

一种考虑转弯约束的改进A*算法在机场滑行道路径规划中的应用研究

文欢¹, 郭建东¹, 张博¹, 韩金池¹, 江广顺², 王揽月³, 周世桢¹, 胡周刁¹

¹电子科技大学, 软件工程学院, 四川 成都

²解放军95007部队, 广东 广州

³95666部队保障部, 四川 成都

Email: 2819448394@qq.com

收稿日期: 2020年12月29日; 录用日期: 2021年1月19日; 发布日期: 2021年2月24日

摘要

航空业务发展迅猛, 机场场面交通变得日渐繁忙, 提高有限滑行道资源的利用率变得逐渐重要。研究进出港航空器的优化路径, 是提高利用率的一种方法, 此时如果使用更加贴近实际的道路网络, 可以更符合现实情况。本文中采用GeoJSON数据格式保存滑行道的形状特征, 构建滑行道网模型, 并加入航空器转弯约束条件, 以A*算法为路径搜索方法, 完成对滑行道的路径规划。本文中因为引入了转弯约束条件, 所以对传统的A*算法作了适应性修改, 最终在模拟的路网中寻路表现良好。

关键词

滑行道, GeoJSON, A*算法, 转弯约束

An Improved A* Algorithm Considering Turning Constraints Is Applied to Taxiway Path Planning

Huan Wen¹, Jiandong Guo¹, Bo Zhang¹, Jinchi Han¹, Guangshun Jiang², Lanyue Wang³, Shizhen Zhou¹, Zhoudiao Hu¹

¹School of Software Engineering, University of Electronic Science and Technology, Chengdu Sichuan

²People's Liberation Army 95007 Unit, Guangzhou Guangdong

³Army Support Department 95666, Chengdu Sichuan

Email: 2819448394@qq.com

Received: Dec. 29th, 2020; accepted: Jan. 19th, 2021; published: Feb. 24th, 2021

文章引用: 文欢, 郭建东, 张博, 韩金池, 江广顺, 王揽月, 周世桢, 胡周刁. 一种考虑转弯约束的改进 A*算法在机场滑行道路径规划中的应用研究[J]. 软件工程与应用, 2021, 10(1): 17-23. DOI: 10.12677/sea.2021.101003

Abstract

With the rapid development of aviation business, airport traffic becomes increasingly busy, and it becomes increasingly important to improve the utilization rate of limited taxiway resources. It is a method to improve the utilization rate to study the optimal path of aircraft entering and leaving the port. At this time, if the road network is more close to the reality, it can be more in line with the reality. In this paper, the shape characteristics of taxiway are saved in the GeoJSON data format, the taxiway network model is constructed, the aircraft turning constraints are added, and the path planning of taxiway is completed by using A* algorithm as the path search method. In this paper, due to the introduction of turning constraints, the traditional A* algorithm is adapted and finally performs well in pathfinding in simulated road networks.

Keywords

Taxiway, GeoJSON, A* Algorithm, Turning Constraints

Copyright © 2021 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

随着我国经济的不断发展, 越来越多民众选择乘坐飞机作为出行方式。但机场在建成后拥有的滑行道资源有限, 面对日益增多的交通运输量, 机场面临的压力也越来越大。虽然机场可以扩建, 但当前的扩建速度远远跟不上航空运输需求的增长, 并且许多机场受场地限制, 导致其扩建范围十分有限。因此, 在有限的滑行道资源上, 提高运营效率, 变得十分有必要。

对进出港航空器的滑行路径进行优化, 在保证安全的前提下, 缩短滑行时间, 是解决当前问题的方法之一。许多学者对此做了大量的研究, 并取得了一定的成果。张兆宁等人结合安全间隔、滑行规则和冲突避免限制规则, 以减少滑行时间和油耗作为目标, 优化滑行路径[1]。李善梅等人提出基于变长的滑行时间窗口和 A*算法结合的方法, 以滑行总时间最短为目标, 研究无冲突的优化路径[2]。姜雨等人引入距离阻抗权重和转向阻抗权重因素, 基于 Dijkstra 算法, 优化航空器在滑行道上的动态调度[3]。

以上的学者以及许多以往的学者, 在研究滑行道路径优化时, 大多使用的是简化的滑行道路网。这些简化的路网图将道路中转弯部分的细节省略, 直接以直线段相连的方式组成路网模型, 将直线段和直线段是否相连代表是否可以转弯换道。这样的处理方法, 除去了实际路网中的许多细节, 在某些情况下, 会导致不合理的路径规划结果。

