

蚁群算法在解决 空中交通飞行冲突中的应用

郭 茜¹ 聂润兔¹ 王 超¹

中国民航大学，空中交通管理学院，天津 300300

摘 要：本文主要研究了蚁群算法在解决空中交通冲突问题中的应用。首先对空中交通冲突解决问题的研究背景以及研究现状进行了系统的概述，随后建立了相关的数学模型，将空中交通飞行冲突问题转化为有约束的非线性整数规划问题，并设计了基于蚁群算法的求解思路。最后的仿真计算证明该方法在较短时间内能提供多种不同的有效的冲突解决方案。

关键词：空中交通管理；冲突解决；蚁群算法；有约束非线性整数规划

中图分类号：V355.2

文献标识码：A

文章编号：1672-4747(2009)02-0116-05

Application of Ant Colony Algorithm to Aircraft Conflict Resolution

GUO Qian¹ NIE Run-tu¹ WANG chao¹

College of Air Traffic Management,

Civil Aviation University of China, Tianjin 300300, China

Abstract: This paper mainly focuses on the application of Ant Colony Algorithm (ACA) to air traffic conflict resolution problem. Firstly, the background of the conflict detection and research situation in the world is systematically introduced. Then, a mathematic model which transforms air traffic conflict resolution problem into a constrained nonlinear integer optimization problem was established, and a solving process based on ACA was designed. Finally, the paper undertook a simulating computation of the algorithm and showed that this method is feasible as it can give many different effective solutions for a given problem in a short amount of time.

收稿日期：2008-04-14.

基金项目：天津市科技支撑计划重点项目“面向新一代的空管自动化应用模拟系统”，编号 07ZCKFGX01600。

作者简介：郭 茜（1975-），女，河南南阳人，中国民航大学空中交通管理学院硕士。

Key words: Air traffic management, conflict resolution, ant colony algorithm, constrained nonlinear integer programming problem

0 引言

随着空中交通需求的不断增长,空中交通系统面临着越来越严重的航线拥挤,空域系统的容量已逐渐成为被严重关注的问题。从20世纪90年代开始,美国等一些航空业发达国家提出“自由飞行(Free flight)”的概念,即在开放的天空中,飞机速度和所飞的路径由飞行员自己决定。自由飞行解决空中航路拥挤的局面,更高效地利用空间资源开辟了另外一条新的道路。飞行冲突探测与解决是自由飞行的关键问题,世界各国的研究者们做了大量的有关自由飞行情况下飞行冲突的探测与解决的研究工作。文献[1]提出了基于优先权的最优化算法,给每一架飞机赋予不同的优先权,权值高的飞机在不考虑权值低的飞机的情况下选择出自己认为最优的线路,如此类推,其基本的方法是寻找树的最短路。文献[2]应用了遗传算法,并且把飞行过程中影响速度的不确定因素也考虑了进去。文献[3]建立了混合整数规划的数学模型,用改变航向的方法解决飞行冲突。文献[4]将Delaunay方法应用于多架飞机之间的自由飞行冲突探测。文献[5]对这些众多的方法作出概括、分类和总结。

蚁群算法是基于自然界中蚁群觅食的过程建立起来的,它是由意大利学者Dorigo M等提出^[6],最早应用于成功解决旅行商(TSP)问题。在过去短短十多年的时间里,已在组合优化、网络路由、函数优化、数据挖掘、机器人路径规划等领域获得了广泛的应用,取得了较好的效果。本文根据我国空管安全规定,并结合民用飞机的实际情况,对飞行冲突的探测和解决问题进行合理简化^[7],将该问题转化为有约束的非线性整数规划问题,建立了相应的数学模型。在参考国内外针对具有组合优化性质的极小化问题的蚁群算法^[8,9]的基础上,做了一定程度的改进,并应用该算法对自由飞行条件下冲突解决问题进行了研究。

1 问题提出与模型建立

为避免飞机在自由飞行情况下发生冲突,本文在参考国内外文献的基础上^[7,10],将蚁群算法的原理运用于航向的选择,简化问题如下:

(1) 民用飞机一般除了在起飞和降落阶段外,都是在固定的高度飞行。考虑到乘客舒适性和燃油消耗,飞机之间在改变航路以避免冲突时,一般只在同一高度层改变航向,因此,把三维立体问题简化为二维平面问题。

(2) 民用飞机在巡航段正常飞行时,一般没有大角度转弯或转向飞行。当探测到将有飞行冲突发生时,固定的航向改变使飞行员更好地执行冲突解决方案,并且减少反应时间。这里假设飞机在飞行中任何位置只能选择三个飞行方向,即保持原有航向、左转 30° 、右转 30° 。

