



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

东南大学

参赛队号

19102860174

队员姓名

1.朱怡婷

2.梁升一

3.李 凯

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：

飞行器航迹快速规划是飞行器控制的重要技术手段，在飞行过程中由于复杂的环境和自身的系统结构限制，飞行器的定位误差会随着飞行距离不断增加，若误差过大则飞行任务失败，因此及时地校正定位误差是航迹规划的一项重要任务。本问题采用经过定位校正点的方式进行垂直位置校正以及水平位置校正，其最终目标是要从几百个点中选择合适的点，使其能在成功达到终点的前提下，尽可能的减少飞行的长度以及校正的次数。但由于选择的点数量很多，且随着选择校正点类型的不同，会导致不同的误差，这对如何规划出好的轨迹带来很大的挑战。针对题目中的问题，我们给出了如下思路。

问题一：需要在满足误差约束的情况下，选择合适的点来形成航迹。首先，我们基于误差的约束范围建立了**飞行误差限制模型**，为飞行器选择下一校正点的类型和选择范围提供了**参考依据**，保证了选择的点**一定满足约束条件**。在此基础上，为了实现航迹长度尽可能小，以及校正次数尽可能少，我们以局部飞行距离长度和飞行朝向为指标，构建了符合问题要求的**多变量贪婪算法**。结果表明，对于附件一的数据集，达到了 8 个校正点， $1.1509 \times 10^5 \text{ m}$ 的结果。但由于贪婪算法求解目标点时的局限性，导致该算法无法获得全局最优解，因此我们在贪婪策略上进行一定的松弛（通过**概率随机选取**），引入**蚁群算法**，以概率大小为启发函数，根据每条路径的距离长短设计信息素浓度，通过不断迭代形成反馈。最终得到的近似全局最优解，达到了 $1.0638 \times 10^5 \text{ m}$ 。

问题二：考虑到问题的实际背景，在增加了转弯半径限制条件后，为了尽可能飞行足够远的距离，我们使用 Dubins 曲线讨论了使得航迹在满足曲率约束条件下达到最短的情况。我们给出了飞行器飞行路线的基本模型，并给出了计算两校正点间满足题设条件的最短飞行路径的具体方法，然而由于实现难度较大，该模型仅作理论参考。

问题三：在部分校正点有一定概率无法完全校正的情况下，重新规划问题 1 优化目标下的航迹，使得成功到达终点的概率尽可能大。由于每个问题点能否校正成功的概率是一定的，我们对此进行建模，研究出对于任意给定的一条轨迹（轨迹通过的校正点的位置以及类型已知），计算出其成功到达终点的概率模型。最后，我们将其代入蚁群算法，将每条蚂蚁走过路径的概率大小作为信息素，来优化成功到达终点的概率值。最终，对于附件一中的数据，产生的轨迹成功到达终点的概率可以达到 80%。附件二中的数据，产生的轨迹成功到达终点的概率可以达到 26%。

关键词：航迹规划，多变量贪婪算法，蚁群算法，轨迹概率，全局优化

目录

一、问题重述	3
1.1 研究背景	3
1.2 研究问题	3
二、问题分析	5
2.1 问题 1 的分析	5
2.2 问题 2 的分析	5
2.3 问题 3 的分析	5
三、基本假设	6
四、符号说明	7
五、模型的建立与求解	8
5.1 问题 1 模型建立与求解	8
5.1.1 模型概述	8
5.1.2 模型建立	8
5.1.3 结果分析	13
5.2 问题 2 模型建立与求解	17
5.2.1 模型概述	17
5.2.2 模型建立	17
5.3 问题 3 模型建立与求解	18
5.3.1 模型概述	18
5.3.2 模型建立	19
5.3.3 结果分析	21
六、总结	24
七、模型评价与推广	25
7.1 模型优缺点评价	25
7.2 模型推广	25
参考文献	26

一、问题重述

1.1 研究背景

伴随着电子通信与导航技术、材料科学、控制与传感器以及推进动力等技术的不断突破和创新，飞行器系统获得了长足的发展，在军、民等领域应用越来越广泛[1]。近年来飞行器的发展趋势是采用更多的机器人技术和更优异的智能算法，安装高精度的传感器，加载更多的任务载荷，使用更高速的数据链路，从而使飞行器越来越智能化。一般情况下，在保证飞行器飞行安全的同时，应尽可能地使飞行器以最短的时间完成计划安排的任务，这就需要给无人机做飞行路径规划，即寻找一条综合代价最小的航行路径。

航迹规划是指在众多限定条件下，搜索出一条符合飞行器系统能力与特殊场景要求的从出发点到任务目标点的最佳航行路径[2]。复杂情况下的航迹快速规划是智能飞行器控制的一个重要课题，是提高飞行器飞行效率、实现精确作业的重要手段，是飞行器自主安全可靠飞行的技术条件支撑。所以，飞行器系统在军民应用领域的快速发展给航迹规划提出更高要求和更大考验，这也是航迹规划技术成为飞行器行业研究热点的一个重要因素。

1.2 研究问题

由于系统结构限制，飞行器的定位系统无法精确定位，一旦误差累积到一定程度，会导致任务失败。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。在出发地 A 点，飞行器的垂直和水平误差均为 0。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。飞行区域中存在一些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。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。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。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并最终到达目的地。在校正时还存在下述约束：

- (1) 到达终点时垂直误差和水平误差均应小于 θ 个单位。
- (2) 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。
- (3) 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。
- (4) 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。

制定航迹规划的目标是在定位误差、转弯等众多约束条件下使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。

问题 1：不考虑约束 (4) 的限制条件，针对附件 1 和附件 2 的数据分别规划从 A 点飞到 B 点的航迹。并综合考虑：(1) 航迹长度尽可能小；(2) 经过校正区域进行校正的次数尽可能少。讨论算法有效性和复杂度。

问题 2：在问题 1 的知识背景下，考虑飞行器最小转弯半径为 200m，针对问题 1 相同

数据集和优化目标，分别规划飞行器的航迹，绘出规划路径并讨论算法有效性和复杂度。

问题 3：考虑到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，假设飞行器在部分校正点能够成功校正误差的概率是 80%，如果校正失败，则校正后的剩余误差为 $\min(\text{error}, 5)$ 个单位，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。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。

二、问题分析

2.1 问题 1 的分析

问题 1 航迹规划目标是在众多约束下求得从出发点 A 到目的地 B 的最优路径,由于飞行器存在定位误差,需要在飞行过程中不断选择合适的误差校正点进行误差校正,所以问题 1 的难点就是如何确定下一个目标校正点,使得这次飞行选择能同时满足航迹长度尽可能小且校正次数尽可能少。

首先,我们根据题目对于误差校正的限制建立简单模型,分析飞行器应当如何选择校正点及其类型。其次,当飞行器在某一校正点进行误差校正后,根据优化目标确定下一个误差校正点选择策略。对于优化目标一,要实现航迹长度尽可能小,应使选择的下一个误差校正点分布在从该点到终点的直线段周围;对于优化目标二,要使校正次数尽可能少,则应在飞行器选择下一个校正点时飞行尽可能长的距离,由此得到基于当前飞行器位置的下一个目标校正点的可行解范围。对于这一点我们可以考虑贪婪策略[3],但由于贪婪算法求解目标点时的局限性,即严格参照当前的局部信息选择当前代价最小的目标,大大降低了算法获得全局最优解的概率,因此我们在贪婪策略上进行一定的松弛,根据优化目标分配不同可选校正点以不同的概率,为之后的全局优化作准备。最后,为了在全局范围内对上述结果进行优化,我们引入蚁群算法[4][5],以分配的概率值作为启发函数,以每条路径的长度作为信息素,进行全局优化。

2.2 问题 2 的分析

问题 2 在其他条件不变的情况下,增加了对飞行器转弯半径的限制,因此我们需要重新设计飞行航迹并计算飞行路径的长度来更新误差。由于优化目标不变,飞行器的转弯半径应尽可能小使得飞行路径尽可能在直线上。我们首先建立飞行器飞行路线模型,重新计算可选校正点的解集,按照问题 1 相同策略选择校正点,通过蚁群算法得到满足题设要求的最优解。