针对以上问题, 本文对路网模型进行改造, 加入转弯约束条件, 并实现了一个改进的 A*算法对新路网模型进行路径优化。在实际的工程应用中, 取得了良好的效果。

2. 路网处理

在道路交通问题中, 可以将路网抽象为一个图模型[4]。图模型中每一个节点代表一个路口, 每一条边代表一段道路。抽象简化的机场滑行道路网类似于图 1 所示。

图 1 中, 简化的路网只能描述点的位置, 以及点和点之间的连通关系。但在实际滑行道中, 航空器如果需要大幅度转弯, 一般有一段圆滑的弧线衔接两条大角度的路段作为缓冲。为了保存滑行道路径的

更多特征, 可以使用矢量数据格式存储路网数据, GeoJSON [5]就是其中的一种。

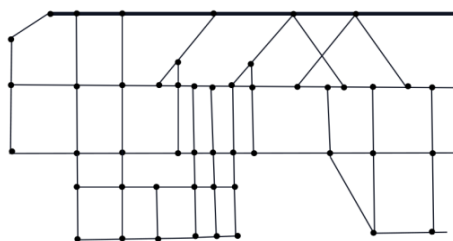


Figure 1. Simplified taxiway network
图 1. 简化的滑行道路网

本文中用 GeoJSON 结构将滑行道路网抽象处理生成 Point 和 Line 两个数据集合, 代表的图模型如图 2 所示。

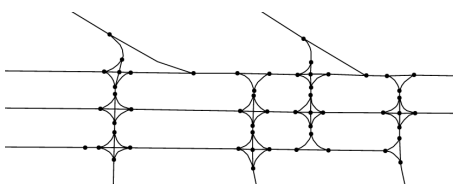


Figure 2. The road network after treatment
图 2. 处理后的路网

Point 集合对应每一个路口和一些重要的位置, 即图 2 中圆点; Line 集合对应每一条道路, 即图 2 中一条条线段, 每一条线段连通着两个圆点。图 2 中圆滑的曲线是引导航器大幅度转弯时的缓冲弧线。最终将滑行道路网抽象处理为满足图模型的结构, 并保存了道路形状信息。

3. 路径规划

3.1. A*算法

A*算法[6]是一种基于启发式的图搜索算法。在寻路的过程中, 为每一个节点赋予一个估计值, 然后选取估计值最优的节点优先遍历。估计值是使用启发式函数估算从起点经过该节点到终点的距离。启发函数如下:

$$F = G + H \quad (1)$$

其中 G 为起点到达当前节点的实际距离, H 为从当前节点到达终点的估算距离。传统的 A*算法一般使用欧式距离或者曼哈顿距离作为 H 的计算方法。

3.2. 加入转弯约束后的问题

依照传统 A*算法的寻路方法, 在下图 3 中搜索路径存在一定问题。

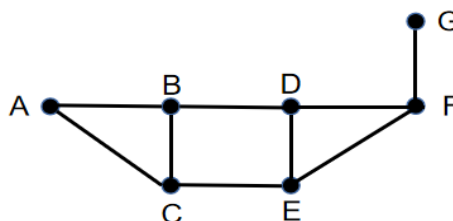


Figure 3. A simulated point-edge graph model
图 3. 模拟的点边图模型

在图 3 中，假设路径中不能存在 90 度及以上的转弯处。如果搜索以 A 为起点，G 为终点的路径。根据 A*算法，搜索到 F 点时，从 A 点到 F 点的最短路径为 A→B→D→F，但在此基础上 F→G 却需要转弯 90 度，因此该条路线不合理，导致算法无法找到合理路径。

但在图 3 中，可以找到从 A 到达 G 的路径 A→C→E→F→G，所以根据加入的转弯约束条件，需要对 A*算法进行一定改进。

3.3. A*算法的改进

在传统的 A*算法中，被扩展节点记录了从起点到该节点的最优路径，但加入了转弯约束条件后，这条最短路径可能并不满足约束条件。如果将到达该节点的所有可能路径都记录的话，就没有发挥 A*算法的优势。

