

航班降落调度的设计与实现

(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 航班着陆调度(Aircraft Landing Scheduling, ALS)旨在为待降落的航班给出有效的着陆调度方案, 保证每架航班能够安全地依次着陆。研究航班着陆调度问题对确保飞行安全及提高飞行效益具有重大的意义。在解决航班着陆调度问题方面, 群体智能优化算法不仅具有强大的全局搜索能力, 而且能够处理非线性的复杂约束及目标函数, 用智能算法解决航班着陆调度问题成为一个研究热点。本文主要结合航班降落调度的具体实例, 设计并实现了基于群体智能优化算法的航班降落调度。主要包括基于粒子群算法的航班降落调度设计与实现、基于遗传算法的航班降落调度设计与实现。其中, 采用粒子群算法得到的优化结果为 700, 航班的降落时间为[165 258 98 106 118 126 134 142 150 180]; 采用遗传算法得到的最好优化结果为 710, 对应的航班降落时间为[166 258 98 106 119 135 127 143 151 181]。

关键词: 航班降落调度; 粒子群算法; 遗传算法

Design and Implementation of Flight Landing Scheduling

Wu Yong

(College of electronic and information engineering, Xi'an Jiaotong University, 710049, Xi'an)

Abstract: Aircraft landing scheduling (ALS) aims to provide an effective landing scheduling scheme for the flights to be landed, to ensure that each flight can land safely in turn. It is of great significance to study flight landing scheduling problem to ensure flight safety and improve flight efficiency. In order to solve the problem of flight landing scheduling, swarm intelligence optimization algorithm not only has a strong global search ability, but also can deal with nonlinear complex constraints and objective functions. In this paper, we design and implement the flight landing scheduling based on swarm intelligence optimization algorithm. It mainly includes the design and implementation of flight landing scheduling based on particle swarm optimization algorithm, and the design and implementation of flight landing scheduling based on genetic algorithm. Among them, the optimal result obtained by particle swarm optimization is 700, and the flight landing time is [165 258 98 106 118 126 134 142 150 180]; the optimal result obtained by genetic algorithm is 710, and the corresponding flight landing time is [166 258 98 106 119 135 127 143 151 181].

Keywords: aircraft landing scheduling; particle swarm optimization; genetic algorithm

随着世界经济的发展,对航空运输的需求也不断增长,这导致飞机数量和飞行流量急剧增加,空中交通拥堵已成为亟待解决的一个难题。每年因空中交通拥挤造成的经济损失已经达到了上百亿美元,更重要的是空中交通拥挤严重影响了空中飞行的安全^[1]。空中交通流量管理的研究也受到了国内外广大学者的广泛注意。面对日益增长的航运需求,航班数量也将相应得到增加,未来 20 年内,中国民航需要补充民用客机三千多架,中国机队规模将扩大到当前的 3 倍。这必然又会反馈到航运总量上,使得航运总量继续增加,导致空中运输量的高速增长。现在机场所使用的交通管理方式已经不能满足快速增长的交通运输需求,落后的交通管理和调度方式会使得终端区调度的瓶颈地位更加明显,从而导致大量航班的延误,机场的更加拥塞,机场运营成本和航班飞行成本的持续提高。随着空中交通运输量的增加,我国原来的空中交通流量管理方法已经不能满足于日益增长的交通量的需求。空中交通流量管理可以分为先期流量管理、终端区交通流量管理以及具有寻径能力的空中流量管理。先期流量管理是在宏观上对交通流量进行管理和调度,主要研究地面等待策略问题;终端区交通流量管理主要是指在本终端区的范围内进行飞行前流量控制和实时交通管理,主要研究终端区飞机的着陆排序问题;具有寻径能力的空中流量管理主要根据各个机场、区域以及航线容量的实时情况和信息及时动态调整各个航班的起飞时间、航路变化和着陆的时间。

航班降落调度是机场管理的重要组成部分,旨在为待着陆的航班安排合理的降落调度方案,保证机场的秩序,减少早到或者晚到造成的经济损失。因此研究航班的调度问题对提高机场的运行效率以及飞行效益具有重大意义。航班调度问题是典型的 NP

问题,在我国航空业发展初期,主要通过运筹学中的线性规划来进行优化。但随着我国航空业的不断发展,航班调度问题的规模逐渐增大,数学规划方法很难给出 NP 问题的最优解。现如今,一些相关研究人员将智能优化算法应用到解决航班调度的问题上来,不再纠结去寻找问题的最优解,而是将目标转移到在合理的时间内寻找到一个可行的,接近最优解的近似解。

航班延误问题已成为困扰航空公司和广大旅客的难题,对此,越来越多的国内外学者投身其中,主要研究导致航班延误的原因和探讨解决问题的优化算法,以减少航班延误,提高航空系统容量,增加飞行安全。2003 年, Kari Anderson 提出了改进空中交通流量的决策工具,对协同进场规划作了基于优化的方法分析,证明了进场排序程序的有效性^[2]; 2006 年 Aditya 利用组合优化方法解决拥挤机场航班的优化问题^[3]; 2008 年 Hanbong 研究了航班排序过程中燃油成本、延误成本等的权重问题^[4]; 2009 年张洪海、胡明华对着陆航班排序问题建立了包括延误损失、延误公平性和管制员负荷的多目标优化模型^[5]; 2009 年曹力等提出了基于机型间隔的机场终端区流量优化模型^[6]; 2013 年张子阳等提出了基于遗传算法的航班着陆调度方法^[7]; 2019 年王璐提出了基于遗传算法的平行跑道航班着陆调度方法^[8]。

