

文章编号: 1002-0640 (2010) 04-0155-03

图结构与 Dijkstra 算法在无人机航迹规划中的应用*

阎 昊, 樊 兴, 夏学知
(武汉数字工程研究所, 武汉 430074)

摘 要: 对无人机任务区域内的航迹可以按照航迹节点和航迹段的方式建立拓扑关系。用图结构的形式对其进行描述, 并用邻接矩阵和权值矩阵对该数据结构进行表达。根据 Dijkstra 算法的原理计算出由出发点 to 各航迹节点间的最短航迹。

关键词: 无人机, 航迹规划, 图, Dijkstra

中图分类号: V 279; V 249. 122 . 3 文献标识码: A

Application of Graph Structure and Dijkstra Algorithm to Track Planning for UAVs

YAN Hao, FAN Xing, XIA Xue-zhi
(Wuhan Digital Engineering Institute, Wuhan 430074, China)

Abstract The tracks in the UAV task area can be used to set up the topological relationships on track nodes and track segments. This paper describes it by graph structure and expresses the graph data structure by its corresponding adjacency matrix and weight matrix. According to the Dijkstra algorithm, we can calculate the shortest track from the starting point to each track node.

Key words UAV, track planning, Graph structure, Dijkstra

引 言

随着无人机 (UAV) 各项技术的不断发展, 无人机已经越来越多的被应用到军事和民用的各个方面, 航迹规划则是无人机的关键技术之一。当无人机在进行日常观测、侦察飞行或者其他任务时, 应在事先确定的任务区域内对航迹进行规划, 无人机将沿着事先规划好的飞行线路执行各项任务。在非战争环境中, 无人机的飞行过程中容易受到来自气候变化, 险要地形阻碍, 观测仪器的范围受限以及飞行时间和燃料供给等各种因素的约束。如何根据飞行过程中的可能会遇到的实际情况对航迹进行规划, 选择一条最短的飞行航迹, 以提高工作效率, 减少飞行时间, 是无人机航迹规划的一个研究方面。

图是一种表达一系列节点和节点间距离关系的数据结构; Dijkstra 算法是一种在图数据结构中解决从网络任一顶点出发, 求它到其他各顶点的最短路径问题的方法。本文通过对整个任务区, 以航迹节点和航迹段的方式建立起图的数据结构, 将 Dijkstra 算法应用到无人机最短航迹的搜索中, 进行了算法实现, 获取了由出发点到各航迹节点的最短航迹线路。

1 无人机任务区域内的图结构的建立方法

无人机在非战争时期执行任务时, 在不考虑外界威胁环境的情况下, 可能有多条可以通航的飞行航迹均能满足预定任务的需求。根据空中各飞行航迹间的基本无障碍特性, 可将多条航迹视为由若干航迹节点和航迹段构成的拓扑集合。这样原来并不在同一条航迹中的航迹节点间也具有构成新的航迹段的可能。当无人机在临时受到外界不利因素的威胁或航迹节点间的通航限制时, 地面可根据这些航迹节点和航迹段重新规划调整无人机的飞行航迹。

收稿日期: 2009-03-28 修回日期: 2009-05-16
* 基金 项目: 船舶重工集团“十五”预研项目 (41901010202)
作者简介: 阎 昊 (1981-), 男, 湖北武汉人, 工程师, 主要研究方向: 计算机软件技术, GIS 应用技

航迹节点视为各个相互独立的节点, 这些节点构成了无人机观测区域内的航迹节点集合。每两个可通航的航迹节点间构成了一条航迹段, 由两个关联节点组成了一个元素对, 则由这些航迹节点所构成的元素对组成了航迹段集合。将所有飞行航迹视为集合 A , 航迹节点集合视为 P , 航迹段集合视为 T , 则有 $A = (P, T)$, 其中集合 $P = \{\text{航迹节点}\}$; 航迹段集合 $T = \{(\text{元素对})\}$ 根据航迹的有向性, 表示航迹段的元素对亦是有序的, 即有方向性的。因此整个任务区域构成一种有向的图结构的表达形式。

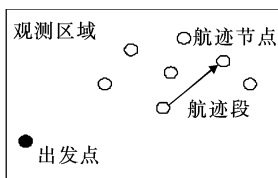
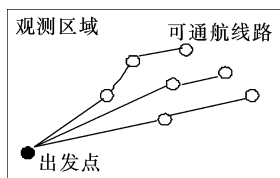


