DOI:10.19651/j.cnki.emt.2208918

基于 Dijkstra 算法改进的飞行器航迹快速规划算法*

郑 弈 谢亚琴
(南京信息工程大学 南京 210044)

摘 要: 当飞行器在航行途中遇到突发情况需要临时更改路径时,这就对航迹规划算法的效率和可靠性提出了很高的要求。针对这一问题,提出了一种加入预搜索的 Dijkstra 算法改进方案。该算法使用归一化熵权法建立了较为客观的航迹评价函数,简化了多目标航迹优化模型。通过加入深度为一的预搜索过程实现 D 算法的回溯功能,解决了经典 D 算法因松弛性不足,在复杂约束条件下路径搜索失败率高的问题。此外,为了进一步减少运算时间,在预搜索遍历过程中加入跳出机制。算法仿真结果表明,本文所提算法的运行时间相较于普通回溯 D 算法减少了 46%,且在复杂约束条件下的航迹搜索成功率与航迹质量均接近智能算法,能够满足复杂条件下快速航迹规划的需求。

关键词: 多目标优化;熵权法;Dijkstra;预搜索;回溯算法
中图分类号: TN967.5 **文献标识码:** A **国家标准学科分类代码:** 510.8050

Improved fast aircraft path planning algorithm based on Dijkstra algorithm

Zheng Yi Xie Yaqin
(Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract: When the aircraft needs to change the path temporarily in case of emergencies during navigation, the efficiency and reliability of the route planning algorithm are urgently required. An improved Dijkstra algorithm with pre search is proposed to solve this problem. A more objective track evaluation function is established by using the normalized entropy weight method to simplify the multi-objective track optimization model. The backtracking function of D algorithm is realized by adding the pre search process with depth of one, which solves the problem of high failure rate of path search under complex constraints due to insufficient relaxation of classical D algorithm. In addition, a break mechanism is added in the pre search traversal process to further reduce the operation time. Simulation results show that the operation time of the proposed algorithm is reduced by 46% compared with ordinary backtracking D algorithm. And the algorithm is close to intelligent algorithm in path search success rate and the accuracy of the shortest path decision under complex constraints, which can meet the requirements of fast path planning under complex conditions.

Keywords: multi-objective optimization; entropy weight method; dijkstra; pre search; backtracking algorithm

0 引 言

如今,智能飞行器已经在物资配送,灾害救援,军事侦察等领域得到广泛使用。复杂条件下的快速航迹规划是智能飞行器的热门研究之一,同时也是无人驾驶系统的关键所在。无人机在飞行过程中自身定位误差会积累,需要途经一些校正点进行及时的误差校正。因此,飞行器需在复杂工作环境下找到一条同时满足路径短,效能高(途经节点少)且可靠性强的航迹。该问题可以建模为一个多目标优化问题。

多目标优化有多种解决思路[1]。最常用的是将多目标简单线性加权求和转化为单目标的方法[2]。但是该方法存在数量级不统一,加权系数选取不合理的问题[3]。一旦目标的单位发生变化,模型就会失效。分层序列法非常适用于规划目标较多的场景[4],也能避免各目标数量级不统一的问题,但要求各目标间有明显优先级关系,适用场景有限。

目前路径规划算法主要有经典法和智能算法两大类。经典法包括早期的 Dijkstra 算法[5](本文中后续出现时简写为 D 算法)、Floyd 算法[6]和 A* 算法[7]等。这些方法面对小规模简单限制条件规划时具有目标性好、速度快等优点。但是经典算法有应用场景的局限性,比如在复杂约束条件下经典 D 算法会因为其贪心算法本质而陷入死循环

收稿日期:2022-01-24
* 基金项目:国家自然科学基金(62001238,62105159)项目资助

导致搜索失败。而且当数据规模增大，约束条件变复杂时经典算法复杂度往往呈指数级增长，这大大限制了其应用。

智能算法的典型代表有遗传算法[8]、蚁群算法[9]等。群体智能算法有较大概率跳出局部最优，且复杂度在可以接受的范围内，如今在航迹规划上有广泛应用。但是由于初始解生成方式以及更新方式的随机性，其收敛速度较慢，需要付出较大的时间成本。为此，文献[10]和[11]分别提出了遗传算法和蚁群算法的改进方案，相比传统智能算法在收敛速度以及路径搜索质量上都有一定提升。

但因为天气状况等突发因素存在，飞行器往往需要在航行途中临时更换路线。这对算法执行效率提出了非常高的要求，而智能算法需要多轮迭代收敛，在快速航迹规划应用中有“先天不足”，因而无法满足这一需求。

为了解决上述问题，本文首先使用归一化加权法将多目标优化转化为单目标优化问题。其次，利用 D 算法目标性强、全局性好的特点，在经典 D 算法基础上加入预搜索过程实现算法的回溯，大大提高算法松弛度。此外，本文在预搜索过程中增加跳出机制，进一步减少算法运行时间。经过 MATLAB 仿真验证了改进的 D 算法在复杂约束条件下的有效性。

1　无人机航迹规划模型

1.1　模型建立

智能无人飞行器正在执行物资运输任务，当到达节点 S 时，由于突发的大气状况，无法按照预先规划路径继续飞行。这时，飞行器需要在行进过程中快速规划出一条能够顺利到达终点 E 的路径。三维飞行区域图如图 1 所示。