本文本文主要结合航班降落调度的具体实例,设计并实现了基于群体智能优化算法的航班降落调度。主要包括基于粒子群算法的航班降落调度设计与实现、基于遗传算法的航班降落调度设计与实现。

1 问题描述及模型建立

1.1 问题描述

随着世界航空运输的迅猛发展,机场终端的航班调度自动化程度是一个非常重要的问题。当一座机场短时间内抵达大量的飞

机，由于机场本身的跑道数量以及机场客运量等等因素的限制，使得机场必须对抵达的飞机进行降落顺序以及时间的安排，对于不同的机型，提前降落与延后降落都会造成一定的额外成本，相关的成本大致满足下图所示的关系：

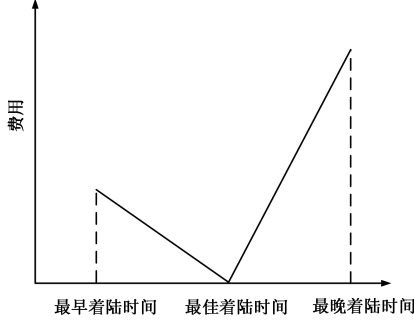


图1 航班降落成本示意图

航班降落调度问题可以描述为：机场在某一段时间内有 D 架需要降落航班 $PLANES$ ，每个航班都有一个最早到达时间 $START_i$ 和最晚到达时间 $STOP_i$ ，在这个时间窗口内，航空公司需要选择一个目标时间 $TARGET_i$ ，并将它作为航班到达时间公布出去，如果比此时间迟到或早到，会带来额外的费用支出，每个航班都定义了早到每分钟的惩罚 $CEARLY_i$ 和晚到每分钟的惩罚 $CLATE_i$ ，同时，在两个航班降落之间需要有一段安全时间间隔 $DIST_{i,j}$ ，其中 $i, j \in PLANES$ 。

1.2 数学模型

(1)定义降落方案变量

$$LAND = \{land_1, land_2, \dots, land_D\}$$

其中 $land_i$ 代表航班 i 的降落时间，根据实际条件，这些变量一般取整数，且应满足最早和最晚时间的约束。因此，可定义约束条件为：

$$START_i \leq land_i \leq STOP_i, \forall i \in PLANES$$

$$land_i \in \mathbb{Z}, \forall i \in PLANES$$

(2)任意两个航班 i, j 之间的实际降落时间间隔要大于最短间隔 $dist_{i,j}$ ，因此，可

定义约束为：

$$|land_i - land_j| \geq dist_{i,j}, \forall i, j \in PLANES$$

(3)定义该降落方案中早到的航班集合

$EPLAN$ 和晚到的航班集合 $LPLAN$ 为：

$$EPLAN = \{n_i \in PLANES \mid TX_{n_i} \leq TA_{n_i}\}$$

$$LPLAN = \{m_j \in PLANES \mid TX_{m_j} \geq TA_{m_j}\}$$

(4)由于航班早到和晚到都会带来额外的惩罚，所以定义该降落方案的总惩罚函数：

$$\min f =$$

$$\sum_{i \in EPLAN} (TARGET_i - land_i) \times CEARLY_i + \sum_{j \in LPLAN} (land_j - TARGET_j) \times CLATE_j \quad (1)$$

1.3 实例数据

机场某一段时间内陆续抵达 10 班航班，其相关时刻已经转为整数形式，即不使用时分秒的格式，但依旧满足数值递增表示时间增加的基本要求。下表给出 10 班飞机的最早，最佳，最迟降落时刻以及提前和延后每分钟的成本参数：

表1 航班的时间窗口和早到与晚到惩罚

航班	ST	TA	SP	CE	CL
1	129	155	559	10	10
2	195	258	744	10	10
3	89	98	510	30	30
4	96	106	521	30	30
5	110	123	555	30	30
6	120	135	576	30	30
7	124	138	577	30	30
8	126	140	573	30	30
9	135	150	591	30	30
10	160	180	657	30	30

这 10 班航班由于机型不同，每班航班降落之后都需要地面工作人员进行乘客分流等工作，因此需要在相临航班之间预留时间窗口，下表给出每班航班与其他航班相临

降落时需要的时间窗口：

表 2 航班降落时间间隔

	1	2	3	4	5	7	8	9	10
1	0	3	15	15	15	15	15	15	15
2	3	0	15	15	15	15	15	15	15
3	15	15	0	8	8	8	8	8	8
4	15	15	8	0	8	8	8	8	8
5	15	15	8	8	0	8	8	8	8
6	15	15	8	8	8	8	8	8	8
7	15	15	8	8	8	0	8	8	8
8	15	15	8	8	8	8	0	8	8
9	15	15	8	8	8	8	8	0	8
10	15	15	8	8	8	8	8	8	0

