

多约束条件下智能飞行器航迹快速规划

牛寅

上海外国语大学贤达经济人文学院数据科学学院, 上海

收稿日期: 2025年3月11日; 录用日期: 2025年4月11日; 发布日期: 2025年4月18日

摘要

本文主要围绕智能飞行器航迹规划问题, 基于飞行器飞行过程中产生的水平误差、垂直误差、校正点位置, 校正点处的校正类型以及校正概率等约束要求, 结合题目中给出的校正点的坐标数据, 规划从起始点到终点的最优路径。本文利用不等式抽象出相关约束条件, 构建路径动态规划模型、0-1规划模型、分层规划模型, 并运用改进的Dijkstra智能算法来寻找智能飞行器的最优航行轨迹; 此外, 考虑到实际情况, 在有一定概率校正失败条件下, 权衡模型与校正点数量和航迹长短之间的关系, 让飞行器通过校正点最少, 从而使得最后寻找到的最优路径更加符合真实情况。问题的难点在于在保证误差的前提下, 使得飞行器航行路径通过校正点的次数最少且航迹最短。本文首先将飞行器飞行过程看作一个误差累积的过程, 并依据各个校正点的校正类型, 结合初始点到其余各校正点之间的距离误差准则, 筛选出初始点到各点的最优距离, 然后根据待选取校正点, 将误差约束条件与改进的图论Dijkstra算法进行融合, 从而规划出最优航迹, 使得飞行器在经过校正点次数最少同时航迹的距离也最短。在数据集一、二中, 最优航迹距离分别为106802.676米和118,730.036米。

关键词

航迹规划, Dijkstra算法, 0-1规划

Fast Planning of Intelligent Vehicle Trajectories under Multi-Constraints

Yin Niu

School of Data Science, Xianda College of Economics and Humanities, Shanghai International Studies University, Shanghai

Received: Mar. 11th, 2025; accepted: Apr. 11th, 2025; published: Apr. 18th, 2025

Abstract

This paper mainly focuses on the intelligent vehicle trajectory planning problem, based on the

constraint requirements such as horizontal error, vertical error, location of correction point, type of correction at the correction point and probability of correction generated during the flight of the vehicle, combined with the coordinate data of the correction point given in the title, to plan the optimal path from the start point to the end point. In this paper, we use inequalities to abstract the relevant constraints, construct the path dynamic planning model, 0-1 planning model, and hierarchical planning model, and apply the improved Dijkstra intelligent algorithm to find the optimal trajectory of the intelligent vehicle. In addition, taking into account the actual situation, under the condition of a certain probability of correction failure, the relationship between the model and the number of correction points and the length of the trajectory is weighed so that the aircraft passes through the least number of correction points, and thus the optimal paths found are more consistent with the real situation. The difficulty of the problem is to minimize the number of correction points and shortest trajectory under the premise of guaranteeing the error. In this paper, we first regard the flight process of the aircraft as a process of error accumulation, and according to the correction type of each correction point, combined with the distance error criterion between the initial point and the rest of the correction points, we filter out the optimal distance from the initial point to each point, and then according to the correction points to be selected, we integrate the error constraints with the improved graph theoretic Dijkstra algorithm, so as to plan the optimal trajectory, which makes the vehicle pass through correction points the least number of times and have the shortest trajectory. In data sets I and II, the optimal trajectory distances are 106802.676 meters and 118730.036 meters, respectively.