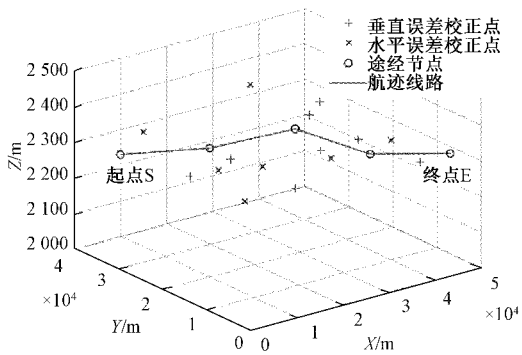

图 1　航迹规划区域示意图

在实际飞行过程中，无人飞行器自身定位的水平方向误差和垂直方向误差会随着航行距离的增加而累积。因此，为了保证飞行器能够顺利到达终点，在飞行过程中需要通过一些校验点来对飞行器的实时位置进行校正，从而避免由于误差积累导致其偏离航线。校正点包括垂直误差校正点和水平误差校正点两种，分别如图 1 所示。

针对上述问题，无人机航迹规划的目标是在累积误差小于一定数值的前提下，找到使得飞行器的航迹距离最短且途经的校准点数也尽量少的飞行路径。表 1 中总结了飞行器航迹规划模型中出现的主要符号。文中⊙表示哈达玛乘积。

表 1　主要符号说明

符号	意义	大小
num	途经校正点数	—
L	航迹距离	—
δ	飞行过程中误差距离比	—
θ	飞行器正常飞行的误差阈值	—
α_1,α_2	垂直误差校正时的垂直和水平误差阈值	—
β_1,β_2	水平误差校正时的垂直和水平误差阈值	—
$\boldsymbol{G}$	包括起点终点的全部节点集	n
$\boldsymbol{G}_p$	已标记节点集	可变
$\boldsymbol{W}$	有向图的权值矩阵	$n\times n$
$\boldsymbol{A}$	有向图的邻接矩阵	$n\times n$
$\boldsymbol{T}$	实际飞行路径的距离矩阵	$n\times n$
$\boldsymbol{t}$	矩阵 $\boldsymbol{T}$ 的各行元素和	$n\times 1$
$\boldsymbol{ve}$	各节点处飞行器的垂直误差	$n\times 1$
$\boldsymbol{he}$	各节点处飞行器的水平误差	$n\times 1$
$\boldsymbol{Q}$	飞行器误差矩阵	$n\times 2$
$\boldsymbol{P}$	飞行器飞行距离矩阵	$n\times 2$
$\boldsymbol{\theta}$	正常飞行的误差阈值矩阵	$n\times 2$
$\boldsymbol{\alpha}$	垂直误差校正的阈值矩阵	$n\times 2$
$\boldsymbol{\beta}$	水平误差校正的阈值矩阵	$n\times 2$
$\boldsymbol{v}$	垂直误差校正列向量	$n\times 1$
$\boldsymbol{h}$	水平误差校正列向量	$n\times 1$
$\boldsymbol{H}_v$	垂直误差校正矩阵	$n\times 2$
$\boldsymbol{H}_h$	水平误差校正矩阵	$n\times 2$
$\boldsymbol{H}$	垂直水平误差校正矩阵	$n\times 2$

首先，将起点 S、终点 E 和各校正点 P_i 共 n 个元素记录在节点集 $\boldsymbol{G}$ 中，$\boldsymbol{G}=\{S,P_1,P_2,\cdots,P_{n-2},E\}$。随后，读取集合 $\boldsymbol{G}$ 中各点的坐标以及校正点类型，由这 n 个节点构成一个有向图，其权值矩阵用 $\boldsymbol{W}$ 表示，$\boldsymbol{W}\in n\times n$。

为了方便目标函数的表示，本文引入 0～1 规划的思想，用邻接矩阵 $\boldsymbol{A}$ 记录选择的路线，$\boldsymbol{A}\in n\times n$。不妨把前一个途经节点记做 p_i，后一个途经节点记做 p_j，下标为节点编号，则当 p_i 和 p_j 相连时该有向图的邻接矩阵中元素 A_{ij} 为 1，否则为 0。这样可以得到飞行器途经的校正点数 num 满足：

$$num = sum(\boldsymbol{A}) \tag{1}$$

而航迹距离 L 可以由下式获得：

$$L = sum(\boldsymbol{A}\odot\boldsymbol{W}) \tag{2}$$

飞行器每飞行 1 m，垂直误差 VE 和水平定位误差 HE 都增加 δ 个单位，且为了保证飞行器能够顺利到达终点，在飞行过程中任何时刻飞行器的水平误差和垂直误差都必须小于 θ 个单位。也就是说航迹线路中到达各中转点时的误

差积累值不能超过 θ 个单位。将到达各节点时的垂直误差记录在列向量 $\boldsymbol{ve}$ 中，$\boldsymbol{ve} \in n \times 1$。同理将水平误差记录在列向量 $\boldsymbol{he}$ 中，$\boldsymbol{he} \in n \times 1$。令实际航迹权值矩阵 $\boldsymbol{T} = \boldsymbol{A} \odot \boldsymbol{W}$，其中，$\boldsymbol{T} \in n \times n$。令矩阵 $\boldsymbol{T}$ 的各行元素之和构成的列向量为 $\boldsymbol{t}$，则飞行器在飞行过程中必须满足式(3)的约束条件：

$$\boldsymbol{Q} + \delta \boldsymbol{P} < \boldsymbol{\theta} \tag{3}$$

其中，

$\boldsymbol{Q} = [\boldsymbol{ve}, \boldsymbol{he}]$，$\boldsymbol{P} = \boldsymbol{t}\boldsymbol{1}_{1\times 2}$，$\boldsymbol{\theta} = \theta \boldsymbol{1}_{n\times 2}$

