 |
| 250 | 11.42932512 | 24.46221036 | 0 |
| 340 | 24.19844923 | 12.76912411 | 1 |
| 277 | 12.00237598 | 24.77150009 | 0 |
| 612.00 | 16.97 | 25.99 | 终点 B |

表 4 sheet2 问题一 12 校正点路径

| 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 | 出发点 A | |
| 163 | 13.28789761 | 13.28789761 | 0 | |
| 114 | 18.62205093 | 5.334153324 | 1 | |
| 8 | 13.92198578 | 19.2561391 | 0 | |
| 309 | 19.44631118 | 5.524325401 | 1 | |
| 305 | 5.968714547 | 11.49303995 | 0 | |
| 123 | 15.17310764 | 9.204393096 | 1 | .. |
| 45 | 10.00616142 | 19.21055451 | 0 | |
| 160 | 17.49129596 | 7.485134547 | 1 | |
| 92 | 5.776163625 | 13.26129817 | 0 | |
| 93 | 15.26088202 | 9.484718396 | 1 | |
| 61 | 9.834209702 | 19.3189281 | 0 | |
| 292 | 16.38812359 | 6.553913884 | 1 | |
| 326 | 6.960509275 | 13.51442316 | 终点 B | |

表 5 sheet1 问题二 9 校正点路径

| 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 | 出发点 A |
| 503 | 13.38972848 | 13.38972848 | 1 |
| 69 | 8.809532174 | 22.19926065 | 0 |
| 237 | 21.31172074 | 12.50218857 | 1 |
| 233 | 10.82404737 | 23.32623594 | 0 |
| 33 | 15.99639794 | 5.172350566 | 1 |
| 315 | 15.46084151 | 20.63319208 | 0 |
| 403 | 23.49413299 | 8.03329148 | 1 |
| 594 | 11.0349412 | 19.06823268 | 0 |
| 501 | 22.23521613 | 11.20027492 | 1 |
| 612 | 8.490014015 | 19.69028894 | 终点 B |

..

表 6 sheet1 问题二 8 校正点路径

| 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 | 出发点 A |
| 503 | 13.38972848 | 13.38972848 | 1 |
| 69 | 8.809532174 | 22.19926065 | 0 |
| 237 | 21.31172074 | 12.50218857 | 1 |
| 233 | 10.82486229 | 23.32705086 | 0 .. |
| 598 | 24.83437288 | 14.00951059 | 1 |
| 561 | 10.95799439 | 24.96750498 | 0 |
| 485 | 21.80384981 | 10.84585541 | 1 |
| 248 | 4.222203171 | 15.06805858 | 0 |
| 612 | 23.73565409 | 19.51345092 | 终点 B |

表 7 sheet2 问题二 12 校正点路径

| 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 | 出发点 A | |
| 163 | 13.30025396 | 13.30025396 | 0 | |
| 114 | 18.69286574 | 5.392611782 | 1 | |
| 8 | 13.98659117 | 19.37920295 | 0 | |
| 309 | 19.51101401 | 5.524422841 | 1 | |
| 305 | 5.99799833 | 11.52242117 | 0 | |
| 123 | 15.20591216 | 9.207913826 | 1 | .. |
| 45 | 10.0069466 | 19.21486043 | 0 | |
| 160 | 17.49262561 | 7.485679011 | 1 | |
| 92 | 5.776952377 | 13.26263139 | 0 | |
| 93 | 15.26249631 | 9.485543938 | 1 | |
| 61 | 9.834787321 | 19.32033126 | 0 | |
| 292 | 16.42168006 | 6.586892737 | 1 | |
| 326 | 6.960509275 | 13.54740201 | 终点 B | |

表 8 sheet1 问题三 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 | 出发点 A | |
| 578 | 12.01073848 | 12.01073848 | 01 | |
| 417 | 19.97535503 | 7.964616544 | 11 | |
| 80 | 12.00373006 | 19.9683466 | 01 | |
| 237 | 16.63147462 | 4.62774457 | 11 | |
| 233 | 10.8239642 | 15.45170877 | 01 | .. |
| 33 | 15.99499137 | 5.171027179 | 11 | |
| 194 | 10.50284979 | 15.67387697 | 01 | |
| 450 | 16.47999404 | 5.97714425 | 11 | |
| 286 | 12.77131463 | 18.74845888 | 01 | |
| 485 | 13.9875233 | 1.216208667 | 11 | |
| 612 | 23.57093923 | 24.7871479 | 终点 B | |

