计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(2):255−259 [doi: 10.15888/j.cnki.csa.007772]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于改进 A*算法的无人机避障路径规划①

李晓辉, 苗　苗, 冉保健, 赵　毅, 李　刚

(长安大学 电子与控制工程学院, 西安 710064)
通讯作者: 苗　苗, E-mail: 963623514@qq.com

摘　要: 近年来, 物流行业的飞速发展, 运输是物流的重要环节之一, 根据数据显示, 运输的成本占据整个物流成本的 50% 以上. 无人机的使用有效的控制了运输成本, 合理规划物流无人机的飞行路线, 也起着至关重要的作用. 在物流无人机的航迹规划中, 必须保证无人机飞行过程中能够准确避开禁飞区. 本文基于 A*算法, 结合多种类型的禁飞区, 设计出一种改进算法, 能够找到任意两客户点间无人机避障飞行的最优路线. 仿真结果表明, 本文所设计的算法能够有效解决多类型禁飞区并存的无人机避障路径规划问题.

关键词: 无人机; 路径规划; 避障; A*算法

引用格式: 李晓辉,苗苗,冉保健,赵毅,李刚.基于改进 A*算法的无人机避障路径规划.计算机系统应用,2021,30(2):255–259. http://www.c-s-a.org.cn/1003-3254/7772.html

Obstacle Avoidance Path Planning for UAV Based on Improved A* Algorithm

LI Xiao-Hui, MIAO Miao, RAN Biao-Jian, ZHAO Yi, LI GANG

(School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, China)

Abstract: In recent years, transportation has been playing an important role in blooming logistics. Indeed, transportation accounts for more than 50% of the whole logistics cost. Express UAVs could effectively reduce the cost. Moreover, proper path-planning of UAV is also essential. Particularly, UAV should accurately avoid non-fly zones during flight. In this study, the obstacle avoidance path planning of UAVs is comprehensively discussed. With the improved A* algorithm and considering various obstacles, we propose a method that can find the optimal obstacle avoidance path between any two customer points. The simulation results prove the effectiveness of solving the obstacle avoidance path planning problem for express UAVs.

Key words: UAV; path planning; obstacle avoidance; A* algorithm

无人机作为我国科技创新的重要产业, 正处于井喷式发展时期. 它的技术日渐成熟, 能够执行复杂的危险任务. 多用于军事侦查、电力巡检、货物运输、物流配送等, 在航空领域占有一席之地. 随着自动控制技术、传感技术、导航技术、实时监控技术、计算机技术的飞速发展, 无人机的功能也越来越完善, 应用领域不断扩大. 因此, 研究无人机路径规划问题有着重要意义[1,2]. 我国物流运输行业迅猛发展, 已经成为我国国民生产的重要组成部分, 与人们的生活密切相关. 而随着市场需求的增大, 配送成本也在不断升高. 合理有效的运输方案, 可以帮助企业降低运输成本, 提高运输质量, 同时还能实现节能减排的环保目标[3]. 无人机因其高效、灵活、低成本的特点, 渐渐成为各大物流公司的宠儿. 无人机本身是科技的产物, 应该合理正确的使用,

① 基金项目: 陕西省科技厅重点产业创新链项目 (2020ZDLGY-09-03, 2019ZDLGY03-01); 西安市科技项目 (201805045YD23CG29)
Foundation item: Key Industry Innovation Chain Project of Science and Technology Bureau, Shaanxi Province (2020ZDLGY-09-03, 2019ZDLGY03-01); Science and Technology Program of Xi'an Municipality (201805045YD23CG29)
收稿时间: 2020-06-14; 修改时间: 2020-07-10; 采用时间: 2020-07-17; csa 在线出版时间: 2021-01-27

© 中国科学院软件研究所 http://www.c-s-a.org.cn

不应该再出现无人机碰撞飞机、使用无人机偷窥等负面新闻, 这些都是十分危险的行为. 再加上我国严格的航空管制, 设置禁飞区显得十分有必要. 因此, 无人机路径规划问题的研究常常与禁飞区避障问题相结合.

