



中国研究生创新实践系列大赛

## “华为杯”第十八届中国研究生 数学建模竞赛

学 校

上海海事大学

参赛队号

21102540382

队员姓名

1.姜家乐

2.赵燕

3.董亮

---

# 中国研究生创新实践系列大赛

## “华为杯”第十八届中国研究生

### 数学建模竞赛

#### 题目 航空公司机组优化排班问题探析

---

#### 摘 要：

本文研究的是航班公司机组优化排班问题，通过建立线性优化模型，明确表达飞行时间、执勤时间、休息时间等约束，把航班直接分配给机组人员，以尽可能多的航班满足机组配置，尽可能少的机组人员的总执勤成本等要求作为目标。将复杂的机组优化排班问题简化为线性模型，通过设计改进的遗传算法进行求解。

针对问题一，机组优化排班的目标是尽可能多的航班满足机组配置，尽可能少的乘机次数，尽可能少的替补资格，通过考虑机组人员与航班的时间约束，机组人员在航班上的身份约束，人员所在地与航班起飞地的地点约束三方面进行约束建立线性优化模型。通过结合层次分析法对3个目标进行权重计算，将多目标优化问题转换为单目标优化问题，通过改进的遗传算法与CPLEX进行精确解求解，结果表明，本文提出的改进遗传算法具有很强的鲁棒性和搜索能力，小规模下的遗传算法结果和精确解得到的结果一致，对于A组算例，所有航班都满足最低机组配置，对于B组算例，有16班航班不满足最低机组配置。

针对问题二，在问题一的基础上，航班优化问题引入执勤概念，执勤表示员工在一天内执行的起飞航班，机组排班优化的目标是机组人员的总执勤成本最低，机组人员的执行时长尽可能的平衡，通过引入日期 $t$ ，将员工执勤起飞时间为 $t$ 的航段记作为员工执勤了 $t$ 日期的航段，通过计算最小的员工时长与平均时长的差值，可以得到机组人员 $i$ 的执勤时长平衡程度。通过层次分析法对5个目标进行权重计算，并用改进的遗传算法进行求解，结果表明，在A组算例中，所有的机组都满足最低机组配置，对于B组算例，有23组算例不满足最低机组配置。

针对问题三，在问题二的基础上，引入任务环以及排班的概念，机组排班优化的目标是机组人员的总体任务环成本最低，机组人员之间的任务环时长尽可能的平衡，通过层次分析法对7个目标进行权重计算，并通过改进的遗传算法进行求解，结果表明，在A组算例中，有108个航班不满足最低机组配置，在B组算例中，有59个航班不满足最低机组配置。

---

## 目录

第一章 问题重述 .....	4
1.1 问题背景 .....	4
1.2 本文提出的问题 .....	4
第二章 模型假设、相关定义与符号 .....	5
2.1 模型的假设 .....	5
2.2 相关符号及其定义 .....	5
第三章 问题分析 .....	7
3.1 针对子问题一 .....	7
3.2 针对子问题二 .....	7
3.3 针对子问题三 .....	7
第四章 子问题一的模型建立与求解 .....	8
4.1 模型的建立 .....	8
4.1.1 确定目标函数 .....	8
4.1.2 确定约束条件 .....	8
4.2 算法描述 .....	10
4.2.1 改进遗传算法设计 .....	10
4.2.2 层次分析法计算权重 .....	11
4.3 算例分析 .....	13
4.3.1 A组算例求解 .....	13
4.3.2 B组算例求解 .....	13
4.4 算法有效性和复杂性分析 .....	13
第五章 子问题二模型建立与求解 .....	15
5.1 模型建立 .....	15
5.1.1 新增目标函数 .....	15
5.1.2 新增约束条件 .....	15
5.2 层级分析法计算权重 .....	16
5.3 算例分析 .....	16
5.3.1 A组算例求解 .....	16
5.3.2 B组算例求解 .....	17
第六章 子问题三模型建立与求解 .....	19
6.1 模型建立 .....	19
6.1.1 新增目标函数 .....	19
6.1.2 新增约束条件 .....	19
6.2 层次分析法计算权重 .....	19
6.3 算例分析 .....	20
6.3.1 A组算例求解 .....	20
6.3.2 B组算例求解 .....	21
第七章 模型的评价 .....	23
7.1 模型评价 .....	23
7.1.1 模型的优点 .....	23
7.1.2 模型的缺点 .....	23
参考文献 .....	24