2.3 问题 3 的分析

问题 3 在问题 1 的知识背景和优化目标下,将校正点进一步分为正常点和问题点,在问题点处飞行器有一定概率校正失败,导致垂直误差或水平误差中的某一个不能完全校正为 0,并增加了第三个优化目标,即使得飞行器成功到达终点的概率尽可能大,因此如何衡量飞行器到达终点的概率是本小题的难点。在问题 1 中我们已经获得了满足前两个优化目标的可行解,随着迭代次数的增加会逐渐向全局最优解收敛。为了把飞行器在问题点处的概率校正信息量化到飞行过程中,我们选择通过概率统计的方法,针对轨迹的已知特征建立一个概率模型。

由于每个问题点能否校正成功的概率是一定的,受蚁群算法的启发,我们首先通过二叉树模型算出蚂蚁经过的每一条路径的成功概率,将成功的概率大小作为信息素,并融合解决问题 1 时用的信息素模型,通过不断迭代形成正反馈,使得蚂蚁最终都跑向成功概率较大的路径。

三、基本假设

问题一：

1. 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. 不考虑飞行器飞行过程中产生的垂直误差和水平误差的方向性；
3. 假设校正点处的校正区域可以抽象为校正点所在的空间点；

问题二：

4. 假设飞行器在转弯半径所构成的球体上可以进行一定弧度的转弯；
5. 假设飞行器转弯时定位误差不变；
6. 假设飞行器转弯时总是以最小转弯半径转弯；

问题三：

7. 假设飞行器到达校正点时即可知道在该点处能否校正成功。

四、符号说明

参数符号	符号含义
α_1	垂直误差校正的垂直误差最大阈值
α_2	垂直误差校正的水平误差最大阈值
β_1	水平误差校正的垂直误差最大阈值
β_2	水平误差校正的水平误差最大阈值
θ	终点处允许的垂直和水平误差最大阈值
δ	每飞行 1m，垂直误差和水平误差增加的单位值
α	信息素重要程度因子
β	启发函数重要程度因子
ρ	信息素挥发因子
$\tau_{ij}(t)$	第 t 次迭代时校正点 i 和 j 间的信息素浓度
$P_{ij}^k(t)$	第 t 次迭代蚂蚁 k 从校正点 i 转移到校正点 j 的概率
d_{ij}	校正点 i 和 j 之间的距离
L_k	第 k 只蚂蚁经过的路径长度
$\eta_{ij}(t)$	蚂蚁从校正点 i 转移到校正点 j 的期望程度
$\Delta\tau_{ij}$	所有蚂蚁在校正点 i 和校正点 j 之间释放的信息素总和

字母 A 和 B 分别表示三维空间中的出发点和目的地。

五、模型的建立与求解

5.1 问题 1 模型建立与求解

5.1.1 模型概述

飞行器航迹规划的目的是在众多约束下求得从出发点 A 到目的地 B 的最优路径,若没有约束条件,则最优路径是从 A 到 B 的直线段,如图 5.1 中的 AB 段。本题的难点在于存在飞行定位误差,需要在飞行过程中寻找合适的校正点进行误差校正,即飞行节点的搜索问题。飞行器飞行空间如图 5.1 所示。

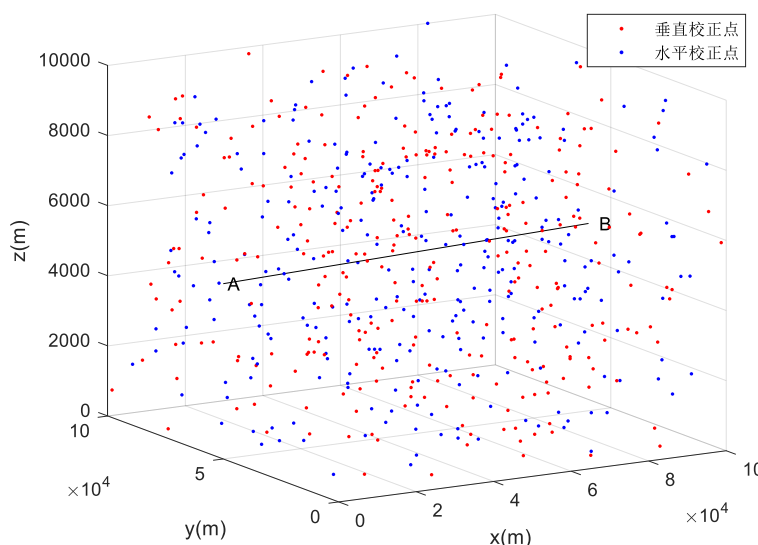


图 5.1 飞行器的飞行空间

我们首先想到的是利用飞行器的误差校正限制确定一个大致的球形搜索范围,并根据优化目标确定代价函数,选择代价最小的目标校正点,即贪婪算法的思想。但由于搜索范围过大,算法复杂度过高。同时,算法存在明显可以优化的地方,比如优先考虑上述搜索范围内当前点到终点 B 之间的空间校正点。因此我们建立了一个目标校正点搜索模型,极大缩小了搜索范围,并获得一个局部最优解。又因为搜索过程是从当前点出发,未考虑全局信息,因此我们在单条航迹规划建模的基础上引入了蚁群算法模型,进行多条轨迹的全局优化。

5.1.2 模型建立

5.1.2.1 单条航迹规划建模

在明确了模型输入和输出后,为了生成一条满足条件的航迹,首先需要确定约束条件中对飞行器到达校正点时的垂直误差和水平误差限制的影响,因此需要建立飞行误差限制子模型,在此基础上才能够进行后续的目标校正点搜索。此部分建模框架如下:

单条航迹规划建模:

输入: 飞行器的出发点 A 和目的地 B;

分布在三维空间中的无规律误差校正点;

输出: 从 A 到 B 的由误差校正点组成的轨迹;

每个误差校正点未校正前的垂直误差和水平误差；

过程：

1. 建立飞行器飞行误差限制模型；
2. 从当前点出发计算满足误差限制条件的理想飞行距离；
3. 根据理想校正点确定目标校正点类型和搜索范围；
4. 根据目标校正点搜索模型选取下一个校正点；
5. 得到一条从 A 到 B 的局部最优航迹规划；

1) 飞行误差限制子模型

因为飞行器飞行距离和垂直水平误差成比例关系，所以为了叙述问题方便，直接用垂直水平误差（无量纲）来图解飞行器可以飞行的距离范围，实际编程时会换算到具体的三维空间距离（以 m 为单位）。对于本题中的航迹快速规划，首先需要的是在垂直和水平误差限制下对飞行器飞行模型的建立，如图 5.2 所示。