例如航班 5 和航班 8 相临降落时需要至少间隔 8 个时间单位。

2 基于粒子群算法的航班降落调度设计与实现

2.1 粒子群算法简介

粒子群算法 (Particle Swarm Optimization, PSO) 最早是由 Eberhart 和 Kennedy 于 1995 年提出^[10]，它的基本概念源于对鸟群觅食行为的研究。设想这样一个场景：一群鸟在随机搜寻食物，在这个区域里只有一块食物，所有的鸟都不知道食物在哪里，但是它们知道当前的位置离食物还有多远。最简单有效的策略寻找鸟群中离食物最近的个体来进行搜索。PSO 算法就从这种生物种群行为特性中得到启发并用于求解优化问题。

用一种粒子来模拟上述的鸟类个体，每个粒子可视为 N 维搜索空间中的一个搜索个体，粒子的当前位置即为对应优化问题的一个候选解，粒子的飞行过程即为该个体的搜索过程。粒子的飞行速度可根据粒子历史

最优位置和种群历史最优位置进行动态调整。粒子仅具有两个属性：速度和位置，速度代表移动的快慢，位置代表移动的方向。每个粒子单独搜寻的最优解叫做个体极值，粒子群中最优的个体极值作为当前全局最优解。不断迭代，更新速度和位置。最终得到满足终止条件的最优解。

在 D 维空间中，有 N 个粒子，粒子的适应度函数为被优化的目标函数，粒子 i 第 k 次迭代所处的位置：

$$\mathbf{x}_i^k = (x_{i1}^k, x_{i2}^k, \dots, x_{iD}^k) \quad (2)$$

粒子 i 第 k 次迭代时的速度：

$$\mathbf{v}_i^k = (v_{i1}^k, v_{i2}^k, \dots, v_{iD}^k) \quad (3)$$

粒子 i 前 k 次搜索到的最佳位置：

$$\mathbf{pb}_i^k = (pb_{i1}^k, pb_{i2}^k, \dots, pb_{iD}^k) \quad (4)$$

粒子群前 k 次搜索到的最佳位置：

$$\mathbf{gb}_i^k = (gb_{i1}^k, gb_{i2}^k, \dots, gb_{iD}^k) \quad (5)$$

粒子的位置和速度更新方式：

$$\begin{aligned} \mathbf{v}_{id}^{k+1} &= w\mathbf{v}_{id}^k + c_1r_1(\mathbf{pb}_{id}^k - \mathbf{x}_{id}^k) + c_2r_2(\mathbf{gb}_d^k - \mathbf{x}_{id}^k) \\ \mathbf{x}_{id}^{k+1} &= \mathbf{x}_{id}^k + \mathbf{v}_{id}^{k+1} \end{aligned} \quad (6)$$

其中， w 为惯性权重，非负实数，调节对解空间的搜索范围； c_1, c_2 为加速度常数，非负实数，调节学习最大步长； r_1, r_2 为 $[0,1]$ 上的随机数，增加搜索随机性。

在迭代中，若 \mathbf{x}_{id}^k 和 \mathbf{v}_{id}^k 超出了边界，则取边界值：

$$\mathbf{x}_{id}^k \in [\mathbf{x}_{\min,d}, \mathbf{x}_{\max,d}]$$

$$\mathbf{v}_{id}^k \in [-\mathbf{v}_{m,d}, \mathbf{v}_{m,d}]$$

粒子群优化算法流程图如下图：

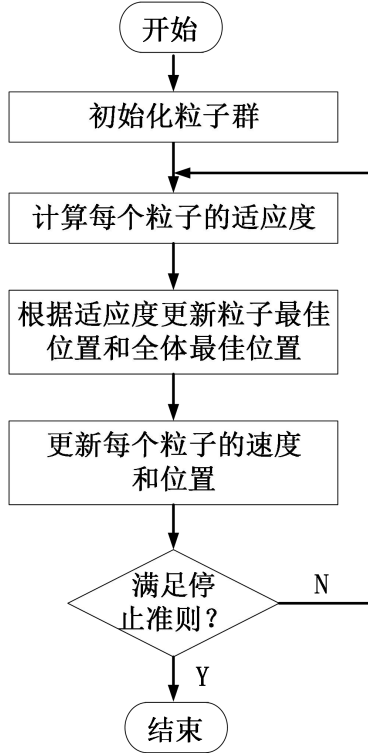


图2 粒子群优化流程图

2.2 粒子群算法设计

在航班降落调度实际问题中，一共有 10 架飞机，那么每个粒子的维度为 10。由于每架飞机的最早着陆时间、最晚着陆时间不相同，因此每个维度的位置范围也不同，10 个维度的位置下边界 x_min 为：

$$[129, 195, 89, 96, 110, 120, 124, 126, 135, 160] \quad (7)$$

上边界 x_max 为：

$$[559, 744, 510, 521, 555, 576, 577, 573, 591, 657] \quad (8)$$

因此，位置的边界值为：

$$x_{id}^k \in [x_min(d), x_max(d)], \quad d = 1, 2, \dots, 10 \quad (9)$$

由于速度的范围与位置有关，因此 10 个维度的速度范围也不相同，将最大速度设为每维变量变化范围的 $\alpha \in (0, 1)$ 倍，则每维的速度范围为：

$$v_{id}^k \in [-\alpha(x_max(d) - x_min(d)), \alpha(x_max(d) - x_min(d))], \quad d = 1, 2, \dots, 10 \quad (10)$$