路径规划问题一直是国内外学者研究的热点, 无人机的路径规划问题最初是由旅行商问题 (Traveling Salesman Problem, TSP)[4] 和车辆路径问题 (Vehicle Routing Problem, VRP)[3] 衍变而来. 无人机避障路径规划问题分为二维和三维两种情况. 本文主要研究二维平面的路径规划问题. 目前大多数的二维平面路径规划问题中, 禁飞区的类型都只是单一的, 比如只存在多边形障碍物[2], 或者只存在圆形障碍物[5,6]. 文献 [2] 提出了一种改进的 A*算法来规划无人机路线, 但只针对单一的矩形禁飞区. 文献 [5] 提出顺序凸规划问题逼近非凸零件的算法来规划无人机航线, 是只针对圆形禁飞区设计的. 文献 [6] 将融合简化稀疏 A*算法与模拟退火算法相结合, 规划出了一条代价较低的航迹, 但禁飞区的类型也是单一的圆形. 本文研究的是多类型禁飞区共存的情况, 即同时包含有圆形、多边形障碍物, 航迹也可以贴着禁飞区的某条边飞行, 这大大提高了算法设计的难度. 这类问题最常用到的算法包括模拟退火算法[6]、改进的粒子群优化算法 (Particle Swarm Optimization)[7]、蚁群算法 (Ant Colony Algorithm)、几何法[8] 等. 文献 [6] 提出了将 A*算法与模拟退火算法相结合来解决无人机避障的路径规划问题, 能够利用较少内存, 快速的得到一条综合代价较低且较为平滑的航迹. 文献 [7] 提出了一种改进的粒子群算法来规划无人机航向路线, 相比较传统的粒子群算法, 改进后的算法适应性更强, 环境规划策略更加灵活, 但是集中式计算负担过大, 有时不能满足需要. 文献 [8] 提出了一种基于几何法的路径规划方法, 在航迹部分加入了螺旋线航迹. 本文采用的是改进后的 A*算法来解决多类型禁飞区下无人机路径规划的问题.

1 问题描述

本文研究的问题是通过算法找出无人机从任意起点到任意终点运输代价最小的安全避障路线. 假设某指定区域内有多个不同类型的禁飞区, 假设物流无人机均为充电旋翼式无人机, 对货物载重、航行时间、电池能耗等有约束限制. 为了确保货物能够安全准时运输至指定客户点, 需要对无人机进行路径规划. 本文假设仅对物流无人机航行阶段路径进行规划, 则无人机的运动可简化为二维平面运动.

无人机路径规划需要满足以下原则:

① 无人机飞行起始点记为 S, 终止点记为 T.

② 任务规划应该得到一条从任务起点到任务终点的路径.

③ 要在无人机避过禁飞区的同时, 保证规划的路径应尽量最优化, 满足无人机航程最短的要求.

无人机路径规划的实现主要包括两个步骤: (1) 环境建模. 根据无人机所在的实际环境, 在仿真系统中建立合适的模型, 将实际的物理空间抽象成能用数学模型求解的虚拟空间. (2) 路径搜索[9]. 根据任务条件规划出一条从任意起点 S 到任意终点 T 的任务路线, 在满足任务条件约束、满足避障要求的同时使路径代价达到最短.

本文想要实现的目标是: 无人机能在准确避过多类型障碍物的同时, 最小化航行里程.

2 算法设计

A*算法是一种静态路网中求解最短路径最有效的直接搜索算法, 它通过启发函数来引导算法的搜索方向. 针对本文研究的问题, 对 A*算法做了一定的改进. 首先输入无人机飞行起始点 S, 终止点 T. 建立两个数组 $C1$、$C2$, 用来存放无人机所经过节点的信息.

$$D_i = f_i + g_i \tag{1}$$

上式中, f_i 表示起点 S 到节点 i 的有效已走距离, 这里的有效已走距离是指已经顺利避过障碍物的路线距离; g_i 表示节点 i 到终止点 T 的直线距离, 这里的直线距离是指不考虑障碍物时节点 i 到终止点 T 的直线距离, 即剩余最小距离, 这是一个最小估算值. D_i 是 f_i 与 g_i 的和, 表示节点 i 目前的最小总代价值.

算法重要步骤:

Step 1. 输入起始点和终止点的坐标, 将起点存入 $C1$ 中.

Step 2. 遍历 $C1$, 找到最小值 $D_{\min}$ 对应的节点对它进行展开, 即找到距离该节点最近的障碍物, 获取该障碍物的节点信息, 将该障碍物的可用节点添加到 $C1$ 中, 并把 $C1$ 中已展开的点移至 $C2$ 中.