(3) 民用飞机,特别是民用运输机是一种非高机动性能飞机,在进行比较稳定的巡航飞行时可以考虑为匀速直线运动,因此,假设飞机在飞行过程中速度保持不变。

(4) 每架飞机在扇区内都有一个理论上的进入和退出点。当每架飞机不作任何航向调整时,每一步的航迹计算点定义为每一步的理论退出点。

(5) 任何时候如果两架飞机间隔少于最小安全间隔 δ ,它们处于冲突状态,必须进行航向调整,最优化的目的为飞机的总延误(即所有飞机终了时刻实际位置与理论退出点的距离的总和)最小。

有了以上的前提条件,我们便得到以下的数学模型:

假设有 n 架飞机进入扇区后,在某些位置将发生飞行冲突,因此,不能按照理论上最短的直线航路飞行,必须改变航向以避免冲突发生。算法中把飞机在扇区内的飞行时间分为 q 步,步长相应于安全间隔 δ 为足够小,以保证及时探测到飞行冲突的发生。在每

一步中, 飞机保持一定的航向。在进入每一步之前, 飞机的航向能作出一定调整 (具体见前提 2、3)。这样在解空间内有三个变量, 将其表示为: 0 表示飞机保持原航向, -1 表示飞机左转 30°, +1 表示右转 30°。由于这里分 q 步, 扇区内 n 架飞机航向的改变被看做 $n \times q$ 个解空间为 -1、0、+1 的变量, 记为 $x_1, x_2, \dots, x_{n \times q}$ 。目标函数如下:

$$y = \min \sum_{i=1}^n S_i \quad (1)$$

式中, n 为扇区内飞机数; S_i 为第 i 架在本扇区内的延误。

约束条件为扇区内任意两架飞机之间满足条件:

$$d = \sqrt{(a'_i - a'_j)^2 + (b'_i - b'_j)^2} > \delta \quad (2)$$

式中, $(a'_i, b'_i), (a'_j, b'_j)$ 为任意 t 时刻第 i 架和第 j 架飞机在该水平面内的坐标, 通过 $x_1, x_2, \dots, x_{n \times q}$ 的取值计算得出。第 i 架飞机 a'_i, b'_i 的计算公式为:

$$\begin{cases} a'_i = a_i^{t-1} + q \cdot \cos(-30x_{(i-1)q+t}) \\ b'_i = b_i^{t-1} + q \cdot \sin(-30x_{(i-1)q+t}) \end{cases} \quad (3)$$

实际上, 这是一个带约束的非线性整数规划问题, 它符合以下特征:

$$\begin{aligned} \min f(x_1, x_2, \dots, x_n) & \quad (4) \\ \text{s.t. } g(x_1, x_2, \dots, x_n) & \geq 0 \\ a_i & \leq x_i \leq b_i \quad (i = 1, 2, \dots, n) \\ x_i & \in Z \quad (i = 1, 2, \dots, n) \end{aligned}$$

式中, Z 为整数空间; a_i, b_i 为整数。

带约束非线性整数规划属于组合优化问题, 也是 NP-complete 问题, 蚁群优化算法为解决它提供了新思路。在离散的求解空间中, 群智能表现良好的搜索效果。下面我们将应用蚁群算法解决以上问题。

2 蚁群算法在飞行冲突解决中的应用

在上述表述的基础上, 我们可以看到, 可行解的空间如图 1 所示, x_i 有 3 个节点, 每个变量取一个值就构成空间一个解。如图 1 中的解为:

$$(x_1, x_2, x_3, x_4, x_5, x_6, \dots, x_n) = (0, 0, +1, -1, 0, +1, \dots, -1)$$

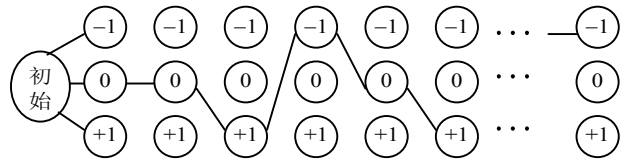


图 1 解空间

Fig. 1 The solution space

选取 n 个变量变成 n 级决策问题。开始 m 个蚂蚁在初始位置, 第 j 级中选择第 i 个节点的概率 p_{ij} 为:

$$p_{ij} = \frac{\tau_{ij}}{\sum_{i=1}^3 \tau_{ij}} \quad (5)$$

τ_{ij} 理解为第 j 级中第 i 个节点的吸引强度。更新方程为:

$$\tau_{ij}^{\text{new}} = \rho \tau_{ij}^{\text{old}} + \frac{Q}{f} \quad (6)$$

式中, ρ 表示强度的衰减系数, 一般取 0.5~0.9; Q 为一正常数; f 为目标函数值。

一条路径对应一个解, 但这些解并不都是可行解。只有在经过冲突探测后, 确定无飞行冲突发生的解才是可行解。下面蚁群算法所搜索到的解均为可行解。蚁群算法初始化时, 各路径信息素取相同值, 让蚂蚁以等概率选择路径, 这样使蚂蚁很难在短时间从大量杂乱无章的路径中, 找出一条比较好的路径, 收敛速度较慢。假如初始化时就给出启发性的信息量, 可以加快收敛速度。改进的方法是, 产生大量的路径 (如 200 条), 从中选择比较优的 (如 30 条), 使这些路径留下信息素 (与路径长度和成反比), 各路径的信息量不同, 以此引导蚂蚁进行路径选择。

蚂蚁每次周游结束后, 不论蚂蚁搜索到的解如何, 都将赋予相应的信息增量, 比较差的解也将留下信息素, 这样干扰后续的蚂蚁进行寻优, 造成大量的无效搜索。改进的方法是, 所有路径弧段的信息量将以 ρ 的倍数挥发一部分, 而算法到目前为止所发现的最好可行解上的信息素得以增强。

改进后的蚁群算法步骤如下:

步骤 1 $nc = 0$ (nc 为循环次数), 给 τ_{ij} 赋相同的数值, 产生大量路径, 从中选择比较优的, 使这些

路径留下信息素, 给出 Q 、 ρ 的值;

步骤 2 将 m 个蚂蚁置于初始位置;

步骤 3 对每个蚂蚁按转移概率 p_{ij} 选择该级中一个节点, 每个蚂蚁走遍 n 个节点;

步骤 4 根据约束条件判断该解是否为可行解。若是可行解, 则计算目标函数值 f ; 否则转 3。

步骤 5 按算法到目前为止所发现的最好可行解的路径修改吸引强度, $nc \leftarrow nc + 1$;

步骤 6 若 $nc >$ 规定的循环次数, 停止运行, 根据 τ_{ij} 选择节点 (若 $\tau_{i_0j} = \max_i \tau_{ij}$, 则第 j 级中选择第 i_0 个节点); 否则转 2。

3 仿真实验

本文解决的是扇区中三架飞机存在的严重冲突问题。如图 2 所示, 把飞机在扇区内总的飞行时间分

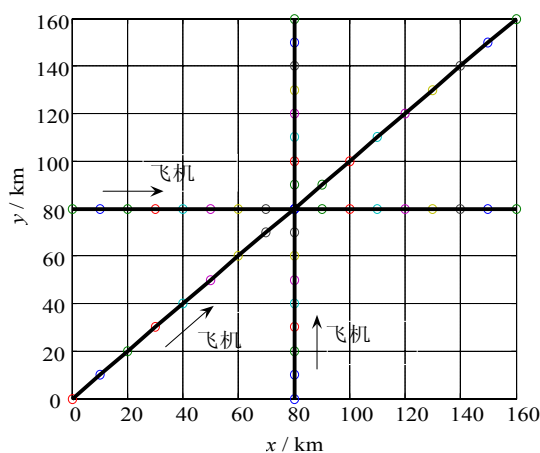


图 2 冲突解决前航迹

Fig. 2 Aircraft tracks before conflict resolution

为 16 步。设每一步步长为 10, $\delta = 20$ 。三架飞机于 $t = 0$ 时刻匀速直线飞行通过 $160 \text{ km} \times 160 \text{ km}$ 的扇区, 其中第一架飞机从西向东飞行, 另一架从南向北飞行, 第三架从西南向东北方向飞行, 同时交叉通过同一扇区, 于第 8 步时发生冲突, 各个冲突之间互相制约。用蚁群算法对航路进行了调整, 图 2 显示的是冲突解决前的航路, 图 3 是调整后的航路。从图 3 中