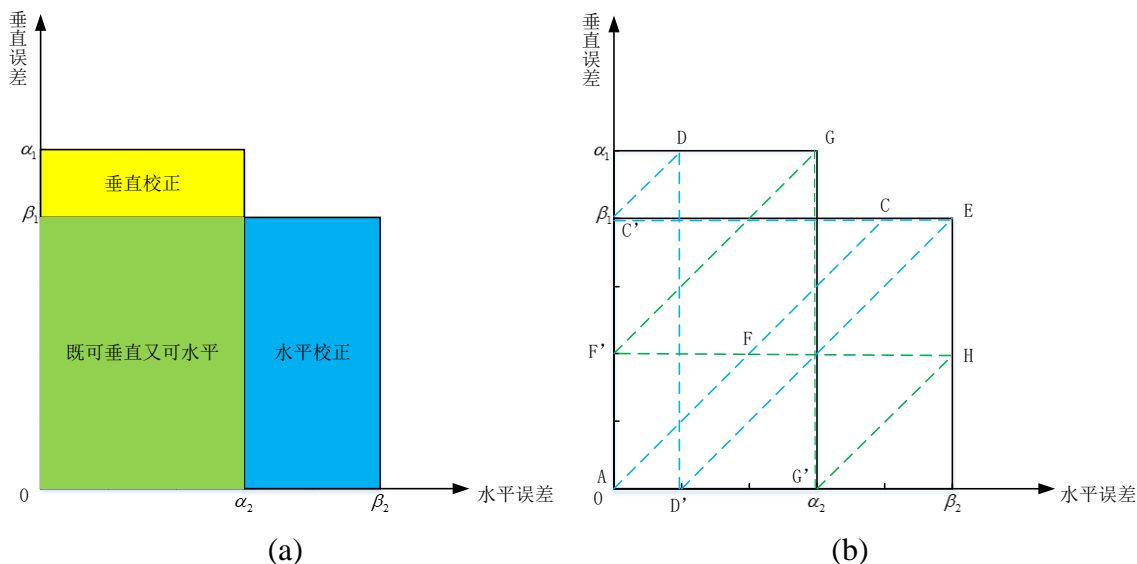


图 5.2 飞行误差限制模型（参数尺度来自数据集 1）

图 5.2 (a) 展示了飞行器在飞行误差限制下的校正模型，水平轴和竖直轴分别表示飞行器在飞行过程中的实时水平误差和垂直误差。由于飞行器每飞行 1m 其垂直和水平误差均增加 δ 个单位，在图 5.2 中则表示为以 45° 向右上方飞行，飞行器经过垂直校正后会落在水平误差轴上，经过水平误差校正会落在垂直误差轴上，虽然误差状态不同，但**飞行器从每个误差校正点出发时均是以 45° 向右上方飞行**。若航迹落在黄色区域，则由于误差校正限制条件，飞行器只能进行垂直校正，落在蓝色区域则只能进行水平校正，而落在绿色区域则代表飞行器既可以垂直校正又可以水平校正。图 5.2 (b) 用于计算到达下一个目标校正点时理想的飞行距离，便于确定下一个目标点的搜索类型和搜索范围。因为优化目标中要求校正次数尽可能少，所以此理想的飞行距离应在满足约束误差限制下尽可能的长，**这就限制了飞行器的校正点类型最优选择是垂直误差校正和水平误差校正交替进行**。例如，飞行器从水平误差轴上（表明刚经过垂直误差校正）的某个校正点出发后，当飞行器尚处于绿色区域时又进行一次垂直误差校正，再次从水平误差轴上出发后最终到达蓝色的水平校正区域，这种情况的校正次数比直接从水平误差轴上出发后直接进入蓝色的水平校正区

域多。

飞行器在飞行过程中还存在如何确定下一个目标点的搜索范围问题。若以图 5.2 (b) 中 AC 段作为可飞行的最长距离，并假设在 C 点存在一个理想的误差校正点，则经过 C 点水平校正后 (C 点在蓝色区域，只可以水平校正) 到达 C' 点。此时飞行器以 C' 点的误差状态出发，则可飞行的最长距离只有 $C'D$ 段，因为飞行器可选择的下一个目标点范围过小，容易造成搜索失败的情况。这里提出初步的解决方法是在 AC 段中间选择合适的点，使得后续的飞行器理想最大飞行长度近似相等。经过实际测试，以 AC 段的中点 F 为第一次可飞行的理想校正点，把 AF 段对应的距离作为飞行器第一次搜索下一个目标点的最长理想半径，在此范围内搜索符合误差限制的水平误差校正点，校正后飞行器状态将落在 F' 点，此时飞行器能够飞行的最大理想距离 $F'G$ 是之前 $C'D$ 的 3 倍，即与上一次误差校正最大理想距离接近。最终的理想飞行循环区段从 $ACC'DD'EC'$ 变为 $AFF'GG'HF'$ ，可以看到飞行器向右上方飞行的各小段距离相差不大，从而可以极大减少因搜索范围突然变小而搜索不到目标校正点的情况，提高了算法的有效性。

2) 目标校正点搜索子模型

根据飞行误差限制子模型可以得到飞行器从当前点出发的理想最大飞行距离，但正如问题一分析中提到的，此时的搜索半径很大，算法效率很低，因此我们引入理想校正点的概念，利用当前点与终点的直线段获得理想校正点，以此理想校正点为球心进一步缩小搜索范围。因为理想目标校正点兼顾了当前点到终点的最短航迹长度要求和满足飞行误差限制的最少校正次数要求，若空间中实际存在此校正点，则是可以选择的最优目标点。目标校正点搜索模型如图 5.3 所示。

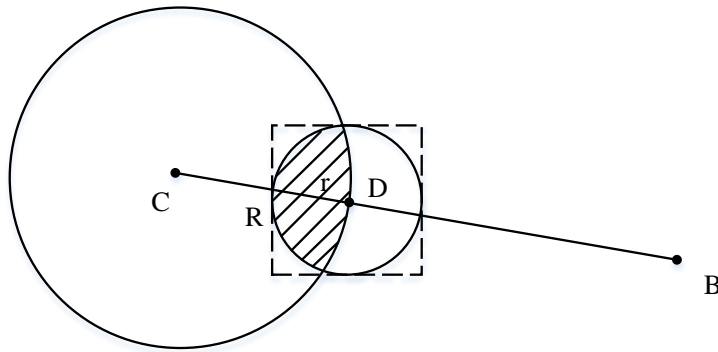


图 5.3 目标校正点搜索模型

图 5.3 中从飞行器当前位置 C 出发搜索下一个目标校正点，实际场景是三维空间，其中 B 为终点，根据上一步飞行误差限制子模型确定当前理想的最大飞行距离 R ，若不考虑优化目标，则在以 C 为球心， R 为半径的球体搜索范围内可任意选择符合目标校正类型的下一个目标校正点。 D 为当前点与终点 B 的连线与上述球体的交点，即为理想校正点，因为若要满足航迹长度尽可能短的优化目标，所选目标点应尽可能落在 CD 段上，同时又要满足校正次数尽可能少的优化目标，该目标点距离当前点 C 应尽可能远，综合两个优化目标之后 D 即为理想校正点。虽然实际的三维空间几乎不存在理想校正点，但可以在理想校

正点 D 附近搜索满足条件的目标校正点。这里根据三维空间中校正点的密度大小确定搜索半径 r ，实际编程时通过虚线正方形来进行初步筛选，又因为目标校正点必须在以 C 为球心的大球体中（要满足误差限制硬约束），所以**最后的搜索范围限定在图 5.3 的阴影部分，这就极大地减少了目标校正点的搜索数量，提高了算法效率。**

确定了图 5.3 中阴影部分的目标校正点搜索范围后，直接根据范围内目标点到理想校正点的距离，选择距离最短的点作为下一个目标校正点。这种选取策略是贪婪算法[6]的思想，从而可以获得一条较好的航迹规划轨迹。

3) 模型改进

根据贪婪策略获得的单条航迹规划有其固有的局限性，因为飞行器在当前点搜索目标校正点时只参考了局部信息，利用终点 B 获得理想校正点只是为了使其获得局部最优解，因此我们还需在此基础上进行模型改进。首先需要思考的就是如何增加选取目标校正点的多样性，使飞行器在基于理想校正点选取目标点时有一定概率跳出局部最优解，防止无法达到全局解。

航迹规划一个考虑全局信息的主流算法是蚁群算法，这里我们通过在目标点周围建立一个概率分布，从而与蚁群算法中的信息素结合，利用局部信息和全局信息进行迭代优化，相辅相成，最后达到近似全局最优解。蚁群算法优化模型如下：

蚁群算法优化建模：

输入：根据单条航迹规划模型获得的 N 条轨迹；

目标搜索模型中的理想校正点和候选目标点；

输出：从 A 到 B 的由误差校正点组成的轨迹；

每个误差校正点未校正前的垂直误差和水平误差；

过程：

1. 对范围内的目标点建立概率分布模型，概率选取下一个校正点；
 2. 得到多条从 A 到 B 的局部最优航迹规划；
 3. 对局部可行解运用蚁群算法进行全局优化；
 4. 改进蚁群算法，通过信息素和启发函数融合形成新的概率模型；
 5. 迭代 N 次获得全局最优解。
-

5.1.2.2 蚁群算法模型

(1) 模型建立

蚂蚁在寻找食物源时，会在其经过的路径上释放一种信息素，并能够感知其他蚂蚁释放的信息素。信息素浓度越高，表示对应的路径距离越短。通常，蚂蚁会以较大概率优先选择信息素浓度较高的路径，并释放一定信息素，以增强该条路径上的信息素浓度，形成一个正反馈。最终蚂蚁能够找到一条从巢穴到食物源的最佳路径，即最短距离。