---

附录: .....	25
-----------	----

---

# 第一章 问题重述

## 1.1 问题背景

众所周知，一趟民航航班必须在满足特定的条件下才能起飞，这些条件包括国家法律法规，国际公约，政府的行政条例，和公司自身的政策利益，一些国家的工会组织还会对机组人员的福利偏好有规章约束。所有这些条件都是对保证飞航安全和旅客服务质量最重要的因素之一。

广义的机组人员包括飞行员（Pilot，或Flight Deck），乘务员（Cabin Crew）和空警（Air Marshal）。所谓机组排班问题，就是构造特定时间段的机组日程安排，包括每个机组人员在何时何地及哪个航班执行何种任务。

机组排班问题是运筹学应用的最早典范之一，经过过去 60 年的发展，已经形成了比较成熟甚至通用的解决方案，该解决方案把机组排班问题分解成两个子问题求解：以满足规范和节约成本为主要目的任务环生成，和以满足公平合理为主要目的任务环分配，但是两阶段子问题求解法也有其明显的缺陷。比如实际排班过程中需要具备人工预排班功能，一个机组人员可能会乘坐指定航班到某地但回程却由系统安排。两阶段解法不能很好地处理这样的不完整任务环问题。而且理论上，两阶段解法缩小了优化空间，所谓的最优解其实是次优解。同时，对列生成法而言，各种规章约束参数主要是在列生成过程中考虑，优化模型本身缺乏这些约束参数的显性表达，通常的参数敏感性分析手段不能采用，因而难以对业务规则的制订起到指导作用。

## 1.2 本文提出的问题

本题的宗旨是建立线性优化模型，明确表达飞行时间、执勤时间、休息时间等约束，把航班直接分配给机组人员。

**子问题一：**建立线性规划模型给航班分配机组人员（或者说给机组人员分配航班），依编号次序满足目标：尽可能多的航班满足机组配置、尽可能少的总体乘机次数、尽可能少使用替补资格。

**子问题二：**引进执勤概念。假定每个机组人员的每单位小时执勤成本给定（可以设想为小时工资）。本子问题除了需要满足子问题1的所有目标外，还需满足如下目标：机组人员的总体执勤成本最低、机组人员之间的执勤时长尽可能平衡。

**子问题三：**编制排班计划。假定每个机组人员的每单位小时任务环成本给定（注：不包括执勤成本，可以设想为出差补贴）。本子问题除了需要满足子问题1和2的所有目标外，还需满足如下目标：机组人员的总体任务环成本最低、机组人员之间的任务环时长尽可能平衡。

## 第二题 模型假设、相关定义与符号

### 2.1模型的假设

- 1、机组人员之间可以任意组合；
- 2、允许存在因为无法满足最低机组资格配置而不能起飞的航班；
- 3、不满足最低机组资格配置的航班不能配置任何机组人员；
- 4、机组人员可以乘机摆渡，即实际机组配置可以超过最低配置要求，乘机机组人员的航段时间计入执勤时间，但不计入飞行时间；
- 5、起飞时间和降落时间确定，不受外界因素影响。

### 2.2相关符号及其定义

表2-1 索引及其意义

索引	意义
$i, e$	机组人员
$j, j'$	航班号
$t$	日期
$h$	机场

表2-2 集合及其意义

集合	意义
$\Phi$	机组人员集合， $\Phi = \{1, 2, \dots, I\}$
$\Psi$	航班集合， $\Psi = \{1, 2, \dots, J\}$
$\Omega$	日期集合， $\Omega = \{1, 2, \dots, T\}$
$OS_i$	机组人员 $i$ 的基地起飞的航班集合， $OS_i \subseteq \Psi$
$OF_i$	抵达机组人员 $i$ 的基地的航班集合， $OF_i \subseteq \Psi$
$\Lambda$	基地机场集合， $\Lambda = \{1, 2, \dots, H\}$