可以看出, 飞行冲突现象已经得到解决。

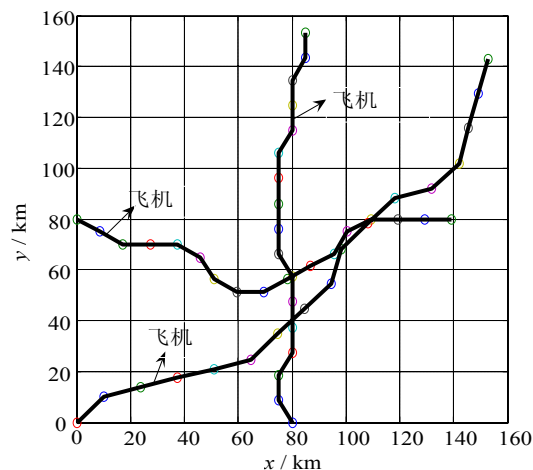


图 3 冲突解决后航迹

Fig. 3 Aircraft tracks after conflict resolution

用蚁群算法进行仿真计算时, 蚂蚁数目 $m = 50$, $nc_{\max} = 100$, $Q = 1000$, $\tau_{ij} = 100$, ρ 取不同的数值测试, 算法各运行 50 次, 结果显示 $\rho = 0.8$ 时效果最好。这是因为: ρ 太小, $\rho \tau_{\text{old}}$ 较小, 以前的激素影响力太小, 体现不出蚁群算法的优点; ρ 太大, $\rho \tau_{\text{old}}$ 较大, 表示以前的激素影响力较大, 体现不出蚂蚁运动的影响。 $\rho = 0.8$, $nc_{\max} = 100$, $Q = 1000$, $\tau_{ij} = 100$, m 取不同的数值测试, 结果显示蚂蚁数目越大, 效果越好。这是因为随着蚂蚁数目的增加, 搜索范围变大, 不过收敛速度慢运行时间增加了。

4 结束语

本文论述了蚁群算法在空中交通飞行冲突解决中的应用。蚁群算法是一种最新发展的仿生优化算法, 在解决许多复杂优化问题方面已经展现出优异的性能和巨大的发展潜力。仿真实验表明, 改进后的蚁群算法可以有效解决飞行冲突。由于本文所建立的数学模型是一个带约束的非线性整数规划问题, 在某种意义上也是对该类组合优化问题做了初步的研究。

- 计站长计算初探[J]. 土木工程学报, 2003, 36(7): 72-77.
- [9] 钱颂迪. 运筹学[M]. 北京: 清华大学出版社, 2004.
- [10] Awan Javy, Solomon Norman et al. Highway capacity Manual 2000[M]. The TRB National Committee on Highway Capacity and Quality of Service, 2000.
- [11] GB 50220-95. 城市道路交通规划设计规范[S]. 北京: 中华人民共和国建设部, 1995.
- [12] 沈景炎. 乘客动态分布与站台宽度的研究[J]. 城市轨道交通研究, 2001, (1): 21-25.
- [13] 宋年秀, 王耀斌, 骆勇. 运输枢纽与场站设计[M]. 北京: 机械工业出版社, 2006.

(中文编辑: 吴继屏)

上接第 119 页

参考文献

- [1] Alliot J-M., Durand N., and Granger G. FACES: A free flight autonomous and coordinated embarked solver [J]. Air Traffic Control Quarterly, 2000, 8: 109-130.
- [2] Durand N., Alliot J. and Noailles J. Automatic aircraft conflict resolution using genetic algorithms [C]. In Proc. 11th Annual ACM Conference on Applied Computing. Reading: 1996, 289-298.
- [3] Pallottino L., Feron E. M., and Bicchi A. Conflict resolution problems for air traffic management systems solved with mixed integer programming [J]. IEEE Transactions on Intelligent Transportation Systems, 2002, 3(1): 3-11.
- [4] 刘星, 韩松臣. 用于自由飞行冲突探测的 Delaunay 方法[J]. 数据采集与处理, 2002, 17(4): 446-449.
- [5] Kuchar J. K. and Yang L. C. A review of conflict detection and resolution modeling methods [J]. IEEE Trans. on Intelligent Transportation Systems, 2000, 1(4): 179-189.
- [6] Colorni A., Dorigo M., Maniezzo V. Distributed optimization by ant colonies [C]. Proc. European Conference on Artificial Life, Reading. 1991, 134-142.
- [7] 刘星, 胡明华等. 遗传算法在飞行冲突探测解脱中的应用[J]. 南京航空航天大学学报, 2002, 34(1): 35-39.
- [8] Stützle T., Dorigo M. A short convergence proof for a class of ant colony optimization algorithms [J]. IEEE Trans. on Evolutionary Computation, 2002, 6(4): 358-365.
- [9] 高尚. 蚁群算法理论、应用及其与其它算法的混合[D]. 南京: 南京理工大学, 2005: 6-14.
- [10] Delahaye Daniel, Alliot J. M., Scheenauer Marc, et al. Genetic algorithms for partitioning airspace [C]. In Proceedings of the Tenth Conference on Artificial Intelligence Application. IEEE, 1994.

(中文编辑: 吴继屏)