粒子的适应度函数设计方面，适应度更新的原则往目标函数值降落的方向优化，而又要考虑到两架飞机降落的时间间隔要大于规定的时间间隔，在这里，对不符合时间间隔要求的粒子进行筛选是通过将其适应度设置为正无穷来实现的。也即，当粒子符合时间间隔要求时，粒子的目标函数为式 (1)，当其不符合间隔要求时，目标函数值设置为正无穷。

2.3 粒子群算法实现

将粒子群算法与飞机降落调度实际数据结合，通过 MATLAB 编程实现，经过多次调参，结果最优的参数为：

$$\alpha = 0.2, w = 0.4, c_1 = 0.5, c_2 = 2$$

其中 α 为速度范围系数。

在此参数下，航班降落调度结果如下图所示（运行代码 PSO1.m 可得）：

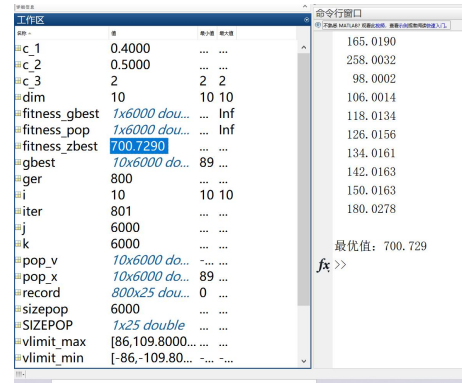


图3 粒子群优化结果

结果为小数是因为未对变量限制为正整数，在此小数结果的基础上，对 10 架航班的降落时间取整，并对成本重新计算，可得到航班降落时间分别为：

$[165, 258, 98, 106, 118, 126, 134, 142, 150, 180]$
此时的成本损失为 700。

此时的粒子群收敛迭代过程如下图所示：

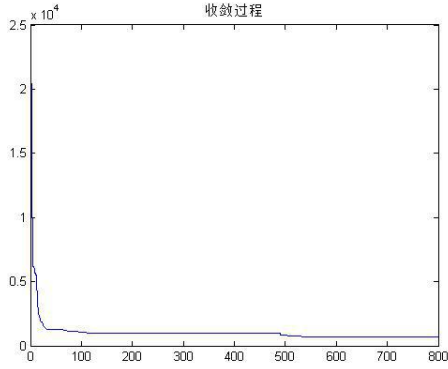


图4 粒子群算法优化收敛过程

可以发现，迭代大约 500 次后，优化结果达到 700 左右后基本不再变化。

2.4 粒子群算法分析

2.4.1 粒子数量的影响

为研究粒子数量对算法收敛的影响，将粒子数量从 100 开始，以 400 为间隔，设置了 25 个不同的值，也即粒子数量分别为 100,500,900,...,10000,其他参数与最优参数一致，得到优化结果（各个粒子数量下迭代的最优值）随粒子数量变化的曲线图：

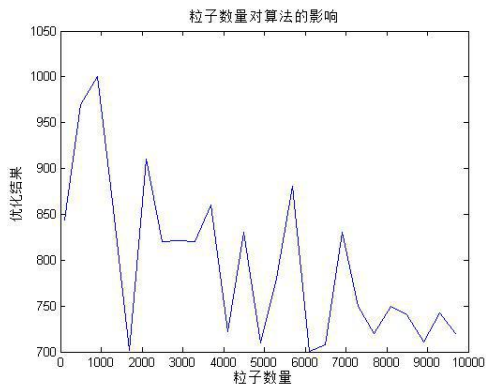


图5 粒子数量对算法的影响

可以发现，随着粒子数目的增加，目标函数值也越来越小，优化结果越来越好，但当粒子数达到一定值后，优化的结果基本在一条水平线上下波动（图中为 725 左右），说明随着粒子数量的增加，算法的收敛效果越来越好，但达到一定值后，粒子数量的增加不会有显著的效果提升。