同时，规定飞行器进行垂直误差校正时垂直误差阈值为 α_1，水平误差阈值为 α_2。用垂直误差校正列向量 $\boldsymbol{v}$，$\boldsymbol{v} \in n \times 1$ 记录节点类型，属于垂直误差的校正点取 1，否则取 0。可以得到式(4)的约束条件：

$$\boldsymbol{Q} + \delta \boldsymbol{P}_v < \boldsymbol{\alpha} \tag{4}$$

其中，

$\boldsymbol{t}_v = \boldsymbol{T}\boldsymbol{v}$，$\boldsymbol{P}_v = \boldsymbol{t}_v \boldsymbol{1}_{1\times 2}$，$\boldsymbol{\alpha} = \boldsymbol{1}_{n\times 1}[\alpha_1, \alpha_2]$

垂直方向误差得到校正后得到邻接矩阵 $\boldsymbol{A}$ 和飞行器误差矩阵 $\boldsymbol{Q}$ 满足以下等式：

$$\boldsymbol{A}^{-1}(\boldsymbol{Q} + \delta \boldsymbol{P}) \odot \boldsymbol{H}_v = \boldsymbol{Q} \tag{5}$$

其中，

$\boldsymbol{H}_v = [\boldsymbol{1}_{n\times 1} - \boldsymbol{v}, \boldsymbol{1}_{n\times 1}]$

同理，规定飞行器进行水平误差校正时垂直误差阈值为 β_1，水平误差阈值为 β_2。用垂直误差校正列向量 $\boldsymbol{h}$，$\boldsymbol{h} \in n \times 1$ 记录节点类型，属于水平校正点取 1，否则取 0。可以得到约束条件为：

$$\boldsymbol{Q} + \delta \boldsymbol{P}_h < \boldsymbol{\beta} \tag{6}$$

其中，

$\boldsymbol{t}_h = \boldsymbol{T}\boldsymbol{h}$，$\boldsymbol{P}_h = \boldsymbol{t}_h \boldsymbol{1}_{1\times 2}$，$\boldsymbol{\beta} = \boldsymbol{1}_{n\times 1}[\beta_1, \beta_2]$

水平方向误差得到校正后得到邻接矩阵 $\boldsymbol{A}$ 和飞行器误差矩阵 $\boldsymbol{Q}$ 满足以下等式：

$$\boldsymbol{A}^{-1}(\boldsymbol{Q} + \delta \boldsymbol{P}) \odot \boldsymbol{H}_h = \boldsymbol{Q} \tag{7}$$

其中，

$\boldsymbol{H}_h = [\boldsymbol{1}_{n\times 1}, \boldsymbol{1}_{n\times 1} - \boldsymbol{h}]$

合并式(5)和(7)可以得到式(8)：

$$\boldsymbol{A}^{-1}(\boldsymbol{Q} + \delta \boldsymbol{P}) \odot \boldsymbol{H} = \boldsymbol{Q} \tag{8}$$

其中，

$\boldsymbol{H} = [\boldsymbol{1}_{n\times 1} - \boldsymbol{v}, \boldsymbol{1}_{n\times 1} - \boldsymbol{h}]$

综合式(1)～(8)可以得到，飞行器的航迹规划问题可以建模为式(9)所示的多目标优化问题：

$$\text{s.t.} \begin{cases} \min\limits_{\boldsymbol{A}} num \\ \min\limits_{\boldsymbol{A}} L \\ \boldsymbol{Q} + \delta \boldsymbol{P} < \boldsymbol{\theta} \\ \boldsymbol{Q} + \delta \boldsymbol{P}_v < \boldsymbol{\alpha} \\ \boldsymbol{Q} + \delta \boldsymbol{P}_h < \boldsymbol{\beta} \\ \boldsymbol{A}^{-1}(\boldsymbol{Q} + \delta \boldsymbol{P}) \odot \boldsymbol{H} = \boldsymbol{Q} \end{cases} \tag{9}$$

1.2　模型预处理

为了简化式(9)所示的优化问题的计算，本文使用归一化熵权法[12]将双目标优化问题转化为单目标优化问题，同时避免了各目标数量级之间不统一的问题。

但在使用 D 算法进行路径搜索之前，路径长度以及途经节点数的最大值和最小值是未知的，故无法直接进行归一化处理。为此，本文首先使用禁忌搜索算法[13]初步搜索得到 m 条可行路径的长度和途经节点数组成部分可行解集 $\{L_1, num_1; L_2, num_2; \cdots; L_m, num_m\}$，其中路径长度最大值记为 $L_{\max}$，途经点数最大值记为 $num_{\max}$。但部分可行解集中未必有实际的最短航迹距离和最少途经中转点数。不妨将起点 S 和终点 E 之间的直线距离 L_{SE} 作为最短距离，即：