Step 3. 重复 Step 2, 直到当前展开点与终点连线不再经过任何障碍物, 可直接到达终点时停止, 或者直到 $C1$ 为空时停止循环.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

Step 4. 对得到的最优路径再次检查修复.

Step 4 是对路径的修复, 如图 1 所示, 假设我们已经找到 S 到 T 点的最优路径, 图中可以看出 S 点到 V_4 点的连线不穿过禁飞区, 可以直达, 这种情况下就可以删除 S 到 V_4 之间的多余节点; 而 S 到 V_6 点之间的连线穿过了禁飞区, 则不可修复. 为了最终结果更加完美, 代码中对每个节点都进行了检测, 测试它们到其他节点是否可直达, 若可达则删除多余节点来修复路径[10].

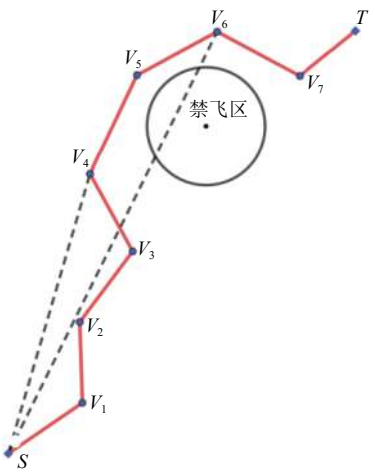

图 1 路径修复图

重点算法部分的伪代码见算法 1 和算法 2.

算法 1. 重点算法

输入: 起点 S 坐标, 终点 T 的坐标, 障碍物信息
输出: 最优解 best_solution

```
//C1, C2 保存点的信息, 包括每个点的 f 值、g 值、坐标、上层节点
读取障碍物点信息, 起始点信息;
设 best_solution 的初始 f 值为 inf;
判断 S→T 是否闯过障碍物, 不穿过障碍物可直接返回;
若穿过障碍物, 则将 S 加入到 C1 中, 进入 while 循环;
While 停止机制: 当前迭代的节点与终点 T 连线不再穿过任何障碍
物, 或者 C1 为空
    找 C1 中找到 min(f+g) 对应的节点 minPoint, 对它展开;
    找到距离 minPoint 最近的障碍物的顶点信息 (若是圆形障碍物,
找左右两个切点);
    If 顶点信息输出为空 (即已经不穿过任何障碍物)
        更新 best_solution, 跳出循环;
    else 继续寻找
        更新 best_solution;
        更新 C1 (删去 C1 中已展开的点, 加入最近障碍物顶点);
end
修复 best_solution 中的路线 (具体代码见算法 2);
画出路线;
结束.
```

算法 2. 修复部分算法

输入: best_solution
输出: new_best_solution

```
//num 表示最优解 best_solution 中所经过节点的个数
for i=1,2,⋯, num–2
  for j=i+2,i+3,⋯, num–1
      判断 best_solution 中第 i 个点到第 j 个点的连线是否闯过障碍物;
      if 没有穿过障碍物
          将点 i 和点 j 间所有的多余节点删去;
      end
  end
end
更新 best_solution 中的路线即最终距离;
new_best_solution=best_solution;
结束.
```

举例说明:

假设现有多边形障碍物 3 个, 圆形障碍物 3 个. 如图 2 所示, 起始点为 S, 终止点为 T, 创建 $C1$、$C2$, 它们保存的节点信息包括: f 值、g 值、节点坐标、上层节点坐标. 从起始点 S 出发, 将 S 保存到 $C1$ 中, 遍历 $C1$, 找到 $D_{\min}$ 对应的节点 S, 对 S 展开, 找到距离 S 最近的障碍物节点 P_4、P_5、P_6, 将 P_4、P_5、P_6 中可用的节点添加到元胞数组 $C1$ 中, 并将 $C1$ 中已展开的点 S 移动到 $C2$ 中; 再次遍历 $C1$, 找到 $D_{\min}$ 所对应的节点 P_6, 再继续寻找距离 P_6 点最近的障碍物, 并将该障碍物的可用节点信息存入 $C1$ 中, 这里注意, 当找到的距离 P_6 点最近的障碍物是圆形时, 求 P_6 与终点 T 之间的左右两条切线, 将可用切点当作备选点存入 $C1$ 中; 再次遍历重复上述步骤, 直到当前展开点与终点 T 的连线不再穿过任何障碍物时, 就可得出 S 到 T 的最优路径.