Keywords

Trajectory Planning, Dijkstra's Algorithm, 0-1 Planning

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

1.1. 问题背景

随着信息技术、自动化、计算机技术的飞速发展,现代智能飞行器技术已经发生了巨大的变化。智能飞行器起初是一种用在军事上,对敌方进行侦查、干扰以及一定必要破坏的工具。其原型类似于无人驾驶机,但智能飞行器整体更加小巧智能。智能飞行器的结构相对复杂多变,甚至最新的研究,已经向智能变体飞行器的方向发展,以适应多变的环境,提高其多约束条件下的适应性能,在未来作战中将发挥越来越重要的作用,将成为未来战场上的主要作战工具之一。智能飞行器的协同性更强,性能越来越高,但这也使得飞行器的操控与规划预定也越加复杂。飞行器的轨迹规划就是在综合考虑飞行器到达时间、油耗、角度及飞行区域等因素的前提下,以路径最短、用时最短或误差最小等为目标,为飞行器规划出最优或者是最满意的飞行轨迹,保证智能飞行器圆满地完成既定的飞行任务,是实现飞行器自动导航的一项关键技术[1]。航迹规划技术为提高作战效能,实施精确打击提供有效保障。目前,航迹规划技术,已经广泛应用于导弹制导、军用和民用的无人机的控制系统之中。但是随着现代战争进程的发展及战场情报体系的日益完善,航迹规划需要考虑的约束条件越来越多,对智能飞行器的航迹规划算法的求解效率和规划精度也提出了越来越高的要求[2]。

通常航迹规划问题是一个包含多个优化目标和多个多约束条件的非线性规划问题,其核心就是建立一个有效处理各种约束条件的目标函数的数学模型。由于所涉及的约束条件通常较多,所以常采用分层

规划的思想进行分析。首先根据已知的环境信息任务、任务要求等规划出一条满足约束条件的全局最优参考路线。然后再进行局部航迹动态优化,以提高飞行器的飞行精度,更有利于应付突发威胁,恶劣天气等意外状况,确保航迹规划的有效性和实时性。

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。飞行器的飞行物理空间是指,从它的起飞点到目标点之间的区域。在实际生活中,航迹规划的空间都是提前设定好的。这个已经设定的区域就是要进行路径搜索的空间,常称为规划空间,即为路径的搜索范围,也是进行优化的函数定义域。由于智能飞行器系统的结构限制,这类飞行器在实际飞行时,其定位系统无法对自身进行精准定位,往往会随着时间的推移,产生一定的水平误差和垂直误差。一旦定位误差累计过大,就可能导致飞行器偏离事先规划的飞行路线,使飞行器无法正确地到达目标点位置,进而导致飞行任务失败。因此,在其飞行的过程中,实时地对系统的航迹规划进行校正,消除所产生的飞行误差,保证飞行器按照规划的路线飞行,是智能飞行器航迹规划中一项重要任务。本论文研究的就是在系统定位精度限制等多约束条件下,智能飞行器的航迹快速规划问题。

1.2. 问题重述

假设智能飞行器在一个三维空间坐标系中飞行,飞行器的出发点为点 A (X_1, Y_1, Z_1),目的地为点 B (X_2, Y_2, Z_2),其航迹约束条件如下:

(1) 飞行器在空间飞行过程中需要进行实时定位,其定位误差可简单划分为垂直误差和水平误差两种。飞行器在飞行的过程中每飞行 1 m,垂直误差和水平误差将各增加 δ 个专业单位,即每段距离产生的误差大小与飞行的长度线性相关。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 θ 个单位,水平误差不大于 θ 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 θ 个单位,水平误差不大于 θ 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 200 m。

按照上述约束条件,本文主要解决如下问题。

问题 1: 针对数据集 1 和数据集 2 中的校正点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。

其中数据集 1 数据的参数为:

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

数据集 2 中数据的参数为:

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

分别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。

2. 模型假设与符号说明

2.1. 模型假设

为了方便问题的研究,本文对题目中的某些条件进行如下的简化及合理的假设:

假设 1: 若飞行器的水平误差和垂直误差在校正点处满足校正要求且能得到及时校正,飞行器按照预定的航线飞行。

假设 2: 在终点处,当垂直误差和水平误差均小于 θ 个单位时,飞行器仍能按照规划路径到达终点。

假设 3: 在出发点处(即 A 点),飞行器的垂直和水平误差均为 0。

假设 4: 问题 1 中,在校正点进行校正过程的中,由于飞行器校正的时间较短,为简化问题,校正所产生的位移忽略不计,即校正的过程实现即时转弯。

假设 5: 校正点的位置和校正类型,在智能飞行器出发前已知,并且不会随着时间而发生改变。

假设 6: 问题 2 中,由于结构和控制系统限制,无法完成即时转弯。并假设最小转弯半径为 200 m。

假设 7: 问题 3 中飞行器到达校正点时即知道在该校正点处能否校正成功。

假设 8: 为简化问题,本文未考虑空间障碍物和禁飞区域等问题。

假设 9: 不考虑智能无人机飞行过程中,燃料的消耗对飞行状况产生的问题。

假设 10: 不考虑雷电,暴雨等突发天气情况对无人机飞行状况的影响。

2.2. 符号说明

符号	符号解释
θ	按规划路径飞行允许的误差范围
δ	垂直误差和水平误差
α_1	垂直误差校正点的垂直误差限定
α_2	垂直误差校正点的水平误差限定
β_1	水平误差校正点的垂直误差限定
β_2	水平误差校正点的水平误差限定
d_{ij}	校正点 i, j 间的连通欧氏距离矩阵
ω_{ij}	校正点 i, j 间欧氏距离转换的邻接矩阵

3. 问题解答

3.1. 问题分析

相比常规的无人驾驶机和原始的载人侦察机,智能飞行器整体积更加小巧轻便,更加智能。不仅可以实现实时监测,精准打击等基本功能,还可以在飞行过程中自动校正,减少人为操作,协同性更强,飞行性能更好,甚至在向智能变体性飞行器的方向发展。在未来作战中将发挥越来越重要的作用,将成

为未来战场上的主要作战工具之一。航迹规划是指在综合考虑智能飞行器到达目的地所使用的时间、速度、油耗及飞行区域等约束条件下,以路径最短、校正调整最少或到达目的地的误差最小等优化条件下,规划出一条最符合智能飞行器要求的航迹路线,以此提高智能飞行器的作战效能,快速精准地到达设定的目的地,实施监视侦查、战略干扰、打击破坏等战略要求,保证智能飞行器圆满地完成既定的战略飞行任务。

问题 1 的实质是研究在多约束条件下的智能飞行器轨迹规划的优化问题。在满足约束条件(1)~(7)的前提下,建立目标函数,其目的是使得智能飞行器在顺利到达目标点的前提下航迹尽可能最短,并且在飞行过程中,经过校正区域进行校正的次数尽可能少。在此过程的基础上,采用分层规划的思想,通过将约束条件进行融合设计优化算法,分别利用数据集 1 和数据集 2 中的数据进行问题求解,即找出在相应数据集下,智能飞行器的最佳规划路径,与此同时本文讨论了优化模型的有效性及其计算复杂度。

该航迹规划的目的是使得飞行器在航迹尽可能短的同时,经过校正点的次数尽可能少,故此为一种动态规划问题。本文共给出两个固定数据集:数据集 1 和数据集 2。数据集 1 中包含 613 个节点(1 个出发点、1 个目标点和 611 个校正区域)及相应位置坐标。数据集 2 中包含 327 个节点(1 个出发点、1 个目标点和 325 个校正区域)及相应位置坐标。所以,根据各数据集所给的点位置坐标,空间内的任意两点之间的距离,都可以通过欧氏距离公式来进行计算。这就将飞行区域的空间三维问题,转化为了求解最小路径的平面二维带有约束的优化问题。

此外,本题中的飞行器校正点又分为水平校正点和垂直校正点,分别对飞行器进行水平校正处理和垂直校正处理。而对于不同类型的校正点,智能飞行器需要进入该校正点进行校正,由于飞行器存在实时状态误差,因此需要在不同的校正点上建立不同的误差约束。此外,本文还利用一定的限制条件进行权重处理,将距离矩阵转换为邻接矩阵,来表示各点之间的关系,在相应的限制条件的基础上,构建了问题的代价函数,并利用图论 Dijkstra 智能最短路径算法对上述带有约束的目标函数进行求解。在此期间,本文还对 Dijkstra 算法进行改进,将约束条件融合在算法之中,从而使得求出的最优航迹规划更加符合实际情况。最后根据算法迭代过程,分析并讨论了改进算法的有效性和时间计算复杂度。