$$L_{\min} = L_{SE} \tag{10}$$

根据误差限制条件可以得到相邻两节点间的最大平均距离为 $\theta/2\delta$，从而可以得到最少节点数 $num_{\min}$，即：

$$num_{\min} = 2L_{SE}\delta/\theta \tag{11}$$

根据可行解集对 L 和 num 进行归一化处理，得到归一化后的 L^* 和 num^* 表达式为：

$$L^* = \frac{L - L_{\min}}{L_{\max} - L_{\min}} \tag{12}$$

$$num^* = \frac{num - num_{\min}}{num_{\max} - num_{\min}} \tag{13}$$

由于路径搜索结果对权值选取的敏感度很高，因此，本文提出使用熵权法对双目标赋权值。熵权法的基本思想是根据各指标的变异性大小确定其在评价模型中的客观权重[14]。首先对部分可行解集进行归一化处理，得到 $\{L_1^*, num_1^*; L_2^*, num_2^*; \cdots; L_m^*, num_m^*\}$。然后得到航迹距离熵 e_L 和途经点数熵 e_{num} 分别为：

$$e_L = -\frac{1}{\ln m}\sum_{i=1}^{m} p_{L_i} \ln p_{L_i} \tag{14}$$

$$e_{num} = -\frac{1}{\ln m}\sum_{i}^{m} p_{num_i} \ln p_{num_i} \tag{15}$$

其中，

$$p_{L_i} = \frac{L_i^*}{\sum_{i=1}^{m} L_i^*}$$

$$p_{num_i} = \frac{num_i^*}{\sum_{i=1}^{m} num_i^*}$$

最后得到航迹距离权重 w_L 和途经节点数权重 w_{num} 的表达式分别为：

$$w_L = \frac{1 - e_L}{2 - e_L - e_{num}} \tag{16}$$

$$w_{num} = \frac{1 - e_{num}}{2 - e_L - e_{num}} \tag{17}$$

将归一化的航迹距离 L^* 与途经节点数 num^* 加权求和得到等效距离 L_{eq}，即评价函数的表达式为：

$$L_{eq} = w_L L^* + w_{num} num^* \tag{18}$$

综上所述，简化后的无人飞行器的航迹规划模型为：

$$\min_{A}(w_L L^* + w_{num} num^*)$$
$$\text{s.t.}\quad \begin{aligned} &\boldsymbol{Q}+\delta\boldsymbol{P}<\boldsymbol{\theta}\\ &\boldsymbol{Q}+\delta\boldsymbol{P}_v<\boldsymbol{\alpha}\\ &\boldsymbol{Q}+\delta\boldsymbol{P}_h<\boldsymbol{\beta}\\ &\boldsymbol{A}^{-1}(\boldsymbol{Q}+\delta\boldsymbol{P})\odot\boldsymbol{H}=\boldsymbol{Q} \end{aligned} \tag{19}$$

2　快速航迹规划算法

2.1　经典 D 算法介绍

D 算法是一种典型的单源最短路径算法[15]，用于计算图中指定点到其余各点的最短路径。D 算法的基本原理如下：

1)将上文中的全部节点集 $\boldsymbol{G}$ 分为已标记节点集 $\boldsymbol{G}_p$ 和未标记节点集 $\boldsymbol{G}-\boldsymbol{G}_p$ 两部分，标记起点 S。

2)根据权值矩阵 $\boldsymbol{W}$ 选取节点集 $\boldsymbol{G}-\boldsymbol{G}_p$ 中满足误差约束条件且离起点 S 最近的节点 P_j，标记该点。以 $\boldsymbol{G}_p$ 中的节点作为转接点，重新计算 $\boldsymbol{G}-\boldsymbol{G}_p$ 中各点到起点的距离。若新距离小于上次的值，则权值矩阵对应位置更新，否则不变。

3)重复步骤 2)，直到 $\boldsymbol{G}-\boldsymbol{G}_p$ 为空集或找到的节点 P_j 为终点 E，结束循环。

经典 D 算法在简单约束场景下有较好的应用。但是在本文复杂约束场景下，直接使用经典 D 算法存在如下 3 个问题。

1)加入积累误差约束条件后航迹路径最短与路径向远处延伸的目标有时是冲突的。如图 2(a)所示，远处的点 K 使用靠近初始位置的点 I 作为“跳板”时不一定能采用起点 S 到 I 点的最短路径。不妨假设节点 I 为水平校正点，从起点 S 到 I 点的直线航迹满足水平误差 HE 与垂直误差 VE 约束条件，也能进行水平误差的校准。但垂直误差 VE 的积累值太大以致它不能再充当后续节点的中转点，路径无法向后延伸。不过，如图 2(b)所示，若在起点 S 与 I 点间加入垂直校点 C 作为中转点，则垂直偏差 VE 得到及时校正，有利于后续的飞行规划。但是，在经典 D 算法框架下，由于其贪心算法本质[14]以及松弛性的不足，无法精准确定中转点 C 的强制加入条件，从而导致这一问题无法从根源上得到解决。因此，在复杂约束条件下经典 D 算法搜索容易陷入了死循环，无法找到到达终点的路径。