不失一般性，设整个蚂蚁群体中蚂蚁的数量为 m ，校正点 i 和校正点 j 之间的距离为 $d_{ij}(i, j = 1, 2, \dots, n)$ ，第 t 次迭代时连接路径上的信息素浓度为 $\tau_{ij}(t)$ 。初始时刻，各校正点连接路径上的信息素浓度相同。

设 $P_{ij}^k(t)$ 表示第 t 次迭代蚂蚁 k 从校正点 i 转移到校正点 j 的概率，其计算公式为

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in allow_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta} \quad (5.1)$$

其中， $\eta_{ij}(t)$ 为启发函数，表示蚂蚁从校正点 i 转移到校正点 j 的期望程度。关于如何衡量启发函数的值，我们想了三个方案：

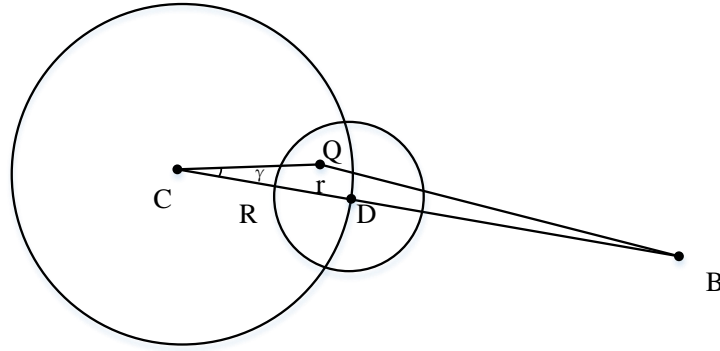


图 5.4 启发函数选择

如图所示， C 表示当前校正点， B 表示终点。理想校正点为 D 。我们需要在可行解范围（即大圆与小圆的交集区域）内按概率分配校正点 Q 。

方案 1：基于可行解 Q 点与理想校正点 D 的距离分配概率（距离越小，概率越大）

方案 2：基于向量 QC 与 QB 的内积分配（内积越小，夹角越大，概率越大）

方案 3：基于 QC 与 BC 的夹角分配（夹角越小，概率越大）

最终通过实验测试，我们选择方案 1 分配的概率作为启发函数。

在蚂蚁释放信息素的同时，各路径上的信息素也在逐渐消失。因此，当所有蚂蚁完成一次路径后，各校正点连接路径上的信息素浓度需进行实时更新。设参数 ρ ($0 < \rho < 1$) 表示信息素的挥发程度，则：

$$\begin{cases} \tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k \end{cases} \quad (5.2)$$

其中， $\Delta\tau_{ij}$ 表示所有蚂蚁在校正点 i 和校正点 j 之间释放的信息素总和。针对蚂蚁释放的信息素，M. Dorigo 等人曾给出 3 种不同的模型，我们选择 ant cycle system 模型，其计算公式如下：

$$\Delta\tau_{ij}^k = \begin{cases} Q / L_k & \text{第 } k \text{ 只蚂蚁从校正点 } i \text{ 到校正点 } j \\ 0 & \text{其他} \end{cases} \quad (5.3)$$

其中， Q 为常数，表示蚂蚁走一遍路径所释放的信息素总量， L_k 为第 k 只蚂蚁经过路径的长度。

(2) 基本步骤

利用蚁群算法计算此题需要以下几个步骤，如图 5.5 所示。

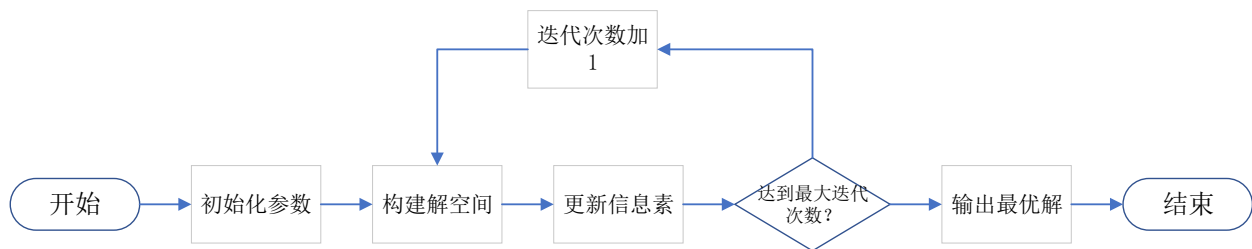


图 5.5 蚁群算法基本步骤

1) 初始化参数

在计算之初，需要对相关参数进行初始化，如蚁群规模 m 、信息素重要因子 α 、启发函数重要因子 β 、信息素挥发因子 ρ 、信息素释放总量 Q 、最大迭代次数 $iter_max$ 等

2) 构建解空间

将所有蚂蚁置于 A 点，对每个蚂蚁，按照式(5.1)计算其下一个待访问校正点。每到达一个校正点，计算一次，直到蚂蚁到达 B 点。

3) 更新信息素

计算各个蚂蚁经过的路径长度 L_k ，根据式(5.2)、(5.3)对各校正点上的信息路径进行更新。

4) 判断是否终止

当未达到迭代次数，则返回步骤 2，重新构建解空间。否则，终止计算，输出最优解。具体代码见附录。

5.1.3 结果分析

根据贪婪策略获得的轨迹结果如图 5.6 所示，校正点校正前误差如表 5-1、表 5-2 所示。对于数据集 1，需要经过 8 个校正点，飞行总距离为 1.1509×10^5 m；对于数据集 2，需要经过 12 个校正点，飞行总距离为 1.1160×10^5 m。