3.2. 模型建立

3.2.1. 飞行器的航迹表示

通常情况下,飞行器的航迹标识号方法有两种[3]:一种是基于飞行器运动学、动力学描述的连续平滑轨迹;另一种是用航迹点表示,相邻航迹点用直线段连接几何折线航迹。相比第一种方法,第二种方法可以将复杂的航迹规划问题,分解为一组同一形式的小规模子问题,对于每个问题只关心一个点的坐标,从而将考察航迹是否可行转变为考虑一个点和一个线段是否满足要求的问题,可以有效地将复杂的实际问题进行合理简化。本题使用第二种表示方法对飞行器的航迹进行描述,便于合理有效地对航迹的动态规划问题进行分析处理。

3.2.2. 欧式距离(Euclidean Metric)

在实际问题中,欧几里得距离(也称欧式距离)是一种通常采用的距离定义。用来描述一个 M 维空间中向量的自然长度,或两点之间的真实距离。在 $M = 2$ 和 $M = 3$ 时,欧氏距离就是表示二维和三维条件下,两点之间的真实距离,相应的公式如下:

二维空间中的欧氏距离:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

三维空间中的欧氏距离:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (2)$$

3.2.3. 0-1 规划

0-1 规划是决策变量取值仅为 0 或 1 的一类特殊的整体规划。在处理经济管理和某些规划问题中, 广泛应用。该方法可将问题中的决策变量, 转换为 0-1 变量(二级制变量), 即决策变量, 可以数量化地描述诸多如留与弃, 开与关等反映逻辑关系、顺序关系以及互斥关系的约束条件。

本文中, 为简化问题的复杂度, 对于校正点的类型问题进行 0-1 规划, 具体定义如公式(3)所示:

$$q1 \begin{cases} 1, \text{该点为垂直校正点} \\ 0, \text{该点为水平校正点} \end{cases} \quad (3)$$

3.2.4. 数据处理

1) 为了更好地便于后面利用算法对问题的处理, 在此本文将校正点的校正状态进行数字化表示:

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
02	表示水平误差校正不成功

2) 坐标数据处理: 附件中给出的是各节点的坐标数据, 将坐标数据转化为连通轨迹的距离数据:

$$d_{ij} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (4)$$

其中 i, j 分别表示不同的校正节点。

3) 空间距离邻接矩阵是表示空间散点之间相邻距离关系的矩阵, 本题中的校正点具有空间特性, 而空间中, 点与点之间距离最短, 所以对附件中的校正点坐标建立邻接矩阵 D 。

以附件 1 为例构建空间距离邻接矩阵。空间区域共有 613 个坐标点, 包括起始点 A、终点 B 和 611 个校正点, 用 W 表示邻接矩阵, 权值 w 由相应两点 i 和 j 的欧式距离给出, 具体表达如公式(5)所示:

$$w_{ij} = \begin{cases} d_{ij}; & \text{当节点 } i \text{ 到节点 } j \text{ 之间满足最大校正误差距离} \\ \infty; & \text{当节点 } i \text{ 到节点 } j \text{ 之间不满足最大校正误差距离} \end{cases} \quad (5)$$

3.2.5. 常用的航迹规划算法

A*搜寻算法[4]俗称 A 星算法, 是一种启发式图搜索算法, 是一类求出最低通过成本的算法。该算法广泛地用于路径优化领域。它的特点是检查最短路径的过程中, 对每个可能的节点同时引入了全局信息, 可以在全局有限的条件中, 对当前节点距终点的距离做出合理的估计, 作为评价该节点处于最短路线上的可能性判断度量, 并保证全局最优解的收敛性, 在处理航路规划类问题时, 可以较好地满足各类限制条件。具有可采纳性、单调性、信息型的特点。

A*算法的核心思想是建立一个启发函数, 如公式(6)所示:

$$f(n) = g(n) + h(n) \quad (6)$$

式中: $g(n)$ 是从起始点到当前节点 n 的实际代价值, $h(n)$ 是从当前节点 n 到目标点的估计值。两者相加得到的就是当前节点的总估计值 $f(n)$, 然后再对 $f(n)$ 的大小做出比较, 选取 $f(n)$ 最小的节点 n 作为