表2-3 参数及其意义

参数	意义
$A_i$	$A_i = 1$ 表示机组人员 $i$ 是正机长,反之则 $A_i = 0$ , $i \in \Phi$
$B_i$	$B_i = 1$ 表示机组人员 $i$ 是副机长,反之则 $B_i = 0$ , $i \in \Phi$
$KS_{jh}$	航班 $j$ 在基地 $h$ 的起飞时间
$KF_{jh}$	航班 $j$ 在基地 $h$ 的到达时间
$S_{jt}$	出发日期为 $t$ 日期的航班 $j$ 的出发时间， $t \in \Omega$ 、 $j \in \Psi$
$F_{jt}$	出发日期为 $t$ 日期的航班 $j$ 的到达时间， $t \in \Omega$ 、 $j \in \Psi$
$P_j$	航班 $j$ 出发机场， $j \in \Psi$
$D_j$	航班 $j$ 到达机场， $j \in \Psi$

---

$C_i$	机组人员 $i$ 的单位执勤成本, $i \in \Phi$
$G_{ih}$	$G_{ih} = 1$ 表示机组人员 $i \in \Phi$ 的基地为 $h \in \Lambda$ , 否则为 0
$MinCT$	最小航段间的连接时间
$MaxBlk$	最大一次执勤的飞行时长
$MaxDP$	最大执勤时长
$MinRest$	最小相邻执勤之间的休息时间
$MaxDH$	最大每趟航班乘机人数
$MaxTAFB$	最大排班周期单个机组人员任务环总时长
$MaxSuccon$	最大连续执勤天数
$MinVacDay$	最大相邻两个任务环之间的休息天数
$M$	足够大的正整数

---

表2-4 决策变量及其意义

决策变量	意义
$x_{ijt}$	$x_{ijt} = 1$ 表示机组人员 $i \in \Phi$ 在日期 $t \in \Omega$ 乘坐航班 $j \in \Psi$ , 否则 $x_{ijt} = 0$
$y_j$	$y_j = 1$ 表示航班 $j \in \Psi$ 满足最低机组资格配置, 否则 $y_j = 0$
$z_{ij}$	$z_{ij} = 1$ 表示机组人员 $i \in \Phi$ 以替补资格执行航班 $j \in \Psi$ , 否则 $z_{ij} = 0$
$k_{ij}$	$k_{ij} = 1$ 表示机组人员 $i \in \Phi$ 以乘机任务执行航班 $j \in \Psi$ , 否则 $k_{ij} = 0$
$m_{ij}$	$m_{ij} = 1$ 表示机组人员 $i \in \Phi$ 以正机长资格执行航班 $j \in \Psi$ , 否则 $m_{ij} = 0$
$n_{ij}$	$n_{ij} = 1$ 表示机组人员 $i \in \Phi$ 以副机长资格执行航班 $j \in \Psi$ , 否则 $n_{ij} = 0$
$l_{ijj't}$	$L_{ijj't} = 1$ 表示机组人员 $i \in \Phi$ 在日期 $t \in \Omega$ 结束执行航班任务 $j \in \Psi$ 后执行航班任务 $j' \in \Psi$ , 否则 $L_{ijj't} = 0$
$u_i$	线性化辅助变量

---

---

## 第三章 问题分析

### 3.1 针对子问题一

机组排班问题需要考虑众多因素，如航班的时间约束，航班的起飞机场节点和到达机场节点，航班的机组人员配置等，因此，制定一个合理的航班分配方案相对比较复杂。针对问题一，合理的给航班分配机组人员，首先需要满足航班最低的机组资格配置，即配备一定数量的正副机长，使航班达到飞行要求，通过考虑各个航班的时空关系，来为机组人员分配飞行任务，使尽可能多的航班满足机组配置。为了能减少总的乘机次数，就需要考虑航班在每个节点的任务和时间衔接。尽可能少的使用替补资格，需要首先为航班分配正机长，再为航班分配副机长，在副机长数量不足的情况下，为让更多的航班满足航班最低的机组资格配置，安排替补副机长。