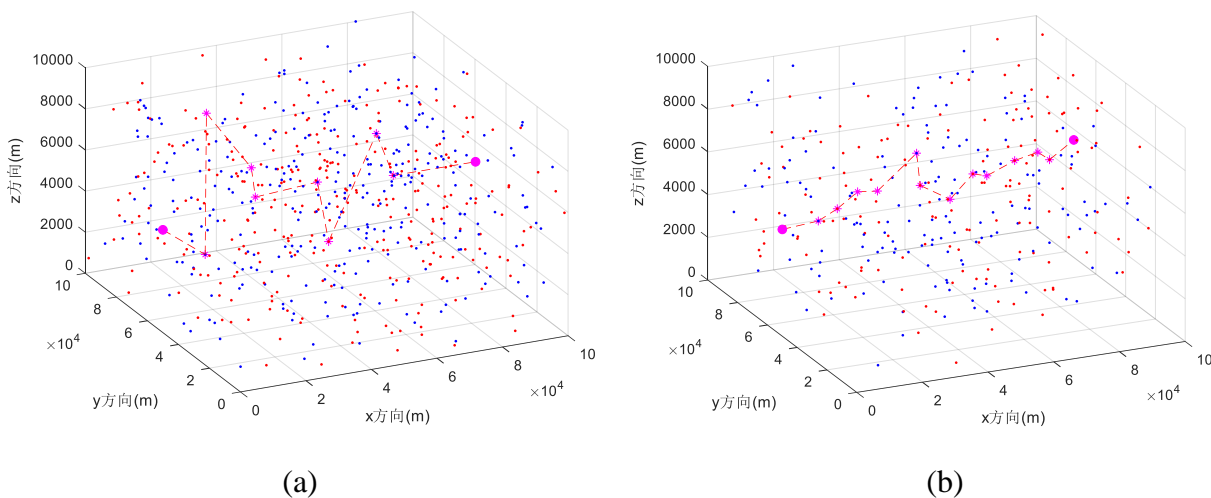


图 5.6 数据集 1 和数据集 2 的航迹规划

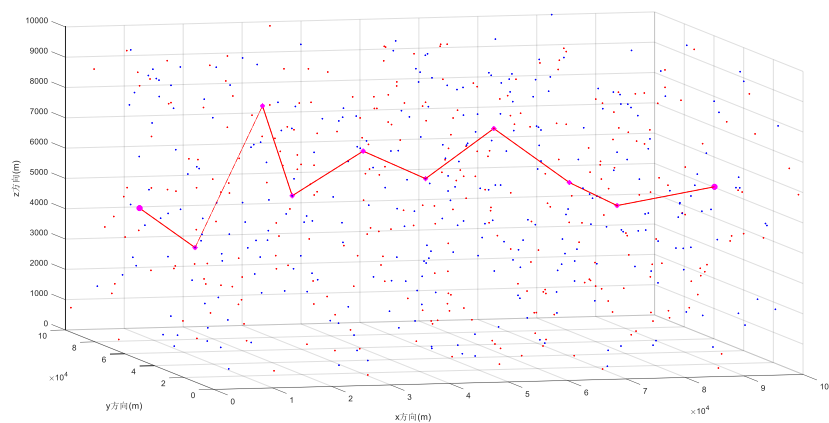
表 5-1: 数据集 1 校正点校正前误差

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	0.00	0.00	出发点 A
431	16.27	22.00	01
417	23.47	7.21	11
151	16.80	24.01	01
237	20.23	3.43	11
278	17.94	21.37	01
375	23.48	5.54	11
448	18.59	24.13	01
485	24.32	5.73	11
612	23.57	29.30	终点 B

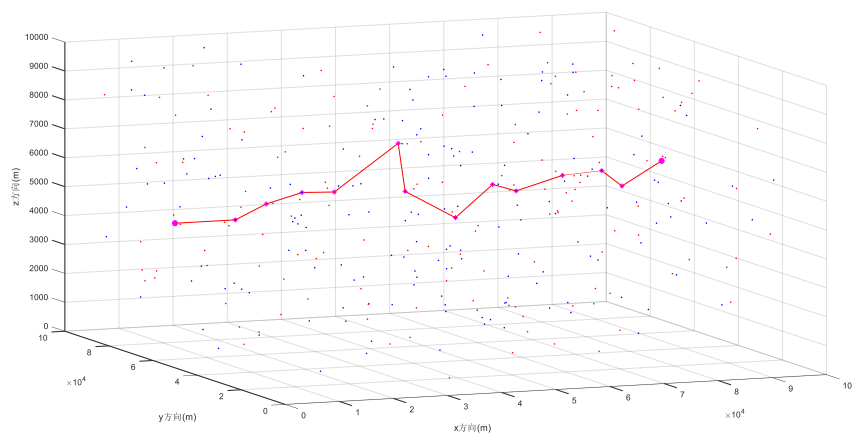
表 5-2: 数据集 2 校正点校正前误差

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	13.29	19.02	01
114	18.62	5.33	11
8	13.92	19.26	01
309	19.45	5.52	11
121	11.25	16.78	01
123	16.60	5.35	11
49	11.79	17.14	01
160	18.30	6.51	11
92	5.78	12.29	01
93	15.26	9.48	11
61	9.83	19.32	01
292	16.39	6.55	11
326	6.96	13.51	终点 B

经过蚁群算法模型 **50** 次迭代后，对于数据集 1，需要经过 **8** 个校正点，飞行总距离为 $1.0638 \times 10^5 \text{ m}$ ；对于数据集 2，需要经过 **12** 个校正点，飞行总距离为 $1.1193 \times 10^5 \text{ m}$ 。飞行路径如图 5.7 所示，校正点校正前误差如表 5-3、表 5-4 所示。



(a) 数据集 1



(b) 数据集 2

图 5.7 数据集 1 和数据集 2 的航迹规划

表 5-3: 数据集 1 校正点校正前误差

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	9.63	9.63	出发点 A
521	21.76	12.13	01
64	11.42	23.55	11
80	23.40	11.98	01
170	10.46	22.43	11
278	21.89	11.44	01
369	13.31	24.75	11
214	22.33	9.02	01
397	16.97	25.99	11
612	9.63	9.63	终点 B

表 5-4: 数据集 2 校正点校正前误差

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	0.00	0.00	出发点 A

163	13.29	13.29	01
114	18.62	5.33	11
8	13.92	19.26	01
309	19.45	5.52	11
121	11.25	16.78	01
123	16.60	5.35	11
49	11.79	17.14	01
160	18.30	6.51	11
92	5.78	12.29	01
93	15.26	9.48	11
61	9.83	19.32	01
292	16.39	6.55	11
326	6.96	13.51	终点 B

从图 5.6 和图 5.7 的轨迹结果对比来看，校正次数都一样。对于数据集 1，通过蚁群算法进行全局优化后的结果明显更好，不仅是航迹距离更短，还因为其三维空间轨迹更平滑。图 5.6（a）的空间航迹明显比图 5.7（a）的空间航迹更抖，这是因为贪婪策略每次都以理想飞行距离的最大限度选择目标点，而图 5.6（b）的空间航迹相对 5.7（b）来说呈现忽长忽短的情况，是因为贪婪策略在第一次选择目标校正点时是以最大理想飞行距离为标准，而不考虑后续的可飞行的理想飞行距离长短。对于数据集 2，通过贪婪策略获得的局部最优航迹和通过蚁群算法全局优化后的航迹距离相近，说明空间校正点变稀疏后，不用蚁群算法也能获得很好的近似全局最优解。但贪婪策略受到空间校正点稀疏程度的影响到底有多大，还需要更多的数据集进行测试。

由于贪婪策略本身参考的是局部信息，所以获得的最优解有其固有的局部性缺陷，虽然对于数据集 2，两种方法获得了相似的结果，但我们并不能因此判断贪婪策略获得的解就是全局最优解，而通过蚁群算法全局优化后的结果我们可以确定其就是近似全局最优解。

（1）有效性分析

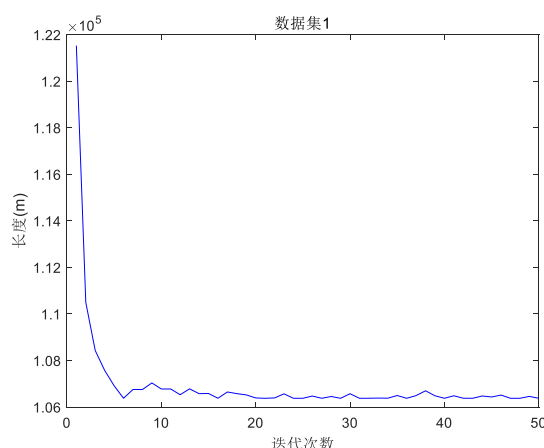


图 5.8 迭代次数与路径长度关系

图 5.8 是使用蚁群算法迭代过程中，数据集 1 和数据集 2 中飞行总路径与迭代次数的关系。由图可知，经过蚁群算法的不断迭代，飞行路径越来越短，算法是收敛的。此外，不同数据集使用此算法均能收敛，体现了算法的有效性和普适性。

(2) 算法复杂度分析

由于我们的算法在一开始就有多个独立个体同时进行并行计算，大大提高了算法的计算能力和运行效率。数据集 1 和数据集 2 的运行时间分别是 34.566898 秒和 49.060766 秒。由于数据集 2 点比较稀疏，所以在寻找下一个校正点时，我们扩大了搜索范围，程序运行时间变长。

5.2 问题 2 模型建立与求解

5.2.1 模型概述

问题 2 在问题 1 的基础上增加了对飞行器转弯半径的限制，由于飞行器不能及时转弯，每到达一个校正点后，我们首先要确定一个半径为最小转弯半径的球面，使得飞行器的飞行路径为直线时与此球面相切，这样就可以保证转弯距离所形成的弧度最小。同时，由于飞行器不能完全按照直线飞行，则势必会导致飞行距离增加，我们需要对每段路径产生的误差重新验证是否满足误差限制要求，以此更新我们的轨迹。与问题 1 相同，我们依旧采用蚁群算法来优化轨迹。