图 1 任务区域内多条飞行 图 2 任务区域内航迹节点和航迹示意图

若集合 A 为由若干航迹节点和航迹段组成的任务区域, 则该区域内航迹节点的集合为 $P(A) = \{P_1, P_2, P_3, \dots, P_n\}$, 航迹段集合为 $T(A) = \{(P_i, P_j) \dots\}$, 则可以用一个 2 维的 $n \times n$ 阶矩阵 $M(A)$ 作为表示航迹节点与航迹段间连通关系的邻接矩阵。对于该矩阵, 如果 P_i 至 P_j 存在一条航迹段即元素对 $(P_i, P_j) \in T$ 则 $M_{i,j}$ 的值为 1, 否则 $M_{i,j}$ 项的值为 0

$$\text{邻接矩阵: } M(A) = \begin{cases} 1 & (P_i, P_j) \in T \\ 0 & (P_i, P_j) \notin T \end{cases}$$

若将各邻接的航迹节点间航迹段的相对距离值 w_{ij} 作为其权值, 则根据邻接矩阵可得到对应的权值矩阵:

$$\text{权值矩阵: } W(A) = \begin{cases} w_{ij} & (P_i, P_j) \in T \\ 0 & (P_i, P_j) \notin T \end{cases}$$

2 Dijkstra 算法在任务区域内搜索最短航迹中的应用

图 3 表示的任务区域内以 A 为出发点, 包括 B, C, D, E 共 5 个航迹节点和 $AB, AD, AE, BC, CB, DB, DE, EB, EC, ED$ 共 10 条航迹段, 并标明了各初始航迹段的权值数据。本文以此为例用 dijkstra 算法进行了最短航迹查找的方法分析。

在图 2 中航迹区域的集合 A 为 $A = (P, T)$, 其中区域内航迹节点的集合 P 为:

$$P(A) = \{P_A, P_B, P_C, P_D, P_E\}$$

由相关联的节点构成的航迹段集合 T 为:

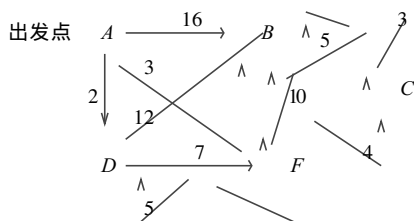


图 3 航迹带权有向图

$$T = \{(P_A, P_B), (P_A, P_D), (P_A, P_E), (P_B, P_C), (P_C, P_B), (P_D, P_B), (P_D, P_E), (P_E, P_B), (P_E, P_C), (P_E, P_D)\}$$

因此该航迹区域内的邻接矩阵为:

$$M(A) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

在集合 T 中, 构成的元素对的航迹节点间的相对距离, 可以视为该航迹段的权值。若 S 为航迹节点 i 至 j 的一条路径, 则在该路径上所有航迹段的权值之和为节点 i 经过 S 到达 j 的权值。没有构成元素对的航迹节点间的权值可视为 ∞ 。因此图 3 的邻接矩阵 $M(A)$ 可以演化成权值矩阵为:

$$W(A) = \begin{bmatrix} 0 & 16 & \infty & 2 & 3 \\ \infty & 0 & 5 & \infty & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & 12 & \infty & 0 & 7 \\ \infty & 10 & 4 & 5 & 0 \end{bmatrix}$$

根据 Dijkstra 算法的思想, 在航路规划中首先以出发点为顶点搜索可与出发点构成元素对的航迹节点中权值最小的点, 然后依次根据每次计算的权值大小选取下一次节点搜索的起始点。算法的主要步骤为: ① 初始化由顶点 V (出发点) 到各航迹节点的权值矩阵; ② 查找在最短路径还没有产生之前最接近顶点的下一航迹节点 P ; ③ 将 P 标记为查找最短路径过程中的下一个顶点; ④ 对于区域 A 中的每一个航迹节点 P_i , 满足从顶点 V 至 P_i 的最短路径还没有产生, 而顶点 V 至 P_i (V, P_i) 所在航迹段存在这一条件, 如果顶点 V 至 P_i 的权值小于它当前的权值, 将该航迹节点的权值进行更新成重新计算出的权值; ⑤ 若有 N 个航迹节点, 则重复以上查找步骤 $N - 1$ 次。算法完成后, 此时得出的该图的权值矩阵第一行即为出发点到各航迹节点的最短路径。