### 3.2 针对子问题二

问题二在问题一的基础上增加了执勤的概念，即上一趟航段的到达和下一趟航段的出发机场一致，上下两趟航班需要满足最短的连接时间、同一次执勤的起飞时间必须在同一天。问题二引入了执勤成本问题，为了让更多的航班满足机组配置，并使机组人员总体执勤成本最低，需要让飞行时间相对较短的航班尽可能多的满足飞行要求。在满足尽可能多的航班满足机组配置要求的情况下，使机组人员之间的执勤时长尽可能均衡，则需要在满足单次执勤时长和休息时间的情况下，使机组人员每天的执勤时长尽可能的均衡。

### 3.3 针对子问题三

问题三在问题二的基础上增加任务环的概念，即由一连串的执勤时间和休息时间组成，机组人员从基地出发并最终回到基地。问题三引入了任务环成本，使机组人员的总体任务环成本最低，需减少任务环的作业时长，即机组休息时间尽可能安排在基地。使机组人员的任务环时长尽可能平衡，需在满足任务环时长的约束的情况下，使机组人员的任务环作业时长均衡。



## 第四章 子问题一的模型建立与求解

子问题一题中要求建立线性规划模型给航班分配机组人员，需要考虑机组人员与航班的时间约束，机组人员在航班上的身份约束，人员所在地与航班起飞地的地点约束等。通过合理安排机组人员的任务，使得尽可能多的航班满足机组配置，尽可能少的总体乘机次数，尽可能少使用替补资格。

### 4.1 模型的建立

#### 4.1.1 确定目标函数

子问题一的目标为尽可能多的航班满足机组配置，尽可能少的总体乘机次数，尽可能少使用替补资格， $y_j = 1$ 为航班满足最低机组配置， $k_{ij} = 1$ 为机组人员 $i$ 以乘机任务执行航班 $j$ ， $k_{ij} = 1$ 为机组人员 $i$ 以替补资格身份执行航班 $j$ ，则目标函数如下所示：

$$\begin{cases} f_1 = \max \sum_{j=1}^J y_j \\ f_2 = \min \sum_{i=1}^I \sum_{j=1}^J k_{ij} \\ f_3 = \min \sum_{i=1}^I \sum_{j=1}^J z_{ij} \end{cases} \quad (1)$$

#### 4.1.2 确定约束条件

##### 1、确定机组人员在航班上的身份约束

1) 如果机组人员 $i$ 执行航班任务 $j$ ，则该机组人员 $i$ 属于 $j$ 航班任务中的正机长，副机长，乘机人员3种身份的一种

$$k_{ij} + m_{ij} + n_{ij} = \sum_{t=1}^T x_{ijt} \quad \forall i \in \Phi, j \in \Psi \quad (2)$$

2) 机组人员 $i$ 是否以替补身份参与航班任务 $j$

$$A_i \cdot B_i \cdot n_{ij} = z_{ij} \quad \forall i \in \Phi, j \in \Psi \quad (3)$$