表 9 sheet1 问题三 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 | 出发点 A | |
| 578 | 12.01073848 | 12.01073848 | 02 | |
| 417 | 19.97535503 | 12.96461654 | 11 | |
| 80 | 12.00373006 | 24.9683466 | 01 | |
| 237 | 16.63147462 | 4.62774457 | 11 | |
| 233 | 10.8239642 | 15.45170877 | 01 | .. |
| 33 | 15.99499137 | 5.171027179 | 11 | |
| 194 | 10.50284979 | 15.67387697 | 01 | |
| 450 | 16.47999404 | 5.97714425 | 11 | |
| 286 | 12.77131463 | 18.74845888 | 01 | |
| 485 | 13.9875233 | 1.216208667 | 11 | |
| 612 | 23.57093923 | 24.7871479 | 终点 B | |

表 10 sheet1 问题三 3

| 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 | 出发点 A | |
| 578 | 12.01073848 | 12.01073848 | 01 | |
| 417 | 19.97535503 | 7.964616544 | 12 | |
| 80 | 17.00373006 | 19.9683466 | 01 | |
| 237 | 21.63147462 | 4.62774457 | 11 | |
| 233 | 10.8239642 | 15.45170877 | 01 | .. |
| 33 | 15.99499137 | 5.171027179 | 11 | |
| 194 | 10.50284979 | 15.67387697 | 01 | |
| 450 | 16.47999404 | 5.97714425 | 11 | |
| 286 | 12.77131463 | 18.74845888 | 01 | |
| 485 | 13.9875233 | 1.216208667 | 11 | |
| 612 | 23.57093923 | 24.7871479 | 终点 B | |

表 11 sheet1 问题三 4

| 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 | 出发点 A | |
| 578 | 12.01073848 | 12.01073848 | 01 | |
| 417 | 19.97535503 | 7.964616544 | 11 | |
| 80 | 12.00373006 | 19.9683466 | 02 | |
| 237 | 16.63147462 | 9.62774457 | 11 | |
| 233 | 10.8239642 | 20.45170877 | 01 | .. |
| 33 | 15.99499137 | 5.171027179 | 11 | |
| 194 | 10.50284979 | 15.67387697 | 01 | |
| 450 | 16.47999404 | 5.97714425 | 11 | |
| 286 | 12.77131463 | 18.74845888 | 01 | |
| 485 | 13.9875233 | 1.216208667 | 11 | |
| 612 | 23.57093923 | 24.7871479 | 终点 B | |

表 12 sheet1 问题三 5

| 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 | 出发点 A | |
| 578 | 12.01073848 | 12.01073848 | 01 | |
| 417 | 19.97535503 | 7.964616544 | 11 | |
| 80 | 12.00373006 | 19.9683466 01 | | |
| 237 | 16.63147462 | 4.62774457 | 12 | |
| 233 | 15.8239642 | 15.45170877 | 01 | .. |
| 33 | 20.99499137 | 5.171027179 | 11 | |
| 194 | 10.50284979 | 15.67387697 | 01 | |
| 450 | 16.47999404 | 5.97714425 | 11 | |
| 286 | 12.77131463 | 18.74845888 | 01 | |
| 485 | 13.9875233 | 1.216208667 | 11 | |
| 612 | 23.57093923 | 24.7871479 | 终点 B | |

表 13 sheet2 问题三 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 | 出发点 A |
| 163 | 13.28789761 | 13.28789761 | 01 |
| 114 | 18.62205093 | 5.334153324 | 11 |
| 8 | 13.92198578 | 19.2561391 | 01 |
| 309 | 19.44631118 | 5.524325401 | 11 |
| 305 | 5.968714547 | 11.49303995 | 01 |
| 123 | 15.17310764 | 9.204393096 | 11 |
| 45 | 10.00616142 | 19.21055451 | 01 |
| 160 | 17.49129596 | 7.485134547 | 11 |
| 92 | 5.776163625 | 13.26129817 | 01 |
| 93 | 15.26088202 | 9.484718396 | 11 |
| 38 | 6.296532293 | 15.78125069 | 01 |
| 110 | 10.2230688 | 3.926536508 | 11 |
| 99 | 9.199888456 | 13.12642496 | 01 |
| 326 | 17.84948327 | 8.649594817 | 终点 B |

表 14 sheet2 问题三 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 | 出发点 A |
| 163 | 13.28789761 | 13.28789761 | 01 |
| 114 | 18.62205093 | 5.334153324 | 11 |
| 8 | 13.92198578 | 19.2561391 | 01 |
| 309 | 19.44631118 | 5.524325401 | 11 |
| 305 | 5.968714547 | 11.49303995 | 01 |
| 123 | 15.17310764 | 9.204393096 | 11 |
| 45 | 10.00616142 | 19.21055451 | 01 |
| 160 | 17.49129596 | 7.485134547 | 11 |
| 92 | 5.776163625 | 13.26129817 | 01 |
| 93 | 15.26088202 | 9.484718396 | 12 |
| 38 | 11.29653229 | 15.78125069 | 01 |
| 110 | 15.2230688 | 3.926536508 | 11 |
| 99 | 9.199888456 | 13.12642496 | 01 |
| 326 | 17.84948327 | 8.649594817 | 终点 B |

..