2.4.2 最大速度的影响

为研究最大速度对算法收敛结果的影响，将速度范围系数 α 分别设置为如下数值：

$$\alpha \in [0.01, 0.05, 0.1, 0.15, 0.2, 0.5, 0.8, 1]$$

得到优化结果（各个速度范围系数下迭代的最优值）随粒子数量变化的曲线图：

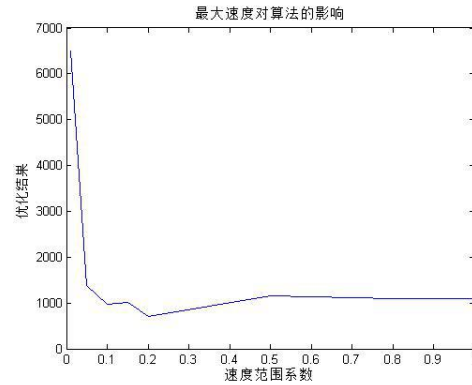


图6 最大速度对算法的影响

由图可以看出，在速度范围系数处于 $[0.15, 0.3]$ 区间时，算法收敛结果较好。当速度过大时，粒子容易飞过最优解；当速度过小时，容易陷入局部最优。其中，在本实验中，当 $\alpha = 0.2$ 时，可以得到最好的优化结果，此时 10 个维度的速度上界 v_{\max} 为：

$$[86, 109.8, 84.2, 85, 89, 91.2, 90.6, 89.4, 91.2, 99.4]$$

速度下界 v_{\min} 为：

$$[-86, -109.8, -84.2, -85, -89, -91.2, -90.6, -89.4, -91.2, -99.4]$$

2.4.3 惯性因子的影响

为研究惯性因子对算法收敛结果的影响，将惯性因子 w 分别设置为从 0 开始，以 0.1 为间隔，设置了 10 个不同的值，也即惯性因子分别为 0, 0.1, 0.2, ..., 1, 其他参数与最优参数一致，得到优化结果（各个不同惯性因子下迭代的最优值）随惯性因子变化的曲线图：

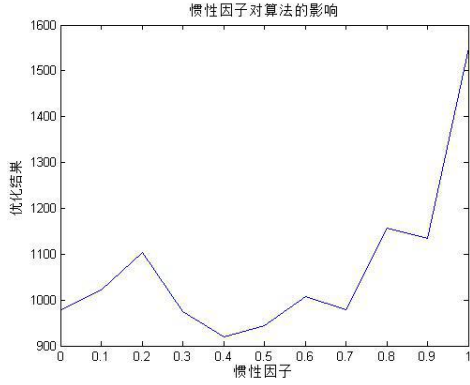


图 7 惯性因子对算法的影响

由图可以看出，当惯性因子约为 0.4 时，优化效果最好。这是因为惯性因子体现的是粒子继承上一次迭代速度的能力。较大的一个惯性权重因子有利于全局的搜索，而一个小的惯性权重因子则更加利于局部的搜索，为了让粒子在搜索过程中权衡局部与全局的搜索，惯性因子不可太大，也不可太小。在本次实例中，与最晚到达时间相比，航班的目标时间更接近最早到达时间，而优化目的是在满足时间间隔的基础上，尽量往目标时间靠近，这样成本才最小，因此在此实例中取得最优值的惯性因子较小。

2.4.4 个体学习因子的影响

为研究个体学习因子对算法收敛结果的影响，将个体学习因子 c_1 分别设置为 0,0.4,0.5,1,2 五种不同的数值，其他参数与最优参数一致，得到优化结果（各个不同学习因子下迭代的最优值）随个体学习因子变化表：

表 3 个体学习因子与结果列表

因子	0	0.4	0.5	1	2
结果	1006	700	820	982	919

由表可知，个体学习因子在适当大小时，可得到令人满意的优化结果，过大或过小均无法得到最优值。当学习因子为 0 时，粒子群算法没有自我，迭代过程中不再考虑个体最优，只考虑群体最优，这样的粒子群易陷入局部最优而无法跳出。

2.4.5 社会学习因子的影响

为研究社会学习因子对算法收敛结果的影响，将社会学习因子 c_2 分别设置为 0,0.4,0.5,1,2 五种不同的数值，其他参数与最优参数一致，得到优化结果（各个不同学习因子下迭代的最优值）随社会学习因子变化表：

表 4 社会学习因子与结果列表

因子	0	0.4	0.5	1	2
结果	1813	1600	1102	921	700

由表可知，社会学习因子在适当大小时，可得到令人满意的优化结果。当学习因子为 0 时，算法收敛过程如下图：

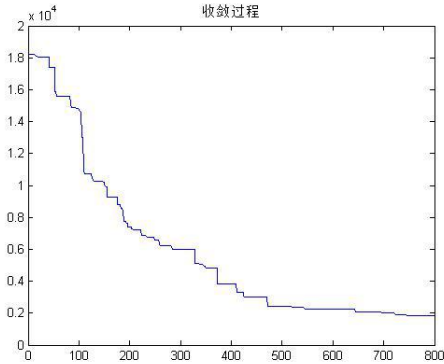


图 8 社会学习因子为 0 时收敛情况

与图 4 对比可知，此时的粒子群算法收敛速度变的极其缓慢，这是因为此时的粒子群算法没有考虑社会属性，迭代过程中仅仅考虑个体最优，不考虑群体最优，完全没有社会共享信息，导致收敛速度缓慢。

3 基于遗传算法的航班降落调度的设计与实现

3.1 遗传算法简介

遗传算法 (Genetic Algorithm, GA) 最早是由美国的 John holland 于 20 世纪 70 年代提出^[11]，该算法是根据大自然中生物体进化规律而设计提出的。是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过