根据滑行道的特点，将道路切换的情况可以归为下图 4 中的四种类型。

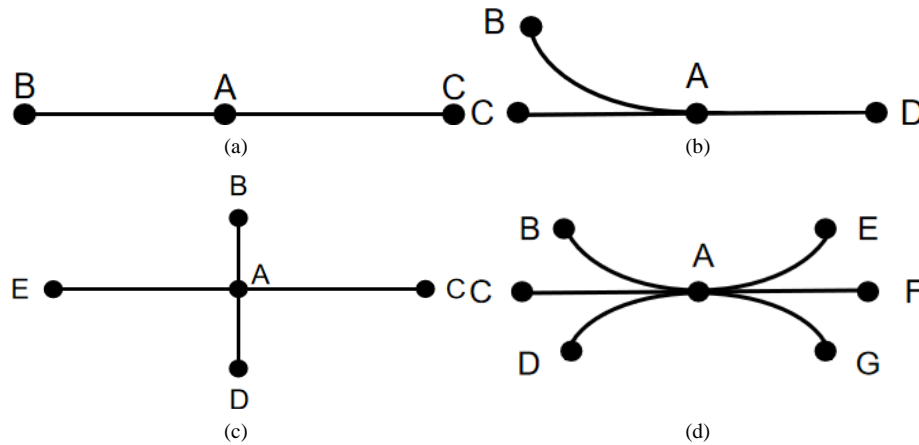


Figure 4. Road switching in taxiway
图 4. 滑行道中道路切换情况

本文中，滑行道路网使用 GeoJson 数据结构记录了道路的形状特征，所以可以根据转弯角度，来判断是否可以转弯。

在图 4 的(a)中，点 A 与 B、C 相连，可以 B→A→C，也可以 C→A→B。因此在 A 点，需要将来自 B、C 方向的最优路径都记录，共两种情况。

在图 4 的(b)中，B、C 都可经过 A 到达 D，所以来自 B、C 方向的最优路径只需要记录为一种，加上来自 D 方向的最优路径，共需记录 2 种情况。

在图 4 的(c)中，因为转弯条件约束，需要点 A 记录四种情况：E→A→C、C→A→E、D→A→B、B→A→D。

对于图 4 中的(d)，B、C、D 都可以通过 A 到达 E、F、G，所以扩展时，从 B、C、D 到达 A 可以视为一种情况。同理 E、F、G 也被视为一种情况。A 点只需要记录(B/C/D)→A→(E/F/G) 和 (E/F/G)→A→(B/C/D)两种情况。

A*算法改进后，同一个节点需要记录来自不同方向的最优路径，可以通过生成多个节点副本来实现。因为滑行道有其自己的特点，每个节点最多记录四种情况，所以最多生成四个副本，就可以将所有情况全部纪录。可以让每个节点都生成四个副本，也可以分析每一个节点的情况，生成对应情况数的副本数量。

3.4. 改进后 A*算法的步骤

步骤如下：算法需要维护 M 和 N 两个集合，M 保存哪些搜索到但还没有加入最优路径树的节点，N

保存加入了最优路径树的节点。为每个节点生成四个副本。

- 1) 取起始节点的一个副本，初始化，并放入 open 中。
- 2) 如果 M 为空，则搜索失败；否则从 M 中选出 F 值最小的节点 P，并将其加入 N 中。
- 3) 如果点 P 是终点，则结束。
- 4) 遍历点 P 相邻的可达节点：
 - ① 检查从 P 点到达相邻点是否满足转弯约束。如果不满足则跳过，继续检查下一个相邻点；如果满足则进入②。
 - ② 如果相邻点不在 M 和 N 中，则取该节点一个副本，初始化，设置父节点为点 P，加入 M 中。
 - ③ 如果相邻点已经在 M 或 N 中。
 - a. 取出 M 和 N 中的所有该节点副本，与当前情况相比较。
 - b. 如果相同情况存在。则在相同情况的副本中，计算通过点 P 到达该节点的 F 值是否更小。如果更小则更新 F 值，并将其父节点设置为点 P，此时如果其在 N 中，则将其从 N 中移到 M 中。
 - c. 如果相同情况不存在。则取该节点一个副本，初始化，设置父节点为点 P，加入 M 中。
- 5) 转到步骤 2。

3.5. 改进后 A*算法的寻路过程

在图 3 的基础上，仍然假设路径不能有 90 度及以上的转弯，但让节点记录来自不同方向的最优路径，寻路从 A 到 G 的过程如下图 5 所示。

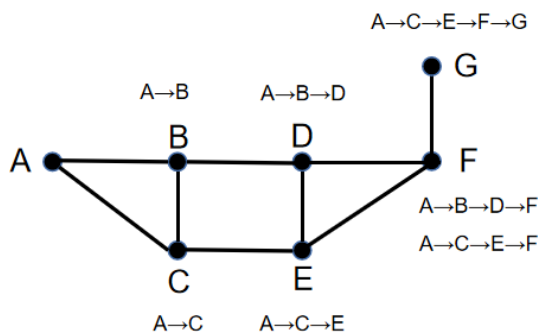


Figure 5. The improved A* algorithm pathfinding process

图 5. 改进后 A*算法寻路过程

图 5 中，在 F 点处，来自于 D、E 方向的最优路径下一步能达到的点不同。来自 D 方向的在 F 处没有下一步可达点；来自 E 方向的在 F 处下一步可达点有 G。因此在 F 处纪录 A→B→D→F 和 A→C→E→F 两种情况。

4. 应用实例结果

图 6 是一幅增加加入了更多特征，更贴近实际的滑行道路网图。

引入转弯约束，在图 6 的基础上，使用改进的 A*算法搜寻从 A→G 的路径。图 6 中粗线条路径是这次寻路的结果。从图中可以看出改进的 A*算法可以成功完成路径规划，粗线条路径连通了 A 点和 G 点，并且路径曲线符合滑行道路转弯约束条件。