3) 机组人员 $i$ 必须具备正机长资格才能以正机长身份执行飞行任务 $j$

$$A_i \geq m_{ij} \quad \forall i \in \Phi, j \in \Psi \quad (4)$$

4) 机组人员 $i$ 必须具备副机长资格才能以副机长身份执行飞行任务 $j$

$$B_i \geq n_{ij} \quad \forall i \in \Phi, j \in \Psi \quad (5)$$

5) 对于满足最低资格约束的航班 $j$ 有且仅有一个正机长

$$\sum_{i=1}^I A_i \cdot m_{ij} = y_j \quad \forall j \in \Psi \quad (6)$$

6) 对于满足最低资格约束的航班  $j$  有且仅有一个副机长

$$\sum_{i=1}^I B_i \cdot n_{ij} = y_j \quad \forall j \in \Psi \quad (7)$$

7) 航班  $j$  的最大乘机人数限制

$$\sum_{i=1}^I k_{ij} \leq MaxDH \quad \forall j \in \Psi \quad (8)$$

8) 如果航班任务  $j$  机组员工  $i$  执行航班任务  $j$ ，则航班  $j$  满足最低要求配置

$$\sum_{t=1}^T \sum_{i=1}^I x_{ijt} \leq y_j \cdot M \quad \forall j \in \Psi \quad (9)$$

$$\sum_{t=1}^T \sum_{i=1}^I x_{ijt} \geq y_j \quad \forall j \in \Psi \quad (10)$$

9)  $l_{ij't}$  和  $x_{ijt}$  的关系约束

$$2 \cdot l_{ij't} \leq x_{ijt} + x_{ij't} \quad \forall i \in \Phi, \forall j, j' \in \Psi, \forall t \in \Omega, j \neq j' \quad (11)$$

10) 员工  $i$  乘坐的航班  $j$  最多有一个后续航班  $j'$

$$\sum_{j'=1}^J l_{ij't} \leq 1 \quad \forall i \in \Phi, \forall j \in \Psi, \forall t \in \Omega, j \neq j' \quad (12)$$

11) 员工  $i$  乘坐的航班  $j'$  最多有一个前续航班  $j$

$$\sum_{j=1}^J l_{ij't} \leq 1 \quad \forall i \in \Phi, \forall j' \in \Psi, \forall t \in \Omega, j \neq j' \quad (13)$$

## 2、确定人员所在地与航班起飞地的地点约束

12) 每个机组人员初始从基地出发并最终回到基地

$$\sum_{t=1}^T \sum_{j=1}^J P_j \cdot x_{ijt} = \sum_{t=1}^T \sum_{j=1}^J D_j \cdot x_{ijt} \quad \forall i \in \Phi \quad (14)$$

13) 每个机组人员的下一航段的起飞机场必须和上一航段的到达机场一致

$$l_{ij't}(P_j - D_{j'}) = 0 \quad \forall i \in \Phi, \forall j, j' \in \Psi, \forall t \in \Omega, j \neq j' \quad (15)$$

14) 员工  $i$  从基地出发，再回到基地

$$\sum_{j=1}^J l_{iojt} = \sum_{j'=1}^J l_{ij'o't} \quad \forall i \in \Phi, \forall t \in \Omega, o \in OS_i, o' \in OF_i \quad (16)$$

15) 员工  $i$  执行飞行任务需首先从基地出发

$$\sum_{j=1}^J l_{iojt} \cdot S_{jt} \leq \sum_{j'=1}^J l_{ij'o't} \cdot S_{j't} \quad \forall i \in \Phi, \forall t \in \Omega, o \in OS_i, o' \in OF_i \quad (17)$$

## 3、确定机组人员与航班的时间约束

16) 每个机组人员相邻两个航段之间的连接时间不小于 $MinCT$ 分钟

$$l_{ijt}(F_j - MinCT - S_{j'}) \leq 0 \quad \forall i \in \Phi, \forall j, j' \in \Psi, \forall t \in \Omega, j \neq j' \quad (18)$$

## 4.2 算法描述

航班分配机组人员问题属于NP-hard问题，遗传算法在求解NP-hard问题中具有很强的适用性，遗传算法（GA）相比其他算法具有鲁棒性强，搜索能力强，但针对机组人员航班问题中，搜索航班任务过程中容易过早收敛，从而可能导致结果为局部最优解，提出了改进遗传算法：（1）在标准遗传算法的基础上采用了一种特殊的编码方式，解决了航班与机组人员多对多的关系。（2）提出针对该编码方式的动态自适应交叉和变异概率，有效的避免了遗传算法局部最优解的产生。（3）提出父代子代融合操作，以防丢失优质解及陷入局部最优解。因此本文采用遗传算法进行求解，考虑目标满足的优先级，其次针对本文多目标的因素，采用层次分析法为每个目标设置权重，将多目标问题转换为单目标问题。