程的计算模型,是一种通过模拟自然进化过程搜索最优解的方法。该算法通过数学的方式,利用计算机仿真运算,将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。在求解较为复杂的组合优化问题时,相对一些常规的优化算法,通常能够较快地获得较好的优化结果。

遗传操作是模拟自然界中生物进化过程中的繁殖、杂交和突变等现象。在遗传算法中,对个体进行编码,组成初始群体。遗传操作的目的是对群体的个体按照其对环境的适应度采取一定的操作,从而实现优胜劣汰的进化过程。遗传操作包括以下 3 个主要步骤:

(1)选择。从群体中根据个体适应度大小,依照选择概率,选择优胜的个体,淘汰劣质个体的操作叫做选择。选择的目的是把优化的个体直接遗传到下一代。目前常用的选择算子有以下几种:轮盘赌选择法、最优个体选择法、期望值法、适应度比例方法、随机遍历抽样法、排挤法^[8]。

(2)交叉。遗传算法中起核心作用的是交叉算子。所谓交叉是指把两个个体的部分结构加以重组而生成新个体的操作。通过交叉操作,遗传算法的搜索能力得以飞跃似的提高。交叉算子根据交叉概率将种群中的两个个体随机地交换某些基因,能够产生新的基因组合,将有益基因组合在一起。根据编码表示方法的不同,可以分为实值重组和二进制交叉,其中二进制交叉又包含单点交叉、多点交叉、均匀交叉等。

(3)变异。变异算子是依照变异概率随机改变一些个体的某些基因值。依据编码方法的不同,可以分为实值变异和二进制变异。遗传算法引入变异目的有两个:一是使遗传算法具有局部的随机搜索能力。当遗传算法通过交叉算子已接近最优解邻域时,利用变异算子的这种局部随机搜索能力可以加速

向最优解收敛。显然,此种情况下的变异概率应取较小值,否则接近最优解的积木块会因变异而遭到破坏。二是使遗传算法可维持群体多样性,以防止出现未成熟收敛现象。此时收敛概率应取较大值。变异概率的选取一般受种群大小、染色体长度等因素的影响,通常选取很小的值,一般取 0.001~0.1。

遗传算法流程图如下图所示:

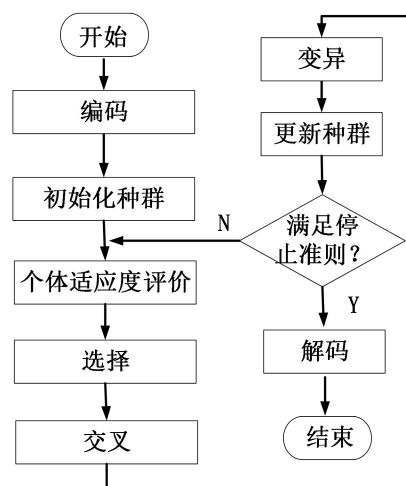


图 9 遗传算法流程图

3.2 遗传算法设计

为了加快收敛速度,且取得更好的优化结果,本次遗传算法设计并未直接使用传统的遗传算法,而是先利用“先到先服务”的思想粗粒度地得到符合时间间隔要求的 10 架航班降落的大致时间,再利用遗传算法细粒度地优化,得到更好的结果,且能大大缩减迭代时间。

先到先服务^[1]是一种典型的动态贪心排序算法,基本思想类似于贪心算法。基于本次航班降落调度的实际数据,先到先服务的步骤为:

根据这 10 架航班的目标到达时间从小到大排序,得到航班序列:

$$(a_1, a_2, \dots, a_{10})$$

基于先到先服务的思想,排在第一位的航班先到达,并且考虑到贪心算法应该让第

一个航班的延迟最小，所以让第一架航班在目标时间准时到达，即

$$REACH_{a_1} = TARGET_{a_1}$$

考虑第二架航班的实际到达时间时，应综合其目标到达时间和其与第一架航班的安全时间间隔，故得到第二架航班的实际到达时间为：

$$REACH_{a_2} = \max(TARGET_{a_2}, REACH_{a_1} + DIST_{a_1, a_2})$$

以此类推，可以得到其他航班的实际到达时间为：

$$REACH_{a_i} = \max(TARGET_{a_i}, \max(REACH_{a_j} + DIST_{a_j, a_i}))$$

根据航班降落调度问题的特点，采用整数序号排序，用航班的实际到达顺序表示染色体基因。例如染色体

$$(6, 1, 5, 9, 2, 3, 4, 8, 7, 10)$$

表示序号为 1 的航班第 2 个降落，序号为 6 的航班第 1 个降落，以此类推。这样可以得到初始化种群。

适应度评价（目标函数）采用式(1)，在计算过程中，需要得到实际降落时间，基于染色体编码，利用“先到先服务”的思想，可以得到该染色体下的实际到达时间解码状态，继而得到目标函数值。选择方法采取确定性选择法，即种群数量一定，适应度约大的个体，下一代的数量越大，采取该选择方法的原因是“先到先服务”已经粗粒度地得到符合时间间隔要求的 10 架航班降落的大致时间。