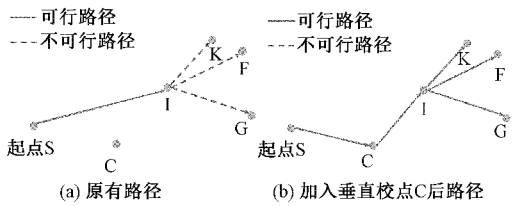

图 2　路径搜索示意图

2)经典 D 算法只能求出最短距离，没有记录路由。当节点数量较大时，若对图中每一个点都用一个专门的集合记录起点到其的途经路线，则会浪费很多存储空间。

3)经典 D 算法为减少时间复杂度常用的堆优化思想[16]在本文稠密图场景下会失效。对于海量节点的稀疏图，王芝麟等[17]将节点按权值升序存放在二叉堆中以快速得到离起点最近的未标记节点 P_j。堆优化后的 D 算法时间复杂度可以由 $O(n^2)$ 降到 $O(n\log n)$。但在本文场景中，节点相对密集，属于稠密图，用堆的存储方式反而会占用更多存储空间，堆优化的思想不仅无法提高搜索效率，反而会少量增加程序运行时间。若要增加 D 算法松弛度，算法复杂度会进一步提高。如何在稠密图中提高 D 算法搜索效率也是一个待解决的问题。

2.2　改进的 D 算法

经典 D 算法只能在简单约束条件下找出最短路径，但是该算法的目标性强、搜索效率高。本文所提出的改进的 D 算法的具体流程如图 3 所示。

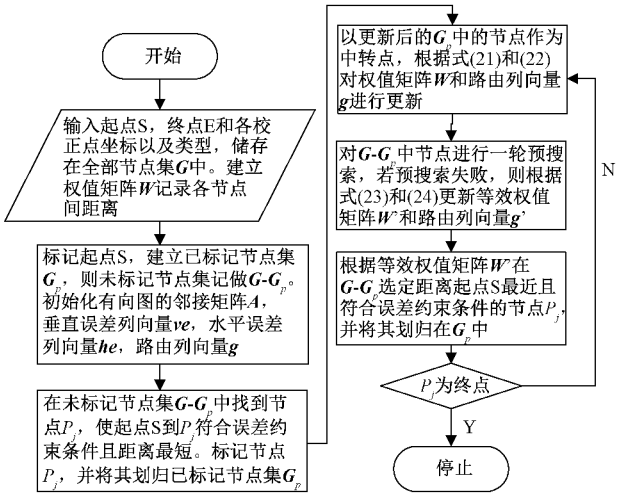

图 3　本文提出的改进的 D 算法流程

为适应复杂约束条件，本文对它进行了如下 4 方面改进。

1)针对经典 D 算法没有记录起点到终点具体路径的问题。受到 Floyd 算法中路由矩阵的启发，用一个长度为 n 的路由向量 $\boldsymbol{g}$ 记录图中每一节点前一个节点的编号。这样只要从终点起对 $\boldsymbol{g}$ 进行简单回溯至起点就能找到途经线路，且实现较为方便。

2)针对本文中等效距离 L_{eq} 最短的航迹规划目标，本文的实现方式是在 D 算法对距离权值矩阵进行更新判别时要求加入中转点后的新路径显著比原路径短。根据式(13)中归一化权重分配 w_{num} 和 w_L 可得到差值 c 的表达式为：

$$c=\frac{w_{num}}{w_L}\cdot\frac{\overline{L}}{num} \tag{20}$$

其中，

$$\overline{L}=\frac{1}{m}\sum_{i}^{m}L_i$$

$$\overline{num}=\frac{1}{m}\sum_{i=1}^{m}num_i$$

更新后的等效权值矩阵 $\boldsymbol{W}'$ 和路由向量 $\boldsymbol{g}'$ 的表达式分别为：

$$\boldsymbol{W}'_{1j}=\min_{\substack{v_i\in G_p\\ v_j\in G-G_p}}(\boldsymbol{W}_{1j},\boldsymbol{W}_{1i}+\boldsymbol{W}_{ij}+c) \tag{21}$$

$$\boldsymbol{g}'_j=\begin{cases}\boldsymbol{g}_j,\boldsymbol{W}'_{1j}=\boldsymbol{W}_{1j}\\ i,\boldsymbol{W}'_{1j}\neq\boldsymbol{W}_{1j}\end{cases} \tag{22}$$

3)针对多约束条件下经典 D 算法路径搜索失败率高的问题，本文通过加入预搜索实现算法的回溯。如图 4 所示，在判断是否将节点 C 作为到达节点 I 的中转点时，若起点 S 到节点 I 已有符合误差限制条件的路径，则先对节点 I 向后进行一轮深度为一的路径预搜索，计算节点 I 在误差约束条件下的出度。如图 4(a)所示，节点 I 按原有路径向后搜索得到其出度为 0，则预搜索失败。这时进行深度为一的回溯，如图 4(b)所示，将节点 C 加入节点 I 的路径中，再进行一轮向后搜索，此时误差约束条件下 I 的出度不为零，预搜索成功。接下来对等效距离矩阵 $\boldsymbol{W}'$ 和路由列向量 $\boldsymbol{g}'$ 进行强制更改，即：

$$\boldsymbol{W}'_{1j}=\boldsymbol{W}_{1i}+\boldsymbol{W}_{ij}+c \tag{23}$$

$$\boldsymbol{g}'_j=i \tag{24}$$

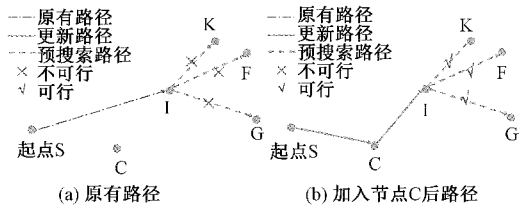

图 4　预搜索示意图

加入预搜索回溯更新条件能从原理上解决路径规划时因追求最短距离而无法向后面点延伸的情况，提高算法松弛性，避免了多约束条件下经典 D 算法可能出现的“找不到路径”的问题。同时，它仍然采用经典 D 算法取最短路径的思想，能够找到全局最优解。