5.2.2 模型建立

为了计算飞行转弯球面上的转弯弧长，以及离开转弯球面后的直线飞行距离，我们需要理清其中的距离关系，因此提出建立飞行转弯模型。如图 5.9 所示，图中 C 为飞行器当前所处的校正点，S 为上一个校正点，D 为从 C 点出发要飞行的下一个目标校正点，O 为飞行器转弯半径构成的球体的球心，R 为最小转弯半径，P 为从 D 点出发与上述转弯球面所形成的切线。DP 段存在多解的情况，这里只画出其中最合理的一段轨迹，其余解应当舍弃。由此可以建立从当前点 C 出发到达 D 点的飞行转弯模型，根据几何约束关系可以求得切点 P 的空间坐标，结合弧度公式可以计算出 CP 段的弧长，从而可以确定 C 到 D 的飞行长度。由于算法实现难度较大，本题的模型仅作理论参考。

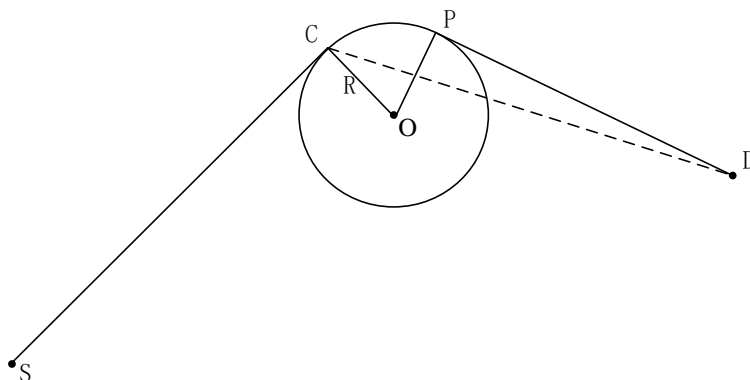


图 5.9 飞行转弯模型示意图

具体求解步骤如下：

- 1) 确定圆心 O 坐标

$$\begin{cases} |(x_o, y_o, z_o) - (x_c, y_c, z_c)| = R & OC = R \\ (x_s - x_c, y_s - y_c, z_s - z_c)' \cdot (x_o - x_c, y_o - y_c, z_o - z_c) = 0 & SC \perp OC \\ \begin{vmatrix} x_o - x_s & y_o - y_s & z_o - z_s \\ x_c - x_s & y_c - y_s & z_c - z_s \\ x_d - x_s & y_d - y_s & z_d - z_s \end{vmatrix} = 0 & O \text{与} S、C、D \text{共面} \end{cases} \quad (5.4)$$

2) 确定 P 点坐标

$$\begin{cases} |(x_o, y_o, z_o) - (x_p, y_p, z_p)| = R & OP = R \\ (x_p - x_d, y_p - y_d, z_p - z_d)' \cdot (x_p - x_o, y_p - y_o, z_p - z_o) = 0 & PD \perp PO \\ \begin{vmatrix} x_p - x_s & y_p - y_s & z_p - z_s \\ x_c - x_s & y_c - y_s & z_c - z_s \\ x_d - x_s & y_d - y_s & z_d - z_s \end{vmatrix} = 0 & P \text{与} S、C、D \text{共面} \end{cases} \quad (5.5)$$

3) 确定在圆弧上飞行的夹角

$$R^2 = (x_o - x_c, y_o - y_c, z_o - z_c) \cdot (x_o - x_p, y_o - y_p, z_o - z_p) \cdot \cos(\theta) \quad (5.6)$$

4) 确定飞行器飞行距离

$$L = CP \text{弧长} + PD = R\theta + |(x_d - x_p, y_d - y_p, z_d - z_p)| \quad (5.7)$$

5) 验证此飞行距离产生的误差是否满足条件，若不满足，重新选择校正点 D。

5.3 问题 3 模型建立与求解

5.3.1 模型概述

问题 3 要求在部分校正点有一定概率无法完全校正的情况下，重新规划满足问题 1 约束条件的航迹，并使成功到达终点的概率尽可能大。由于引入了概率校正的因素，并增加了第三个优化目标，因此如何衡量飞行器到达终点的概率是本小题的难点。在问题 1 中我们已经获得了满足前两个优化目标的可行解，随着迭代次数的增加会逐渐向全局最优解收敛。为了把飞行器在问题点处的概率校正信息量化到飞行过程中，我们通过概率统计的方法，针对轨迹的已知特征建立概率模型。

由于每个问题点能否校正成功的概率是确定的，即校正成功的概率是 0.8，校正失败的概率是 0.2，我们可以利用二叉树结构来确定所有可能的概率方案，最终计算出飞行器成功到达终点 B 时的综合概率，将此概率大小作为新的蚁群算法信息素更新策略，通过不断迭代，使得蚂蚁最终都跑向成功概率较大的路径。此部分的航迹规划概率模型如下：

航迹规划概率建模：

输入：从 A 到 B 的一条近似全局最优轨迹；

误差校正点中的问题校正点和正常校正点；

输出：从 A 到 B 的由误差校正点组成的最优轨迹；
每个误差校正点未校正前的垂直误差和水平误差；
过程：

1. 利用二叉树建立飞行器在校正点处的概率模型；
 2. 对输入的路径遍历所有可能的概率方案；
 3. 综合整条路径的概率获得成功到达终点的最终概率；
 4. 根据最终概率调整蚁群算法的信息素跟新策略；
 5. 迭代 N 次，得到新的从 A 到 B 的近似全局最优解；
 6. 循环多次，使得最终概率增大，得到全局最优轨迹。
-

5.3.2 模型建立

5.3.2.1 概率模型

一条路径成功到达终点的概率取决于该路径上问题点的个数和各校正点之间的距离等，利用二叉树结构可以方便地表示出在各个环节点处的概率分叉。树的节点表示输入轨迹中的校正点，节点的权值表示成功到达下一个校正点的概率。以问题 1 中由数据集 1 获得的近似全局最优轨迹为例，图 5.10 展示了根据这条航迹建立的以二叉树结构展示的概率模型。

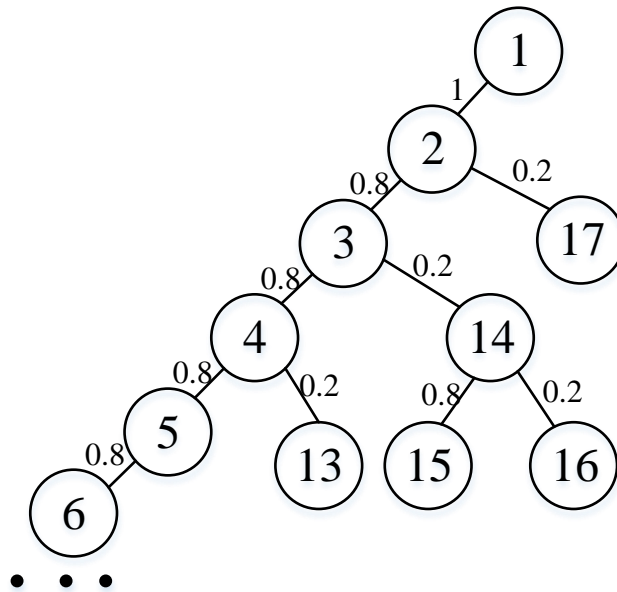


图 5.10 二叉树结构表示概率模型的示意图

通过图 5.10 的示意图可以对概率模型进行初步分析，设出发点 A 在节点 1，其选择节点 2 的概率为 1，因为出发点是正常点。而节点 2 是问题点，有 0.2 的概率校正失败，对应的就有 0.8 的概率到达节点 3，此时节点 3 也有 0.2 的概率校正失败，产生对应的概率分支 14，因为校正失败并不意味着飞行器误差超过限制，因此需要根据节点 14 自身的剩余误差状态判断能否进行下一步飞行，若不能飞行则判定为此分支不通。若飞行器通过当前概率选择的路径最终能够到达终点 B，则回溯到概率分支节点递归判断其它的子节点，直至遍历完这条轨迹整个的概率空间。具体二叉树数据结构通过 MATLAB 代码实现，将各个成功到达终点的分支上的概率分别相乘并相加（同分支的节点权值概率相乘，然后不同分支相加），就能得到该条轨迹成功到达终点的最终概率值。过程如图 5.11 所示。