对于交叉过程，需要考虑到交叉后染色体有重复基因的情况，此时需要采取映射的方法对重复基因^[9]进行消除。例如，对于交叉过程：

$$\begin{array}{ccc} 1\ 5\ 8\ 3\ 7\ 9\ 6\ 2\ 4 & \xrightarrow{\text{交叉}} & 1\ 5\ 9\ 6\ 3\ 2\ 6\ 2\ 4 \\ 1\ 4\ 9\ 6\ 3\ 2\ 7\ 5\ 8 & & 1\ 4\ 8\ 3\ 7\ 9\ 7\ 5\ 8 \end{array}$$

交叉后，同一个个体有相同的编号，以重复数字 2 为例，采取下图所示的方法消除重复基因：

$$\begin{array}{ccc} 1\ 5\ 9\ 6\ 3\ 2\ 6\ 2\ 4 & \rightarrow & 1\ 5\ 9\ 6\ 3\ 2\ 6\ 8\ 4 \\ 1\ 4\ 8\ 3\ 7\ 9\ 7\ 5\ 8 & & 1\ 4\ 8\ 3\ 7\ 9\ 7\ 5\ 8 \end{array}$$

原本重复数字 2 被替换为了 8，按理可处理其他重复数字。

交叉后，按一定的概率进行变异，这里采取的变异操作是根据随机产生的两个随机数，确定基因交换位置，对一条染色体两个不同位置的基因进行交换，达到变异的目的。

3.3 遗传算法实现

将遗传算法与飞机降落调度实际数据结合，通过 MATLAB 编程实现，经过多次调参，结果最优的参数为：

$$P_c = 0.9, P_m = 0.008$$

其中 P_c 、 P_m 分别为交叉概率、变异概率。

在此参数下，航班降落调度结果如下图所示（运行代码 main.m 可得）：

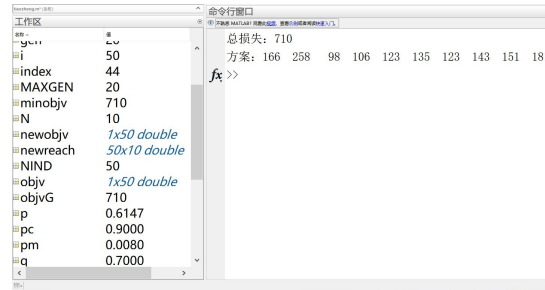


图 10 遗传算法优化结果

此时的航班降落时间分别为：

$$[166, 258, 98, 106, 119, 135, 127, 143, 151, 181]$$

此时的成本损失为 710。

此时的收敛迭代过程如下图所示：

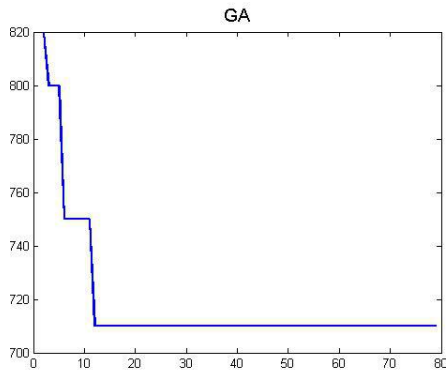


图 11 遗传算法收敛过程

可以发现，迭代大约 12 次后，优化结果达到 710 左右后基本不再变化。

3.4 遗传算法分析

3.4.1 种群数量的影响

为研究种群数量对算法收敛的影响，将粒子数量从 10 开始，以 50 为间隔，设置了 25 个不同的值，也即粒子数量分别为 10,60,110,...,1210,其他参数与最优参数一致，得到优化结果（各个种群数量下迭代的最优值）种群数量变化的曲线图：

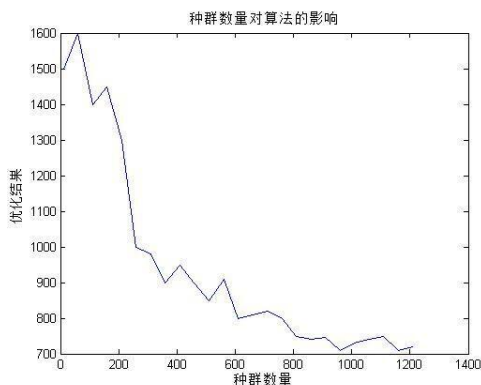


图 12 种群数量对算法的影响

可以发现，随着种群数量的增加，目标函数值也越来越小，优化结果越来越好，但当种群数达到一定值后，优化的结果基本在一条水平线上下波动（图中为 750 左右），说明随着种群数量的增加，算法的收敛效果越来越好，但达到一定值后，粒子数量的增加不会有显著的效果提升。

3.4.2 交叉概率的影响

在实验中多次改变交叉概率，将交叉概率设置为从 0.7 开始，以 0.05 为间隔，设置了 6 个不同的值，也即交叉概率分别为 0.7,0.75,...,0.95,发现当交叉概率约为 0.9 时，算法能够取得收敛的稳定性与优化结果的较好折衷。当交叉概率较小时，种群不能有效更新，收敛速度慢；而交叉概率过大时，经常收敛不到遗传优化最好的结果 710，说明交叉概率过大，随机性增加，易错过最优个体。