有效节点。

A*算法的实现步骤如下：

步骤 1：把起始点作为第一个路径节点，将周围与之相邻的节点作为候选节点，并储存。

步骤 2：分别计算步骤 1 中的候选点对应的启发函数值，再找出启发函数值最小时所对应的节点，把该点作为有效点，并记录该点。

步骤 3：判断步骤 2 中得到的有效点是否满足终止条件，若满足，结束算法；若不满足，则跳转回步骤 2，将下一点作为新的起始点继续搜索，直到满足条件为止。

步骤 4：将所有有效点进行连线，获得最优路径。

虽然 A*算法是比较常用的一种航迹规划算法，但在每一次扩展时，均需对范围内所有节点进行判断，将所有满足要求的节点加入到扩展列表中，循环处理。这就导致了算法计算量过大，大大降低了规划路径的效率，无法满足本题中的快速规划要求，降低算法的时效性。其次，A*算法每一步的最佳搜索方向并不是朝着目标点的方向的，因此得到的路径通常不是最优，或者规划出的路径有一定的曲折，并非最短。所以判断该方法并不适用本文要求，需考虑其它算法处理本题。

3.2.6. 改进的最短路径算法

经典 Dijkstra 算法

最短路径分析是网络分析最基本的功能之一。迪杰斯特拉算法(Dijkstra)算法是荷兰计算机科学家 Dijkstra 提出的一种最短路径算法。迪杰斯特拉算法的主要特点是以起始点为中心向外层层扩展，直到扩展到终点为止。常用于搜索从某一固定起点到其余各点的最短路径，适合解决静态道路规划问题，是目前公认的较好的最短路径算法。最短路径不仅仅指一般意义上的距离最短，诸如时间、费用都可被引申为最短路径。相应地，最短路径问题就成为最快路径问题、最低费用问题等[5]。

经典 Dijkstra 算法将网络结点分成 3 部分：未标记结点、临时标记结点和永久标记结点。网络中所有结点首先初始化为未标记结点，在搜索过程中和最短路径中的结点相连通的结点为临时标记结点。每次循环都是从临时标记结点中搜索距源点路径长度最短的结点作为永久标记结点，直至找到目标结点或者所有的结点都成为永久标记结点来结束算法。

算法的基本思想基于以下原理：两结点间的最短路径包含了其内部其它的最短路径，而算法实现的主要技术是松弛技术，而这种技术的实质是反复减小每个结点实际路径权值的上限，直到该上限等于最短路径的权值[6]为止。即若序列 $\{v_0, v_1, \dots, v_{n-1}, v_n\}$ 是 v_0 到 v_n 的最短路线，则序列 $\{v_0, v_1, \dots, v_{n-1}\}$ 必为 v_0 到 v_{n-1} 的最短路线。

定义两种标号：临时标号 V 和永久标号 U。若给 v_i 点一个 U 标号，表示从 v_0 到 v_i 点的最短路径， v_i 的标号不再改变。若给 v_i 点一个 V 标号，表示从 v_0 到 v_i 点的估计最短路径的上界，是一种临时标号，凡没有得到 U 标号的点，在终点 v_{n-1} 得到 U 标号之前，保持计算。

计算步骤如下：

步骤 1：给起始点 v_0 以标号 U，令 $U(0)=\{v_0\}$ ，余下的点全部标号 V，对于 $i \neq 0$ ，则给定 $l(v_i)=+\infty$ 。

步骤 2：如果 v_i 为刚刚获得 U 标号的点，则考虑 V 标号的点 v_j ，对于 v_j 的 V 标号进行更改： $l(v_j)=\min[l(v_i), l(v_i)+d_{ij}]$ 。

步骤 3：比较所有具有 V 标号的点 v_j ，把最小者 V 标号的点改为 P 标号，即 $\min[V(v_j) \cup U_i] = U_{i+1}$ 。

步骤 4：若终点得到 U 标号，则计算结束，所有获得的 U 标记点即为所算出的最短路径。否则代替 v_i 转回步骤 2。

计算步骤如下:

步骤 1: 初始化 $\text{label}()$ 为 inf 。给起始点 v_0 以标号 U , 令 $U = \{v_0\}$, 余下的点全部标号 V 。以二分法思想, 在考虑校正点类型的情况下, 计算初始点到各顶点的距离, 并计算飞行误差是否满足对应校正点对误差的约束条件。 $\text{label}(v)$ 为起始点到点 v 的距离。在满足的飞行误差前提下更新 $\text{label}(v)$ 。