从搜索最优路径的角度，预搜索的深度多多益善。但需要注意，经典 D 算法时间复杂度为 $O(n^2)$，当加入深度为一层的预搜索后，需要对所有未标记点进行一轮遍历，于是时间复杂度提升到 $O(n^3)$。深度每增加一层，时间复杂度加一次幂。也就是说预搜索回溯过程会大大增加算法时间成本，这与快速规划的目标相左。综合搜索效率与时间复杂度，本文采用深度为一的预搜索。

4)为了进一步提高搜索效率，以满足飞行器飞行途中临时更换路径的需要，本文采用在预搜索过程中加入跳出机制的思想减少运行时间。实际上，预搜索的目的是判断原有路径是否能过向后延伸，以确定是否强制加入中转点，而非找到所有可以达到的节点。所以，当向后预搜索能到达点数达到 n_1(n_1 一般取 2)时，就可以认为预搜索成功，跳出遍历循环，不进行回溯，保留原有路径不变，这样可以显著减少算法运行时间。

3　仿真与分析

为了验证本文所提出的改进的 D 算法的有效性，本文使用 MATLAB2020 软件进行仿真，并将其与经典 D 算法和遗传禁忌混合算法[18]进行对照，分别比较了路径搜索质量与效率、算法鲁棒性以及算法对节点密度的敏感度。实验中，首先在 100 km×100 km×10 km 的范围内生成 600 个校正点模拟实际飞行区域，并随机确定其校正点类型。设置飞行器起点 S 为(0，5，5)，终点 E 为(100，60，5)。其他约束条件参数如表 2 所示。

表 2　约束条件参数设置

	误差比	误差阈值	垂直误差校正阈值		水平误差校正阈值	
参数	δ	θ	α_1	α_2	β_1	β_2
数值	0.001	30	25	15	20	25

3.1　路径搜索质量与效率比较分析

本节首先对比了改进后的 D 算法(包括普通回溯 D 算法和快速回溯 D 算法)和禁忌遗传混合算法搜索路径的效率和质量，以检验本文所提出的方法的先进性。图 5 给出了飞行器航迹规划区域以及航迹规划线路图。

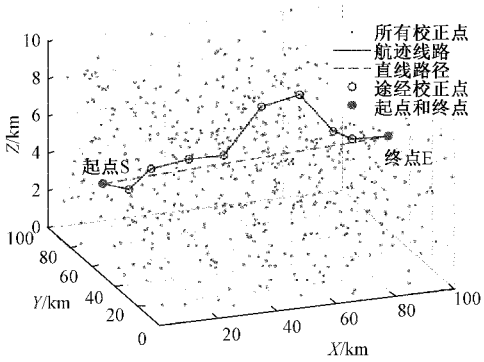

图 5　航迹线路图

不同算法的算法时长和路径长度的对比结果如表 3 所示。

表 3　算法时长和路径对比

算法	路径长度/km	运行时间/s	效率提升/%
禁忌遗传算法	104.9	74	—
普通回溯 D 算法	104.9	0.173	—
快速回溯 D 算法	104.9	0.093	46

结合图 5 和表 3 可以看出，当分别采用禁忌遗传算法、普通回溯 D 算法和快速回溯 D 算法时，3 种算法均找到了

同一条路径，飞行器航迹长度均为 104.9 km，相较于 S、E 两点之间的直线距离 100.5 km 而言，航迹路径只增加了不到 4.5%。且途经校验点分布均匀，校验点之间的平均间距为 11.651 km，航迹路线比较理想。由此可以认为，在本场景下，3 种算法均找到了最佳路径。

从运行时间来看，普通回溯 D 算法和快速回溯 D 算法这两类改进的 D 算法的耗时均小于 0.2 s，搜索效率远远高于禁忌遗传混合算法。假设无人机飞行速度为 800 km/h，采用普通预搜索回溯 D 算法耗时 0.173 s，在此期间无人机飞出约 38 m。而采用加入预搜索跳出机制的快速回溯 D 算法仅耗时 0.093 s，搜索效率又进一步提高了 46%，采用后一种算法时，在执行算法期间无人机飞出距离约 20 m，较前者缩短了近一半。采用上述两种算法时，算法运行期间，无人机飞出的距离均不到节点之间平均间距的 0.35%，满足了飞行器飞行过程中遇到突发情况需要临时改变线路的需求，且后者更符合快速规划的需要。

为了更直观地分析快速回溯 D 算法优于禁忌遗传混合算法的原因，本节又对两种算法的收敛趋势分别进行了仿真，结果如图 6 所示。

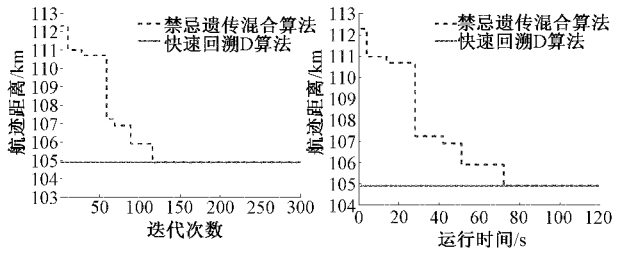

图 6　算法收敛趋势图

由图 6 可以看出，对于禁忌遗传混合算法而言，当迭代次数达到 120 次时才能得到收敛解，共耗时约 74 s。对于飞行速度较高的飞行器而言，该算法无法用于实际飞行过程中进行航迹规划。而本文所提出的快速回溯 D 算法不需要迭代收敛的过程，在取得相同结果的情况下仅耗时 0.093 s，在快速航迹规划中有着明显的优势。