寻优的结果是要经过多次循环产生的, 图 3 完整的复现了这一过程, 图中数值表示的是由公式 1 得出的 D 值. 算法中的每次循环都要遍历 $C1$, 找出最小值 $D_{\min}$ 对应的节点, 继续往下分支. 图 3 中 P_{12} 下的分支 $3B$ 表示节点 P_{12} 和终点 T 关于圆形障碍物 3 所做切线得出的切点, P_{11} 下的分支表示节点 P_{11} 和终点 T 关于圆形障碍物 3 所做切线得出的切点 1 和切点 2. 这里注意, 可用切点不一定都能找到两个, 有的切点严重偏离当前圆形障碍物时, 就属于不可用切点, 不予以考虑. 图 3 中可以看出障碍物点 P_4、P_5、P_6 多次出现, 但它们对应的上层节点有所不同, 所以它们的 f 值、对应的 D 值也都不相同, 代表着各自的意义. 图中有向下箭头分支的节点表示在某次循环中已经被展开过的点, 这类点要及时移动到 $C2$ 中.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

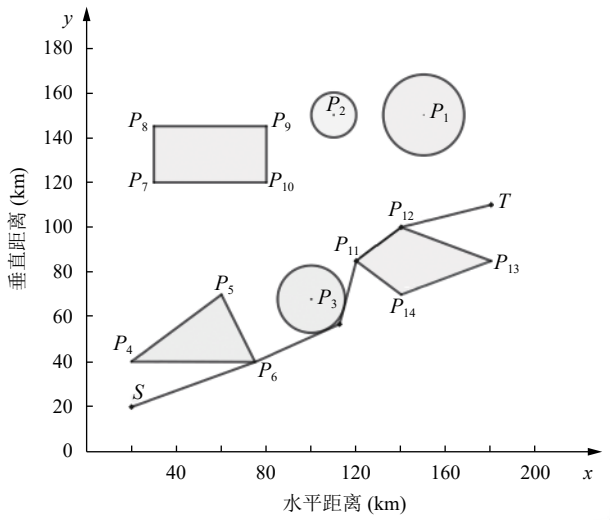

图 2　路径规划图

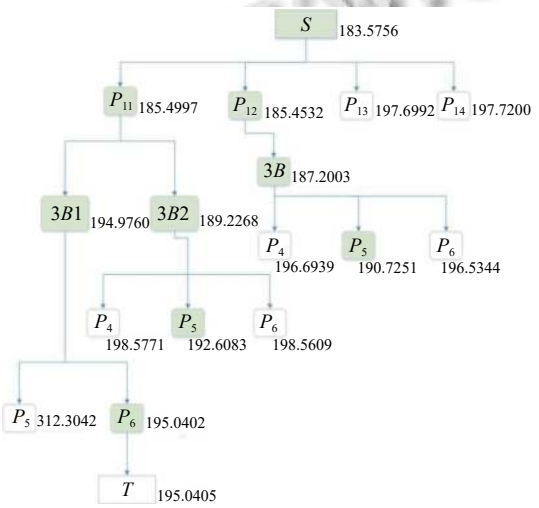

图 3　路径分支图

3　仿真分析

我们对改进的 A*算法进行仿真实验, 对实验结果进行了分析. 实验中用到的 PC 系统为 Windows10, 处理器型号为 Intel(R) Core(TM) i7-8700, 主频率为 3.19 GHZ, 开发环境为 Matlab 2016b, 仿真实验过程通过 Matlab 编程语言实现, 使用 Matlab 对规划环境及规划出的航迹进行作图. 实验中的环境模型大小为 800 km×900 km, 同时包含了多种类型的禁飞区, 共 14 个禁飞区, 其中 6 个圆形禁飞区, 8 个多边形禁飞区.

通过仿真实验, 模拟了 4 条路径, 如图 4 所示, 每条路线的途径点信息如表 1 所示. 可以清楚的看到我们设计的最优路线允许贴着禁飞区的边飞行, 我们的算法也能解决多种形状并存的壁障问题.