下面为图 3 中应用 Dijkstra 算法计算出的从出

发点到各目标点最短路径的权值矩阵的演变过程:

$$W(A) = \begin{bmatrix} 0 & 16 & \infty & 2 & 3 \\ \infty & 0 & 5 & \infty & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & 12 & \infty & 0 & 7 \\ \infty & 10 & 4 & 5 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 14 & \infty & 2 & 3 \\ \infty & 0 & 5 & \infty & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & 12 & \infty & 0 & 7 \\ \infty & 10 & 4 & 5 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 13 & 7 & 2 & 3 \\ \infty & 0 & 5 & \infty & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & 12 & \infty & 0 & 7 \\ \infty & 10 & 4 & 5 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 10 & 7 & 2 & 3 \\ \infty & 0 & 5 & \infty & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & 12 & \infty & 0 & 7 \\ \infty & 10 & 4 & 5 & 0 \end{bmatrix}$$

以上权值矩阵的演变过程可以用一张二维关系表进行描述, 记录每一次搜索到的权值和最小权值的状况。如表 1~ 表 5, 其中第一行表示当前各节点到各起始间的权值; 第二行中用来跟踪记录从出发点开始已找到的最短路径经过的顶点, 将经比较后将具有最小权值的节点标注为 TRUE, 其他为 FALSE, 在标注为 FALSE 的节点中进行比较, 将权值最小点作为下一次搜索的顶点。(下列表中黑体字所表示的路径为该次搜索中所获得的最小权值)

1) 第一次最短航迹搜索前的初始路径, 搜索顶点为 A, 此时由出发点 A 到各航迹节点中的最短路径为 A-D

表 1 第一次最短航迹搜索结果

以 A 为起始点	A-A	A-B	A-C	A-D	A-E
权值	0	16	∞	2	3
最小权值判定	TRUE	FALSE	FALSE	FALSE	FALSE

2) 第二次最短航迹的查找, 搜索顶点为 D, 此时由出发点 A 到各航迹节点中的最短路径为 A-E

表 2 第二次最短航迹搜索结果

以 D 为起始点	A-A	A-D-B	A-D-C	A-D-D	A-E
权值	0	14	∞	2	3
最小权值判定	TRUE	FALSE	FALSE	TRUE	FALSE

3) 第三次最短航迹的查找, 搜索顶点为 E, 此时由出发点 A 到各航迹节点中的最短路径为 A-E

- C

表 3 第三次最短航迹搜索结果

以 E 为起始点	A-A	A-E-B	A-E-C	A-D-D	A-E
权值	0	13	7	2	3
最小权值判定	TRUE	FALSE	FALSE	TRUE	TRUE

4) 第四次最短航迹的查找, 搜索顶点为 C, 此时由出发点 A 到各航迹节点中的最短路径为 A-E-C-B

表 4 第四次最短航迹搜索结果

以 C 为起始点	A-A	A-E-C-B	A-E-C	A-D-D	A-E
权值	0	10	7	2	3
最小权值判定	TRUE	FALSE	TRUE	TRUE	TRUE

5) 第二次最短航迹的查找, 搜索顶点为 D, 此时由出发点 A 到各航迹节点中的最短路径为 A-E-C-B 此时所得到的各个权值为由出发点 A 到各航迹点的最小权值, 即出发点到各航迹节点的最短航迹

表 5 第五次最短航迹搜索结果

以 B 为起始点	A-A	A-E-C-B	A-E-C	A-D-D	A-E
权值	0	10	7	2	3
最小权值判定	TRUE	TRUE	TRUE	TRUE	TRUE

根据上述方法, 地面航路规划人员可以得到无人机由出发点到任务区域内各个航迹节点的最短距离 当飞行环境发生变化时可以根据计算结果重新规划无人机的飞行航迹

3 仿真实验

本文在某无人机仿真环境中进行了仿真实验, 在预定区域内选择 3 条飞行航迹, 共 40 个航迹节点, 建立任务区域内的航迹节点与航迹段的图结构拓扑关系。根据 Dijkstra 算法得出初始的最短飞行航迹, 飞行环境改变后计算出新的最短飞行航迹 实验演示如第 160 页图 4, 图 5 所示。

4 结 论

本文提出了根据无人机任务区域内可通航线路建立图数据结构的方法, 以航迹节点和航迹段集合

2.3.3 定量的分配电磁兼容参数及最优化

依据评估电磁兼容性能的结果定量的分解、重新调整并分配子系统参数后,再次评估系统电磁兼容性能,此过程不断重复直到系统电磁兼容性达到最优。同时,要测试总系统和子系统电磁兼容性能。