### 4.2.1改进遗传算法设计

#### 1) 遗传算法染色体编码

针对航班机组人员分配问题的复杂性，本文基因编码由206\*21行的矩阵构成，每一行代表一个航班，每一列表示机组人员，矩阵中的1表示该人员乘坐了该行对应的航班，如下图所示，表示机组人员2和人员18乘坐了航班1，机组人员1和机组人员18乘坐了航班206。

		人员									
		1	2	3	.....	18	19	20	21		
航 班	1	0	1	0	.....	1	0	0	0		
	2	0	0	1		0	1	0	0		
	⋮	⋮	⋮	⋮	.....	⋮	⋮	⋮	⋮		
	206	1	0	0	.....	1	0	0	0		

图 4-1 染色体编码

#### 2) 交叉

本文采用传统交叉方式，在父代1中随机选择多个位置，保存到子代染色体当中，再从父代2中选出子代中缺失的部分按顺序补充到子代中。

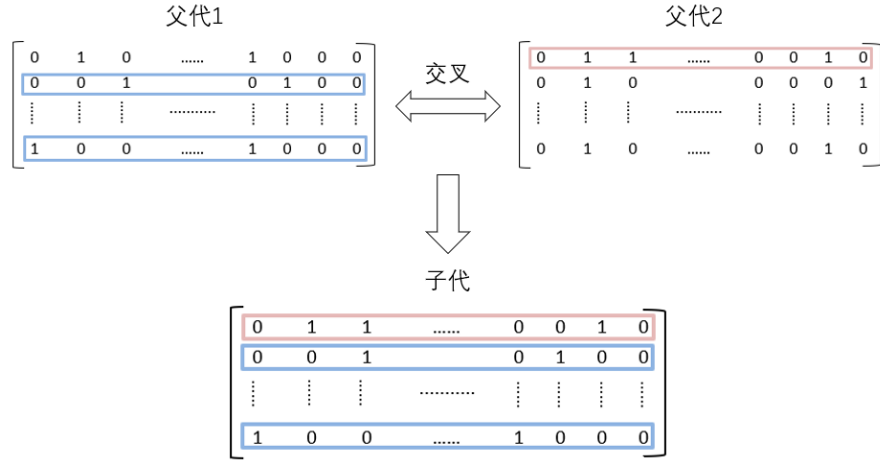


图 4-2 染色体交叉

### 3) 变异

变异随机选择染色体位置进行变异

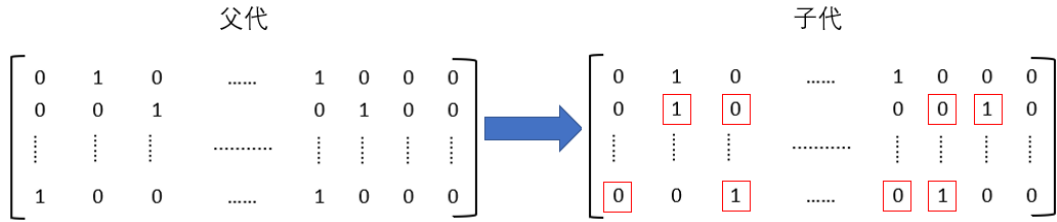


图 4-3 染色体变异

### 4) 染色体调整策略

随机产生的染色体与每代交叉变异后产生的新的种群中可能存在不可行解，即安排给机组人员的航班之间发生冲突，则需要对染色体进行调整。

### 5) 选择及适应度计算

适应度计算公式：

$$f = \frac{1}{\alpha_1 \cdot f_1 \cdot \beta_1 + \alpha_2 \cdot f_2 \cdot \beta_2 + \alpha_3 \cdot f_3 \cdot \beta_3}$$