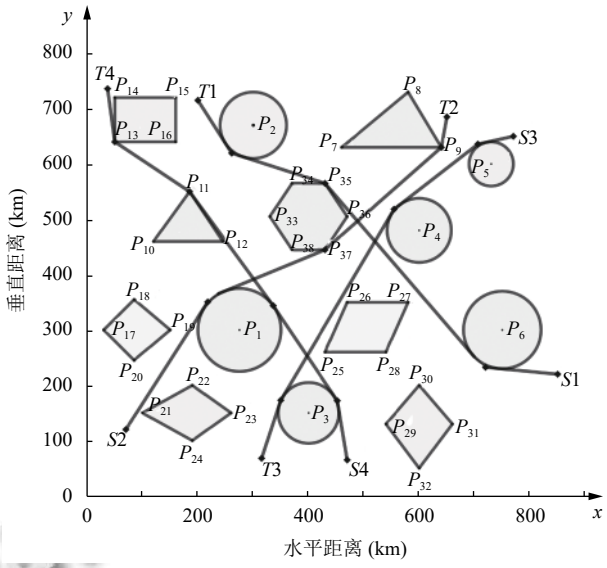

图 4　路径规划图

表 1　最优路径信息

路径序号	途经点信息	代价大小(km)
S1(850,220)→T1(200,715)	(719.715 8,232.602 7)→(430,565)→(261.063 9,619.518 3)	862.844 6
S2(70,120)→T2(650,685)	(217.648 2,350.453 1)→(430,445)→(640,630)	842.695 0
S3(770,650)→T2(315,68)	(705.941 2, 635.586 8)→(554.636 6,518.215 3)→(349.650 8,172.930 1)	769.203 9
S4(470,65)→T4(37,736)	(452.076 7,172.029 8)→(335.938 0,343.773 0)→(185,550)→(50,640)	830.533 7

4　结束语

本文改进了 A*算法, 解决了圆形、多边形禁飞区共存情况下的避障路径规划问题, 对比只有单一类型禁飞区的问题有一定的实用性, 得到的路线更接近最优. 但也有一些不足之处, 比如没有应对突发威胁时的实时航迹规划手段. 本文仅讨论无人机避障路径规划的问题, 后续会对综合的多仓库、多约束条件下的物流无人机路径规划问题进行研究, 在算法自主实时性方面寻求突破, 完善无人机航迹规划策略.

参考文献

1 Goodchild A, Toy J. Delivery by drone: An evaluation of

© 中国科学院软件研究所　http://www.c-s-a.org.cn

unmanned aerial vehicle technology in reducing CO_2 emissions in the delivery service industry. Transportation Research Part D: Transport and Environment, 2018, 61: 58–67. [doi: 10.1016/j.trd.2017.02.017]

2 许卫卫, 张启钱, 邹依原, 等. 改进 A*算法的物流无人机运输路径规划. 华东交通大学学报, 2019, 36(6): 39–46.

3 刘霞. 车辆路径问题的研究 [博士学位论文]. 武汉: 华中科技大学, 2007.

4 Dimitrijević V, Šarić Z. An efficient transformation of the generalized traveling salesman problem into the traveling salesman problem on digraphs. Information Sciences: An International Journal, 1997, 102(1–4): 105–110. [doi: 10.1016/S0020-0255(96)00084-9]

5 Zhang Z, Li JX, Wang J. Sequential convex programming for nonlinear optimal control problems in UAV path planning. Aerospace Science and Technology, 2018, 76: 280 –290. [doi: 10.1016/j.ast.2018.01.040]

6 杨玉, 金敏, 鲁华祥. 融合简化稀疏 A*算法与模拟退火算法的无人机航迹规划. 计算机系统应用, 2019, 28(4): 25–31. [doi: 10.15888/j.cnki.csa.006864]

7 熊华捷, 蔚保国, 何成龙. 基于改进粒子群算法的 UAV 航迹规划方法. 计算机测量与控制, 2020, 28(2): 144–147.

8 常波, 王瑞. 基于几何法的无人机航迹规划. 计算机系统应用 , 2015, 24(1): 109 –113. [doi: 10.3969/j.issn.1003-3254.2015.01.019]

9 田景凡, 王彦恺. 基于弹性绳算法的无人机路径规划方法. 军民两用技术与产品, 2019, (8): 59–63.

10 成浩浩, 杨森, 齐晓慧. 面向城市环境的四旋翼无人机在线避障航迹规划方法. 计算机科学, 2019, 46(4): 241–246. [doi: 10.11896/j.issn.1002-137X.2019.04.038]

www.c-s-a.org.cn

© 中国科学院软件研究所 http://www.c-s-a.org.cn