步骤 2: 选取最小的 $\text{label}(v)$, 将当前的 V 标号替换给 U 标号。

步骤 3: 如果 v_i 为刚刚获得 U 标号的点, 则考虑 V 标号的点, 且 d_{ij} 是否满足距离限定要求, 对于 v_i 的 V 的标号进行更改: $l(v_j) = \min[l(v_i), l(v_i) + d_{ij}]$ 。

步骤 4: 对 v_j 进行误差校正, 若超出校正误差, 则该点不选用, 重复步骤 2。

步骤 5: 比较所有具有 V 标号的点 v_j , 把最小者 V 标号的点改为 U 标号, 即 $\min[V(v_j) \cup U_i] = U_{i+1}$ 。

步骤 6: 若终点得到 U 标号, 则计算结束, 所有获得的 U 标记点即为所算出的最短路径。否则代替 v_i 转回步骤 3。

改进后 Dijkstra 算法的流程图如图 1 所示。

3.2.7. 代价函数(Cost Function)

代价函数(又称损失函数)[7][8], 是将随机事件或其有关随机变量的取值映射为非负实数以表示该随机事件的“风险”或“损失”的函数。在应用中, 损失函数通常作为学习准则与优化问题相联系, 即通过最小损失函数求解和评估模型, 通常处理分类问题和回归问题等, 是最优控制领域中一个基础的判定方式。

本题的多约束条件下的航迹规划问题实质是一个动态规划的问题, 目的是在多约束条件下, 寻找出符合要求的最优路径。因此, 本文引入代价函数, 对优化效果综合地进行定量评定分析。并通过代价函数的引用, 将多个约束条件进行归一处理, 简化了问题的复杂性, 并有利于改进模型。代价函数的表达式如公式(7)所示:

$$f = \sum_{i=1}^n (\alpha_{1i} l_i + \alpha_{2i} k_i + \alpha_{3i} r_i) \quad (7)$$

其中, α_{ji} 代表不同点处, 几个约束条件的权重参数, 可通过人为给定;

l_i 代表路径上第 i 个点的路径长度;

k_i 代表经过第 i 点时, 所经历的校正次数;

r_i 代表第 i 个点处, 进行校正的误差限制条件。

3.3. 模型的求解与分析

根据前面所述建立的模型与搭构算法框架, 在此利用题目中给出的真实数据进行模型求解, 具体结果如下所示:

3.3.1. 数据集 1

处理数据集 1 时, 利用题中所给的参数即: 垂直校正点、水平校正点以及飞行器能够按规划轨迹正常飞行的参数, 如下所示:

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

经过利用上文所提的改进的 Dijkstra 算法处理后, 可得到了满足题目要求的最优规划路径, 以及在校正点处的智能飞行器校正状态。数据集 1 的航迹规划结果表如表 1 所示。在该航迹中, 飞行器共经过 11 个校正点, 飞行路径总长度约为 106,802.676 米, 路径的三维效果图如图 2 所示。

Table 1. Table of trajectory planning results for dataset 1
表 1. 数据集 1 的航迹规划结果表

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	0	出发点 A
503	13.38791985	13.38791985	1
69	8.807342267	22.19526212	0
506	16.67556984	7.868227575	1
28	5.995366388	13.86359396	0
183	17.32335521	11.32798882	1
193	4.689211536	16.01720036	0
288	15.48671202	10.79750049	1
122	4.854493815	15.65199430	0
113	12.91477575	8.060281938	0
485	20.19511573	7.280339975	1
248	4.220562731	11.50090271	0
612	23.73401365	19.51345092	终止点 B