但是，如果使用传统的 A*算法或者 Dijkstra 算法，是无法在图 6 的路网图中搜索到 A→G 的路线，导致路径优化失败。具体情况如下。

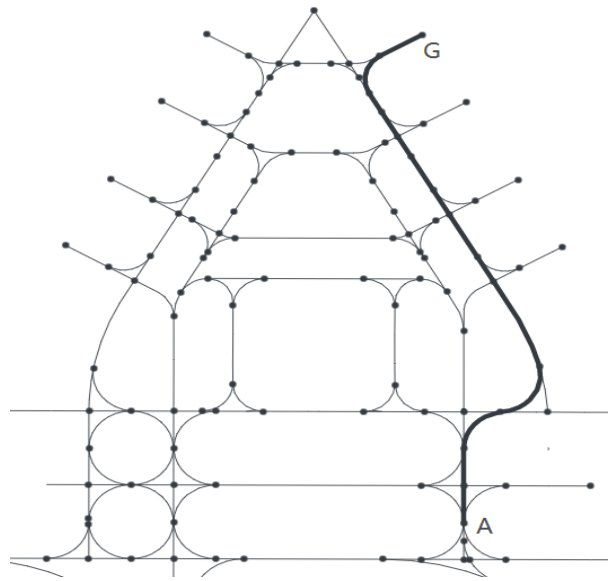


Figure 6. Pathfinding results of the improved A* algorithm
图 6. 改进的 A*算法的寻路结果

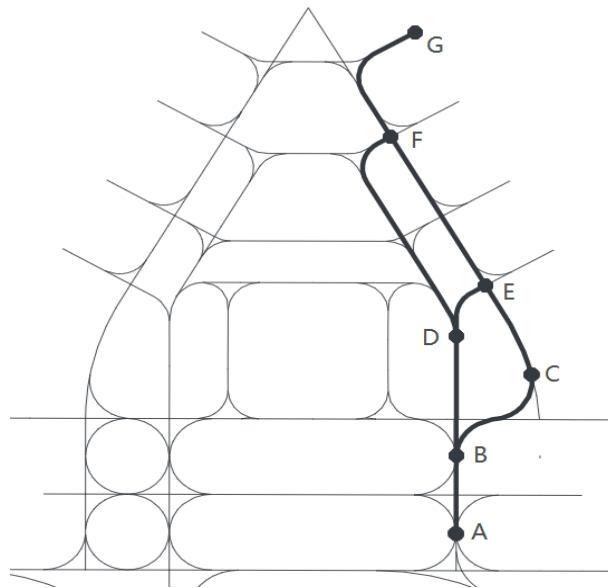


Figure 7. Path finding failure diagram
图 7. 寻路失败情况图解

在图7中,传统的A*算法或者Dijkstra算法只会保存两点之间最短的一条路径。当传统算法搜索A→G路径时,在E点上,只会纪录A→B→D→E,而A→B→C→E路径因为更长而被抛弃,由于转弯条件约束,E点无法进一步E→F→G。同样的在F点处只会纪录A→B→D→F,但F无法进一步到达G点。因此传统的A*算法或Dijkstra算法在这种情况下寻路失败。

5. 结论

本文中,提出使用矢量结构GeoJSON描述滑行道路特征,以更贴近实际情况的机场滑行道路网图作为寻路对象。提出在传统A*算法的基础上进行一定改进,使其满足滑行道的转弯约束条件,进而能在贴近

实际的滑行道路网中进行路径规划。应用实例结果显示,改进后的 A*算法能在贴近实际的滑行道路网中良好工作。本文研究结果可以为机场安全态势分析、冲突预警等方面研究提供部分的参考方案。

参考文献

- [1] 张兆宁,王彤. 基于 Dijkstra 算法的机场滑行路径优化[J]. 航空计算技术, 2018, 48(6): 1-5, 10.
- [2] 李善梅,高艺. 基于改进 A*算法的机场场面滑行路径优化[J]. 计算机仿真, 2020, 37(3): 27-32, 228.
- [3] 姜雨,陈丽丽,刘振宇. 时空网络下的航空器滑行路径动态规划[J]. 航空计算技术, 2020, 50(1): 25-28.
- [4] 陈道蓄. 最短路径问题[J]. 中国信息技术教育, 2020(13): 18-22.
- [5] 胡栋鹏,胡晶. 基于 OpenLayers 的矢量数据可视化系统[J]. 计算机产品与流通, 2018(2): 165-166.
- [6] 梁晓辉,慕永辉,吴北华,等. 关于路径规划的相关算法综述[J]. 价值工程, 2020, 39(3): 295-299.