3 电磁兼容性能预测实例分析

图 2所示的是电子战专用航天器上导航系统服务器的电磁干扰树,其中任务发射机 E_2, E_3, E_4 通过天线端口对导航系统产生干扰,理论上,通过改变天线位置 (D_1)、发射功率 (D_2)、天线上的传输功率 (D_3)、发射机频带外的衰减 (D_4)、导航系统天线接收的能量 (D_5)、阻带 (D_6)、敏感度 (D_7)、导噪声信号 (D_8)、偏振损失 (D_9) 可消除干扰。事实上, E_2 和 E_3 的总系统限定参数 $D_1 \sim D_9$ 是不能修改的,因此,为了排除对发射机 E_2 和 E_3 的干扰,分别将导航系统的阻带频率 $f_1 \sim f_6$ 改为 38.6 dB、37.5 dB、36.4 dB、10.6 dB、10.5 dB 和 10.4 dB。对于发射机 E_4 有三个位置可选择。同样为了排除对发射机 E_4 的干扰,天线要安在位置 3,阻带频率 f_7, f_8, f_9 应分别改为 20.5 dB、19.1 dB 和 17.6 dB。

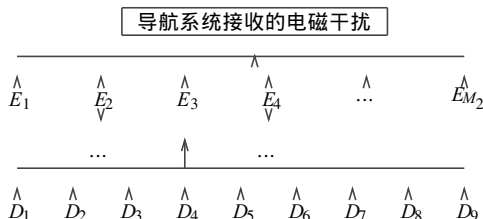


图 2 导弹系统的干扰树

同样地,对每个敏感器的电磁干扰进行分析,通过改变可调整的参数使电磁兼容性能最优化,不断重复最优化的过程使整个系统的电磁兼容性能得到改善,最终得到最合理的电磁兼容设计。

4 结束语

随着电子武器装备的发展和电磁兼容研究的深入,系统设计法这种先进的设计思想和理念将是现代电子系统电磁兼容性设计的总趋势,用这种方法进行系统设计和研制,基本可以避免一般出现的电磁干扰问题或过量的电磁兼容设计问题。该设计法已经成功运用在电子武器装备的电磁兼容应用技术中,并为解决研究和模型发展中的电磁兼容问题提供了技术支持。

参考文献:

[1] 蔡仁刚. 电磁兼容原理、设计和预测技术 [M]. 北京: 北京航空航天大学出版社, 1997.

[2] 马伟明. 电力电子系统中的电磁兼容 [M]. 武汉: 武汉水利电力大学出版社, 1993.
[3] Robinson M P. Analytical Formulation for the Shielding Effectiveness of Enclosures with Apertures [J]. IEEE Trans. On Electromagnetic Compatibility, 1998, 40 (3): 240-248.
[4] 陈淑凤. 电磁兼容性试验技术 [M]. 北京: 北京邮电大学出版社, 2001.

(上接第 157 页)

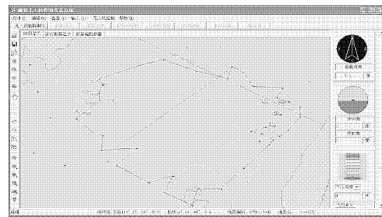


图 4 初始飞行环境中的最短飞行航迹

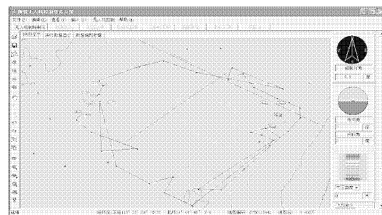


图 5 飞行环境改变后的最短飞行航迹

作为图结构的相关要素进行表达,并以邻接矩阵和权值矩阵表达了航迹节点间的连通和距离关系。应用 Dijkstra 算法,通过权值矩阵的变化演示了最短航迹的查找过程,得到了由出发点到各航迹节点间的最短航迹。

参考文献:

[1] 符晓卫,高晓光. 一种无人机路径规划算法研究 [J]. 系统仿真学报, 2004, 16 (1): 21-24.
[2] 胡晓磊,胡朝晖,江洋溢. 基于 Dijkstra 算法的水平航迹规划 [J]. 火力与指挥控制, 2004, 29 (4): 86-88.
[3] 夏克俭. 数据结构与算法 [M]. 北京: 国防工业出版社, 2001.
[4] Timothy W M, Randal W B. Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles [R]. AIAA 2000243392CP, 2000.
[4] Bortoff S A. Path-Planning for Unmanned Air Vehicles [C]//Proceedings of the American Control Conference, 2000.