3.2　算法鲁棒性分析

考虑到飞行器的系统结构限制以及校正点的坐标偏差，实际情况下，路径搜索过程中的距离计算误差不可避免。为了验证本文所提算法对抗距离测量误差的性能，本文将无误差的有向图权值矩阵 $\boldsymbol{W}$ 加上干扰噪声来进行分析，干扰项服从均值为 0，方差为 σ^2 的正态分布。同时，本节仿真在 3.1 节的仿真参数设置的基础上进一步增加了航迹规划难度，将校正点的分布密度降低为原来 0.6 倍。图 7 为降低校正点密度后的一次成功的航迹规划案例。

如图 7 所示，当校正点的密度降低后，采用本文所提算法获得的飞行器的航迹路径长度为 108.4 km。与图 5 校正点的密度未降低之前相比，航迹路径长度增大，规划的难度也因此增加。考虑到实际运行过程中不可避免存在测量

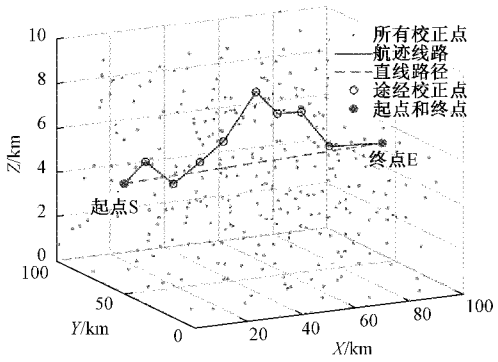

图 7　降低校正点密度后的航迹规划图

误差，本节在校正点密度降低的场景下添加距离测量误差来进行仿真分析。考虑到当相对误差值达到 10%时，测距已经严重偏离实际值，根据概率统计 3σ 原则，对 σ 取值在 0%到 3%之间时进行仿真分析。以 0.5%为间隔对每个 σ 值做 1 000 次路径搜索，再计算路径搜索成功的次数，得到路径搜索成功率如图 8 所示。

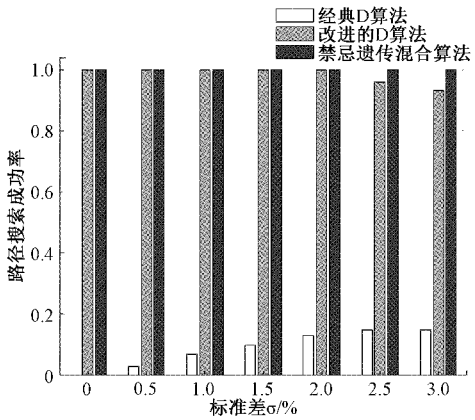

图 8　算法鲁棒性对比

从图 8 中可知，无论距离测量计算是否存在误差，采用经典 D 算法搜索航迹时，路径搜索的成功率总小于 20%，有非常大的概率搜索失败。而加入预搜索的 D 算法在标准差低于 2.0%时总能成功搜索到路径，当测距误差的标准差达到 3.0%时仍然保持了高于 90%的成功率，仅略低于禁忌遗传混合算法。从而验证了本文所提出的算法能够对抗距离测量误差，鲁棒性较好。

3.3　算法对节点密度敏感度分析

为了进一步验证本文所提出的算法的可靠性，本节考虑了不同节点密度条件下 3 种路径搜索算法的性能。本节仿真仍然沿用 3.1 节的仿真参数设置，再将航迹区域范围按比例放大，随机生成相同数量节点，在每个放大倍数下做 1 000 次测试，统计路径搜索成功的次数，从而得到 3 种算法的路径搜索成功率如图 9 所示。

由图 9 可知，路径搜索成功率与航迹区域内的节点密度有很大关系，节点密度越小，路径搜索越困难。当航迹区域校正点密度较大时，经典 D 算法的成功率也较高。但是

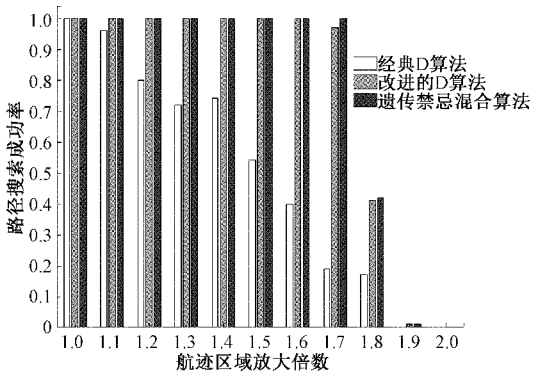

图 9　节点密度敏感性对比

经典 D 算法存在因误差积累导致搜索失败的缺点，当校正点密度变小时搜索成功率会急剧恶化。而加入预搜索的 D 算法在区域放大到原来 1.7 倍，即节点密度为原来 60％时仍能保持 98％的路径搜索成功率。当校正点密度降为 55％及以下时，加入预搜索的 D 算法与禁忌遗传混合算法搜索成功率都显著下降，均不足 50％，两者基本保持一致。由图 9 还可以看到，当区域放大倍数达到 1.9、2.0 倍时，即节点密度为原来的 50％时，本文所提出的加入预搜索的 D 算法和禁忌遗传混合算法的搜索成功率也都接近 0，可以认为此时已经不存在可行路线。由此可见，本文所提出的算法即使在节点密度较低的条件下也能保持较高的搜索成功率，具有较高的可靠性。