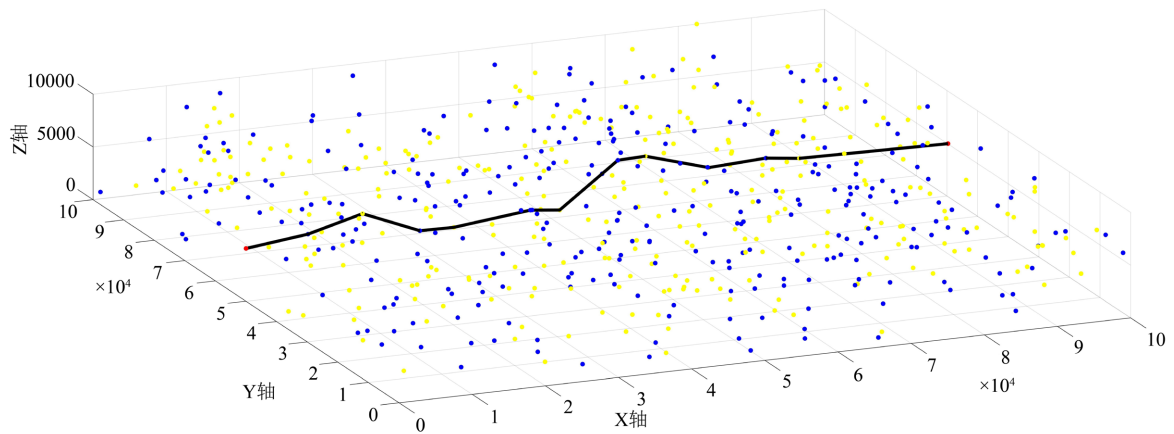


Figure 2. Shortest path of the trajectory for dataset 1
图 2. 数据集 1 的航迹最短路径

该部分获取最优路径的坐标如表 2 所示。

3.3.2. 数据集 2

处理数据集 2 时，垂直校正点、水平校正点以及飞行器能够按规划轨迹正常飞行的参数要求如下：

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

经过我们航迹规划算法处理后，我们得到了满足题目要求的最优规划路径。最优路径的相关参数如表 3 所示。共经过 13 个校正点，路径长度为约 118730.0358 米。路径的三维效果图如图 3 所示。

Table 2. Coordinates of the optimal path
表 2. 最优路径的坐标

校正点编号	X 轴坐标	Y 轴坐标	Z 轴坐标
0	0	50000	5000
503	11392.96074	56973.01824	4097.858018
69	19835.93612	59347.29036	4903.014153
506	27388.62945	58657.65473	2807.726248
28	32952.07895	60824.8649	2263.991244
183	44139.68569	62558.04978	2660.04247
193	48636.48807	63754.13391	2079.649575
288	58196.77435	67621.58034	5278.055614
122	62703.29285	69398.64706	4962.722351
113	69920.50138	65952.91537	3959.23888
485	76763.63882	63601.13263	4761.696421
248	80615.35145	61875.60118	4762.594376
612	100000	59652.34338	5022.001164

Table 3. Table of trajectory planning results for dataset 2
表 3. 数据集 2 的航迹规划结果表

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	0	出发点 A
150	10.87508045	10.87508045	0
238	19.82100418	10.94592372	1
234	2.347539566	13.29346329	0
309	15.65917311	13.31163354	1
305	5.968714547	19.28034809	0
123	15.17310764	9.204393096	1
231	9.436726707	18.6411198	0
160	18.15390257	8.717175866	1
191	2.542440309	11.25961617	0
88	15.35486204	12.81242174	1
50	2.326571646	15.13899338	0
96	13.76445924	11.4378876	1
61	6.481987198	17.9198748	0
326	18.80352704	12.32153984	终止点 B