3.4.3 变异概率的影响

在实验中多次改变变异概率，将变异概率设置为从 0.001 开始，以 0.01 为间隔，设置了 10 个不同的值，也即交叉概率分别为 0.001,0.011,...,0.0910,发现当交叉概率约为 0.008 时，算法收敛较快且优化结果较好。当变异概率过小时，种群的收敛不太稳定，起伏较大，因为种群的多样性下降。

4 两种算法比较

1.均属于全局优化算法。粒子群主要模拟鸟类觅食、人类认知等社会行为而提出，而遗传算法主要借用达尔文的进化论。两种算法都是在解空间随机产生初始种群，继而在全局的解空间进行搜索，且随着迭代的过程，将搜索重点集中在“适应度”好的部分。

2.均采用随机搜索。遗传算法的遗传操作选择、交叉、变异均属于随机操作。粒子群算法中速度的更新中“自我认知”、“社会认知”前均有随机数增加随机性，两种算法均通过随机优化方法更新种群和搜索较优点。

3.对于某些问题，两种算法的收敛性不一定很好，且迭代次数可能很大，收敛性能差，不一定能够得到最优值。

4.在实现方面,相比较而言,遗传算法的编码技术和遗传操作比较繁杂,尤其涉及到多维解时;比如,此次航班降落调度问题中,若采用9位二进制编码,则一个个体需要90位二进制表示,进行遗传操作时也更复杂;而粒子群算法相对于遗传算法,不需要编码,没有交叉和变异操作,仅仅通过内部速度、位置进行更新,对速度、位置进行限制即可,因此原理更简单,实现更容易。

5.粒子群算法有记忆,由更新公式的“自我认知”、“社会认知”可以看出,粒子群的下次更新会参考上一次个体最优信息和全体最优信息,因此是有记忆的;遗传算法没有记忆,以前的知识随着种群的改变被破坏。

6.在实验效果方面,航班降落调度问题粒子群算法的优化结果与遗传算法的优化结果相比较优,但是迭代时间较长。

5 总结

本文主要结合航班降落调度的具体实例,设计并实现了基于群体智能优化算法的航班降落调度。主要包括基于粒子群算法的航班降落调度设计与实现、基于遗传算法的航班降落调度设计与实现。其中,采用粒子群算法得到的优化结果为700,采用遗传算法得到的最好优化结果为710。

参考文献

- [1].崔雪源. 基于遗传模拟退火算法的航班着陆调度问题[D]. 2015.
Cui Xueyuan. Flight landing scheduling based on genetic simulated annealing algorithm [D]. 2015
- [2].Kari Anderson, William Hall. Optimization-based analysis of collaborative airport arrival planning. Transportation Science 2003 INFORMS Vol.37,No.4,2003.
- [3].Aditya P.Saraf. An efficient combinatorial

optimization algorithm for optimal scheduling of aircraft at Congested Airports[J].IEEE,2006.

- [4].Hanboo Lee. Fuel cost, delay and throughout tradeoffs in runway scheduling[J].American Control Conference June11-13,2008,2449-2454.

- [5].张洪海,胡明华.多跑道降落飞机协同调度优化. 交通运输工程学报,2009,9(3):86-91.

Zhang Honghai, Hu Minghua. Cooperative scheduling optimization of multi runway landing aircraft. Journal of Transportation Engineering,2009,9(3):86-91.

- [6].曹力,邓雪云,王旭晖,黄胜国.基于机型间隔的机场终端区流量优化[J].系统工程学报,2009,24(3):360-363.

Cao Li, Deng Xueyun, Wang Xuhui, Huang Shengguo. Terminal area flow optimization based on aircraft type interval [J]. Journal of systems engineering,2009,24(3):360-363.

- [7].张子阳,张晓玮.基于遗传算法的航班着陆调度的设计与实现.电子设计工程学报,2013,21(3):3-6.
Zhang Ziyang, Zhang Xiaowei. Design and implementation of flight landing scheduling based on genetic algorithm. Journal of electronic design engineering, 2013,21 (3): 3-6

- [8].王璐,汪云峰.基于遗传算法的平行跑道航班着陆调度问题研究.航空计算技术学报,2019,49(2):28-33.

Wang Lu, Wang Yunfeng. Research on parallel runway flight landing scheduling based on genetic algorithm. Journal of Aeronautical computing technology, 2019,49 (2): 28-33

- [9].程晓航,薛惠锋.进港飞机调度的精华自适应遗传算法设计[J].交通与计算机,2006,24(6):91-94.

Cheng Xiaohang, Xue Huifeng. Design of

elite adaptive genetic algorithm for incoming aircraft scheduling [J]. traffic and computer, 2006,24 (6): 91-94.

[10].粒子群算法.

https://blog.csdn.net/weixin_40679412/article/details/80571854

[11]. <https://baike.baidu.com/item/遗传算法/838140?fr=aladdin>