4　结　　论

本文采用了归一化熵权法和加入预搜索的 D 算法实现飞行器前进过程中多目标快速航迹规划。归一化过程避免了多目标规划中简单线性加权法量纲不统一问题，而熵权法的使用解决了权值选取随意性问题。通过加入深度为一的预搜索，增加了算法的松弛度，从原理上解决了传统 D 算法在复杂约束条件下容易陷入局部最优而导致路径搜索失败的问题。对于预搜索回溯过程大大增加 D 算法时间复杂度以及稠密图中堆优化失效的问题，本文加入了预搜索跳出机制，有效减少了算法运行时间。仿真结果表明，改进的 D 算法路径搜索结果与禁忌遗传混合算法一致，且搜索效率显著优于禁忌遗传混合算法。另外，鲁棒性分析和节点密度敏感度分析证明了该算法在苛刻条件下仍能保持极高的路径搜索成功率，能够满足飞行器航行途中遇到突发情况需要临时更换路径的快速规划需求。但是改进的 D 算法的路径搜索过程仍然建立在全局遍历的基础上，接下来将会从减少遍历范围角度对该算法做进一步研究与改进。

参考文献

[1]　马学森，谈杰，陈树友，等. 云计算多目标任务调度的优化粒子群算法研究[J]. 电子测量与仪器学报，2020，34(8)：133-143.

[2]　王永强. 基于线性加权和选解法的桥梁联合静动力模型修正[D]. 长春：吉林大学，2021.

[3]　RAMIREZ-ATENCIA C，CAMACHO D. Constrained multi-objective optimization for multi-UAV planning[J]. Journal of Ambient Intelligence and Humanized Computing，2019，10(6)：2467-2484.

[4]　胡玉真，张笋. 基于多目标规划的飞机路径恢复最优化算法研究[J]. 运筹与管理，2020，29(9)：10-17.

[5]　姜辰凯，李智，盘书宝，等. 基于改进 Dijkstra 算法的 AGVs 无碰撞路径规划[J]. 计算机科学，2020，47(8)：272-277.

[6]　左秀峰，沈万杰. 基于 Floyd 算法的多重最短路问题的改进算法[J]. 计算机科学，2017，44(5)：232-234，267.

[7]　董箭，初宏晟，卢杭樟，等. 基于 A 星算法的无人机路径规划优化模型研究[J]. 海洋测绘，2021，41(3)：28-31.

[8]　胡章芳，程亮，张杰，等. 多约束条件下基于改进遗传算法的移动机器人路径规划[J]. 重庆邮电大学学报(自然科学版)，2021，33(6)：999-1006.

[9]　张真诚. 机器人路径规划的改进粒子群-蚁群算法[J]. 电子测量技术，2021，41(8)：65-69.

[10]　吴振，吴红兰. 基于改进遗传算法的无人机航路规划[J]. 电子测量技术，2021，44(24)：52-58.

[11]　李承睿，尹姝屹，毛剑琳. 协同进化算法在三维路径规划中的研究[J]. 电子测量技术，2021，44(11)：73-78.

[12]　杨宇祥，纪群飞，熊智平，等. 电缆故障识别的归一化 STDR 分析方法[J]. 电子测量技术，2021，44(18)：116-121.

[13]　朱毅，杨航，吕泽华，等. 一种基于禁忌搜索的全局最优化模糊聚类算法[J]. 电子学报，2019，47(2)：289-295.

[14]　CUI Y，FANG Y. Research on PCA data dimension reduction algorithm based on entropy weight method[C]. 2020 2nd International Conference on Machine Learning，Big Data and Business Intelligence(MLBDBI)，IEEE，2020：392-396.

[15]　CANDRA A，BUDIMAN M A，HARTANTO K. Dijkstra's and a-star in finding the shortest path：A tutorial[C]. 2020 International Conference on Data Science，Artificial Intelligence，and Business Analytics(DATABIA)，IEEE，2020：28-32.

[16]　GBADAMOSI O A，AREMU D R. Design of a modified Dijkstra's algorithm for finding alternate routes for shortest-path problems with huge costs[C]. 2020 International Conference in Mathematics，Computer Engineering and Computer Science(ICMCECS)，IEEE，2020：1-6.

[17]　王芝麟，乔新辉，马旭，等. 一种基于二叉堆的 Dijkstra 最短路径优化方法[J]. 工程数学学报，2021，38(5)：709-720.

[18]　丁祎男，田科丰，王淑一. 基于遗传禁忌混合算法的敏捷卫星任务规划[J]. 空间控制技术与应用，2019，45(6)：27-32.

作者简介

郑弈，本科，主要研究方向为定位技术和路径规划算法。
E-mail：201983320023@nuist.edu.cn

谢亚琴，工学博士，副教授，主要研究方向为无线资源管理、定位技术和路径规划算法。
E-mail：xyq@nuist.edu.cn