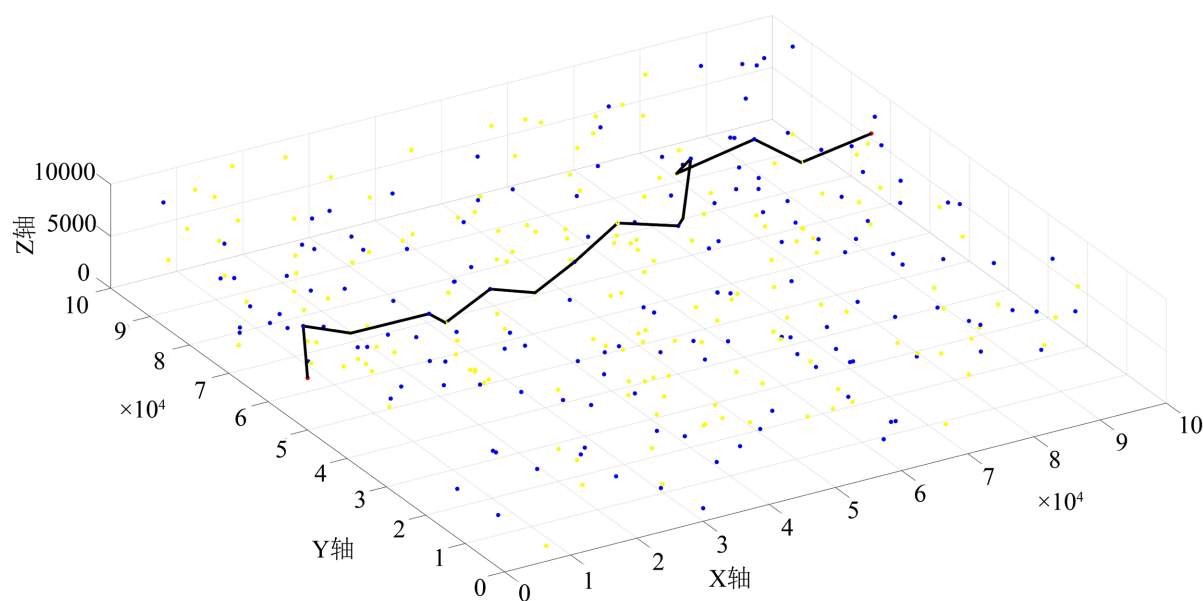


Figure 3. Shortest path of the trajectory for dataset 2

图 3. 数据集 2 的航迹最短路径

4. 结论

根据算法复杂性理论,评价一个算法质量的优劣,主要从算法的时间复杂度或空间复杂性来考虑[9]。算法的时间复杂度是指,执行该算法所消耗的时间代价。在本文所提出的算法中,借鉴经典的 Dijkstra 最短路径算法的思想,在此基础上进行了改进,来解决多约束条件下的航迹规划问题。并通过对数据集内,各校正点的预处理,计算各点之间的欧氏距离。根据题中限定的“飞行器每飞行 1 m,垂直误差和水平误差将各增加个专用单位”,以及各校正点完成校正的误差限制条件,优先对邻接矩阵进行筛选处理,大大减少了备选参考点。通过这样处理大大减少了算法迭代次数以及计算时间。并通过权重设定代价函数,在算法中进一步筛选划分,缩小航迹规划中的预选范围。从而有效地提高了算法的计算效率,减少了算法的计算复杂度。同时,相比于传统的 A 星算法,使用本文所提出的改进算法,进行航迹规划,可以在更短的时间内,得到航迹长度更短,经理校正点的次数更少计算效果,更符合题所设定的优化目标,有效地提高了优化效果。进一步证明了本文提出的算法模型具有更低的复杂度和更有效的优化能力。

参考文献

- [1] 姜玲玲, 陈怀民, 贾宝娟. 三维分层航迹规划算法研究与仿真[J]. 测控技术, 2011, 30(3): 110-113.
- [2] 冯林, 贾菁辉. 基于对比优化的 RRT 路径规划改进算法[J]. 计算机工程与应用, 2011, 47(3): 210-213.
- [3] 王琼, 刘美万, 任伟建. 无人机航迹规划常用算法综述[J]. 吉林大学学报(信息科学版), 2019, 37(1): 61-70.
- [4] Kala, R., Shukla, A. and Tiwari, R. (2010) Fusion of Probabilistic A* Algorithm and Fuzzy Inference System for Robotic Path Planning. *Artificial Intelligence Review*, **33**, 307-327. <https://doi.org/10.1007/s10462-010-9157-y>
- [5] 李元臣, 刘维群. 基于 Dijkstra 算法的网络最短路径分析[J]. 微计算机应用, 2004, 25(3): 295-298, 362.
- [6] 王树禾. 图论及其算法[M]. 合肥: 中国科学技术大学出版社, 1990: 16-20.
- [7] 李航. 统计学习方法[M]. 北京: 北京大学出版社, 2012: 7-10.
- [8] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016: 121-139, 298-300.
- [9] 李士勇. 优化算法原理与应用[M]. 哈尔滨: 哈尔滨工业大学出版社, 2012: 83-87.