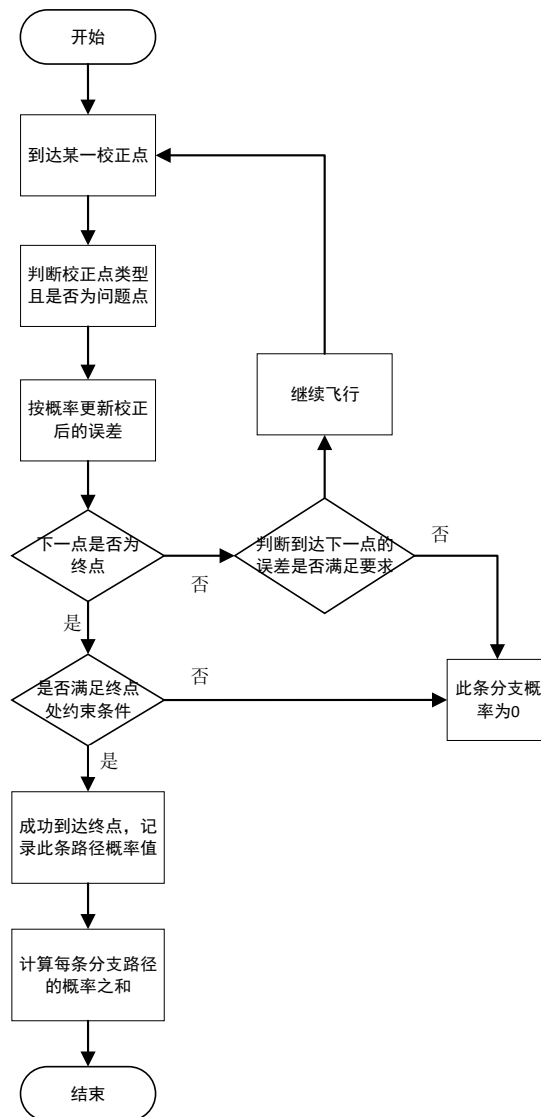


图 5.11 计算某一航迹的成功概率流程图

表 5-5 展示了一条航迹（见图）的概率计算情况，举例说明指定轨迹概率的计算方法。表的左边一列表示校正点的编号（仅显示了不确定的点），由表中可以看到飞行器成功到达终点的所有可能情况以及对应的概率。通过该表可以计算出该条轨迹能够成功到达终点的概率为 0.8，即将所有可能的概率求和得到。

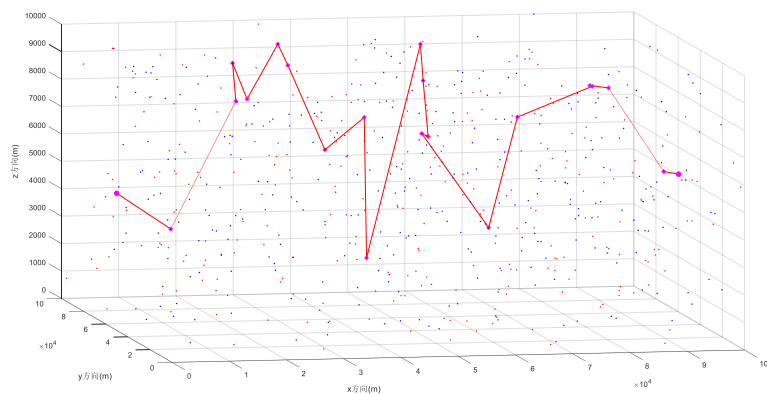


图 5.12 概率计算说明图（仅为说明计算过程）

表 5-5: 轨迹概率计算举例

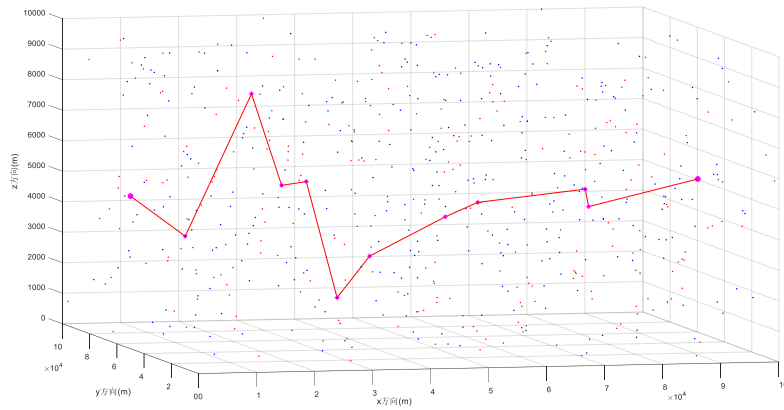
编号	1	2	3	4	5	6	7	8
521	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
64	0.8	0.8	0.8	0.8	0.2	0.2	0.2	0.2
211	0.8	0.8	0.2	0.2	0.8	0.8	0.2	0.2
124	0.8	0.2	0.8	0.2	0.8	0.2	0.8	0.2
其它	1	1	1	1	1	1	1	1
概率	0.4096	0.1024	0.1024	0.0256	0.1024	0.0256	0.0256	0.0064

5.3.2.2 蚁群算法修改

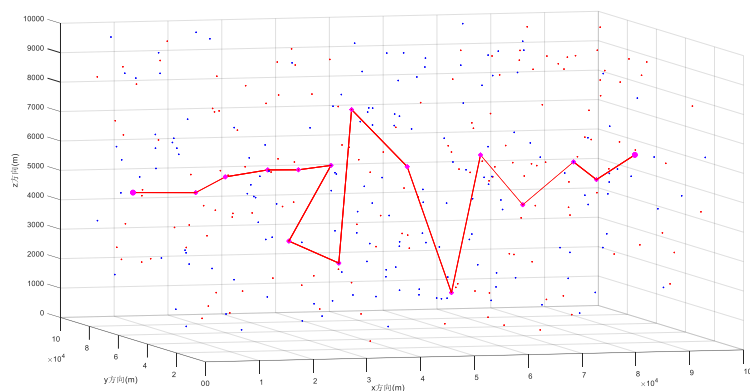
在问题 1 中，我们将蚂蚁跑的每条路径的距离大小作为信息素，基于可行解 Q 点与理想校正点 D 的距离分配概率作为启发函数，来进行每个校正点的概率更新。在此问中，我们将蚂蚁经过的路径的最终成功概率作为信息素，蚂蚁优先选择信息素浓度较高的路径，并形成正反馈，经过不断迭代，寻找概率最高的路径。

5.3.3 结果分析

因为在问题 3 的上述模型计算中，只有第一次根据问题 1 的近似全局最优解得到的二叉树概率模型考虑到了全局航迹长度尽可能少和校正次数尽可能少，而根据此概率模型得到的飞行器到达终点的概率融入信息素的时候没有再考虑上述全局航迹长度这个优化目标，经过多次迭代循环后，修改后的蚁群算法中的启发函数的作用会越来越低，而模型对增加成功概率越来越感兴趣，因此结果存在一定的局限性。此问的数据集 1 和数据集 2 航迹规划结果如图 5.12 所示。



(a) 数据集 1



(b) 数据集 2

图 5.13 满足概率尽可能大的数据集 1 和数据集 2 航迹结果

从结果中可以看出，问题 3 的图 5.13 (a) 比问题 1 的图 5.6 (a) 增加了 2 个校正点，两者的航迹波动大体相似，说明问题点带来的误差校正的不确定性对航迹规划确实产生了负面影响，但总体的影响不是很大，因为到达问题点时只有 0.2 的概率失败，校正失败也只是将对应类型剩余误差降低到 5 个单位，大大降低了误差校正失败对整体航迹规划的冲击。

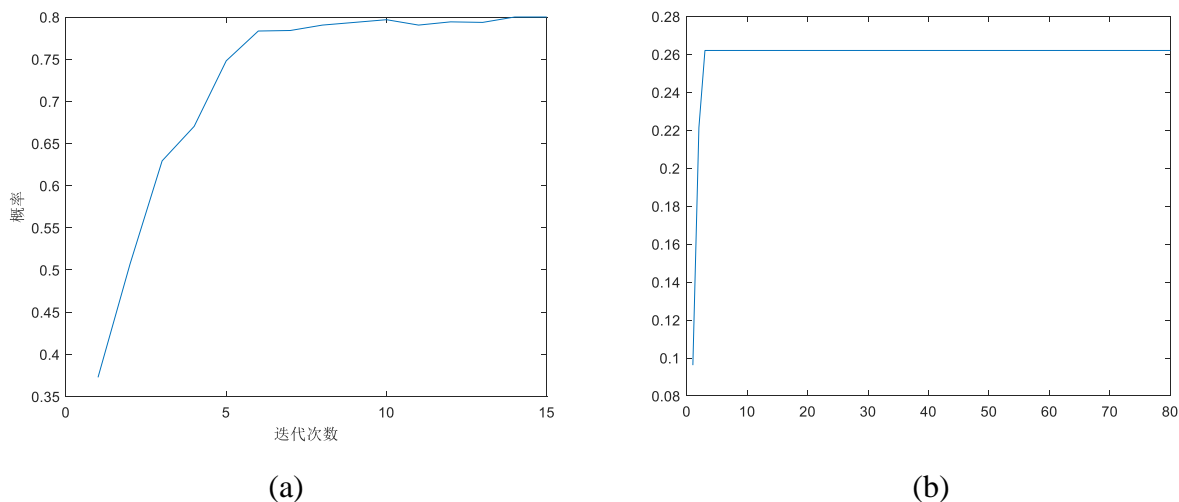


图 5.14 数据集 1 和数据集 2 的轨迹成功概率与迭代次数关系

从图 5.14 (a) 的迭代次数-概率曲线可以知道，随着修改后的蚁群算法迭代次数增加，飞行器成功到达终点的概率在逐渐增加，并收敛于 0.8。这是因为出发点 A 附近密集分布着问题点，飞行器从 A 点出发后极大概率遇到至少一个问题点，这就使得成功到达终点的概率收敛到此问题点成功校正的概率数值，也就是 0.8。图 5.14 (b) 数据集 2 的结果分析类似数据集 1。

数据集 1 和数据集 2 在校正点校正前误差如表 5-6、5-7 所示。

表 5-6：数据集 1 校正点校正前误差

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	0.00	0.00	出发点 A
521	9.63	9.63	01
64	21.76	12.13	11

80	11.42	23.55	01
237	16.05	4.63	11
371	6.83	11.46	01
183	13.87	7.04	11
194	13.61	20.65	01
450	19.59	5.98	11
594	18.05	24.03	01
67	20.28	2.23	11
612	19.32	21.55	终点 B

表 5-7：数据集 2 校正点校正前误差

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	0.00	0.00	出发点 A
163	13.29	13.29	01
114	18.62	5.33	11
8	13.92	19.26	01
309	19.45	5.52	11
255	11.71	17.23	01
148	19.77	8.07	11
90	11.18	19.25	01
156	18.88	7.70	11
33	10.70	18.40	01
289	19.94	9.24	11
50	7.19	16.43	01
323	15.12	7.93	11
61	9.29	17.22	01
292	15.84	6.55	11
326	6.96	13.51	终点 B

六、总结

飞行器航迹快速规划是飞行器控制的重要手段，在飞行过程中由于复杂的环境和自身的系统结构限制，飞行器的定位误差会随着飞行距离不断增加，若误差过大则飞行任务失败，因此及时地校正定位误差是航迹规划的一项重要任务。本文从飞行器控制的误差校正入手，结合路径选择，对从出发点到目的地的航迹规划进行一般性建模。

问题一：根据校正点处的垂直误差限制和水平误差限制，建立飞行误差限制模型，确定理想飞行距离后再建立具体的目标校正点搜索模型。首先考虑到的是贪婪策略，即确定当前点飞向目标点的代价选取策略。对于优化目标一，要实现航迹长度尽可能小，应使选择的下一个误差校正点分布在从该点到终点的直线段周围；对于优化目标二，要使校正次数尽可能少，则应在飞行器选择下一个校正点时飞行尽可能长的距离。综合以上两点考虑，根据当前点与终点的位置建立理想校正点，从而在理想校正点附近根据距离大小选取距离理想点最近的目标点。其次，由于贪婪算法求解目标点时的局限性，我们引入蚁群算法来进行全局优化，并对理想点附近的点根据距离建立概率分布，以概率大小选择飞行目标点，从而可以获得多个可行解，并把此概率作为蚁群算法的启发函数。再次，根据每条可行解航迹的距离长短设计蚁群算法信息素浓度，通过不断迭代形成正反馈，最终得到全局最优解。

问题二，根据最小飞行转弯半径建立当前点到目标点的飞行转弯模型，可以利用空间几何约束关系求解飞行器转弯球面与目标点的切点，从而确定飞行过程中的弧线长度和直线段长度，获得最短航行距离。在此基础上的目标校正点搜索模型和通过蚁群算法进行全局优化的部分，均与问题一类似。

问题三：根据飞行器在问题校正点处的校正成功概率，建立二叉树结构的概率模型，用来衡量飞行器成功到达终点 **B** 的概率。然后根据问题一的航迹结果，生成对应的二叉树概率模型，综合各分叉路径的概率可得到该航迹成功到达 **B** 点的飞行概率，再将此概率大小作为蚁群算法的信息素更新策略，通过迭代优化使得该成功概率不断增加，即可获得一条满足多目标的近似全局最优解。

七、模型评价与推广

7.1 模型优缺点评价

本文采用了蚁群算法来进行多目标优化。针对问题 1，我们先通过约束条件初步选择下一校正点的可行解范围；再根据优化目标，分配可行解的选择概率；最后引入蚁群算法，将每个校正点可选择的下一个目标点的概率作为启发函数，以蚂蚁经过的路径总长度分配信息素并进行迭代更新，问题 2 和问题 3 同样在蚁群算法的基础上选择不同的信息素进行迭代，通过结果最终证明了算法的有效性。

该模型的优点：

(1)采用正反馈机制，使得搜索过程不断收敛，最终逼近最优解。

(2)每个个体可通过释放信息素来改变周围环境，且每个个体能感知周围环境的实时变化，个体间通过环境实现间接通讯。

(3)搜索过程中采用分布式计算方式，多个个体同时进行并行计算，大大提高了算法的计算能力和运行效率。

(4)启发式的概率搜索方式不容易陷入局部最优，易于寻找全局最优解。

缺点：

(1)在处理较大规模的组合优化问题时，运算量较大，时间较长。

(2)由于蚁群算法中蚂蚁个体的运动过程的随机性，当群体规模设置较大时，很难在较短时间内从杂乱无章的路径中找出一条较好的路径。

(3)信息素更新策略，路径搜索策略和最优解保留策略都带有经验性，没有经过严格的理论论证。因此基本蚁群算法的求解效率不高、收敛性较差、求解结果具有较大的分散性。

7.2 模型推广

以蚁群算法为代表的群智能已经成为当今研究的一个热点，在交通、通信、化工、电力等领域成功解决了许多组合优化问题，如调度问题、指派问题、旅行商问题等。在实际应用中，限制只能飞行器飞行的条件远不止本题中所涉及的这些，这时我们可以将有些条件引入到数据处理过程中，另外一些不能在数据处理阶段被考虑的条件可以参数化，列入模型的约束条件中，得到一个比较准确的结果。

参考文献

- [1] 肖绍, 基于改进蚁群算法的无人机航迹规划研究[D]。
- [2] 肖宇, 无人飞行器航迹规划的研究[D], 沈阳理工大学, 2013。
- [3] 常友渠等, 贪心算法的探讨与研究[J], 重庆电力高等专科学校学报, 2008(3)。
- [4] Zhonghua H U, Min Z, Shihao L, 基于自适应蚁群算法的无人飞行器航迹规划[J], 计算机集成制造系统, 2012, 18(03):560-565。
- [5] ElectricPower, 蚁群算法 (ACA)及其 Matlab 实现
, <https://www.cnblogs.com/jacksin/p/9163508.html>, 2019.09.19。
- [6] CodeInfo_, 算法(六):图解贪婪算法
, <https://blog.csdn.net/a8082649/article/details/82079779>, 2019.09.19。