问题一的目标函数一共包含3个，分别是  $f_1$  尽可能多的航班满足机组配置、 $f_2$  尽可能少的总体乘机次数、 $f_3$  尽可能少的使用替补资格，本文首先通过  $\beta_1$ 、 $\beta_2$ 、 $\beta_3$  将三个目标函数  $f_1$ 、 $f_2$ 、 $f_3$  将三个目标函数归一化，然后通过权重系数  $\alpha_1$ 、 $\alpha_2$ 、 $\alpha_3$  对三个目标进行求和，取倒数作为该染色体的适应度值。

#### 4.2.2 层次分析法计算权重

权重系数  $\alpha_1$ 、 $\alpha_2$ 、 $\alpha_3$  通过层次分析法进行求解，层次分析法步骤如下：

##### 1) 构建判断矩阵

判断矩阵即以矩阵的形式来表示每一层次中各要素相对其上层要素的相对重要程度。为了使各要素之间进行两两比较得到量化的判断矩阵，引入1~9的标度，见下表所示。

表 4-1 标度表

标度	定义
1	$\sigma$ 因素与 $\theta$ 因素同等重要
3	$\sigma$ 因素比 $\theta$ 因素略重要
5	$\sigma$ 因素比 $\theta$ 因素较重要
7	$\sigma$ 因素比 $\theta$ 因素非常重要
9	$\sigma$ 因素比 $\theta$ 因素绝对重要
2, 4, 6, 8	为以上判断之间的中间状态对应的标度值
倒数	$\sigma$ 因素与 $\theta$ 因素的比较判断和 $\theta$ 因素与 $\sigma$ 因素的比较判断互为倒数

为了构造判断矩阵，结合题目要求，通过对专家进行咨询，根据专家和作者的经验，四个准则下的两两比较矩阵为：

$$A = \begin{Bmatrix} 1 & 5 & 9 \\ 0.2 & 1 & 4 \\ 0.11 & 0.25 & 1 \end{Bmatrix}$$

## 2) 层次单排序及其一致性检验

$$A = \begin{Bmatrix} 1 & 5 & 9 \\ 0.2 & 1 & 4 \\ 0.11 & 0.25 & 1 \end{Bmatrix} \xrightarrow{\text{列向量归一化}} \begin{Bmatrix} 0.7627 & 0.8 & 0.6429 \\ 0.1525 & 0.16 & 0.2857 \\ 0.0847 & 0.04 & 0.0714 \end{Bmatrix} \xrightarrow{\text{按行求和归一化}} \begin{Bmatrix} 0.7627 \\ 0.1525 \\ 0.0847 \end{Bmatrix} = W$$

$$\lambda = A * W / 3 = 3.0724$$

用一致性指标进行检验，检验公式如下：

$$CI = \frac{\lambda_{\max} - n}{n - 1}$$

$$CR = \frac{CI}{RI}$$

RI 取值如下表 2 所示：

表 4-2 平均随机一致性指标 RI 标准值

矩阵阶数	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

最终求得  $CR = 0.0625$ ，不一致程度在允许范围内，所以权重系数分别为  $\alpha_1 = 0.7627$ 、 $\alpha_2 = 0.1525$ 、 $\alpha_3 = 0.0847$ 。

### 4.3 算例分析

#### 4.3.1 A组算例求解

针对与A组算例，应用商业求解器IBM ILOG CPLEX 12.8和改进的遗传算法进行求解，其中遗传算法迭代次数为1500代，交叉概率为0.8，变异概率为0.1，采用MATLAB语言实现，在AMD Ryzen 5 4600H处理器、16GB内存上的计算机上运行。

A组算例求解结果如下表所示，表中CPLEX表示用求解器求得的精确解，GA表示由改进的遗传算法求得的算法最优解。

表4-3 子问题1 A组算例求解结果

结果指标	子问题1	
	CPLEX	GA
不满足机组配置航班数	0	0
满足机组配置航班数	206	206
机组人员总体乘机次数	8	8
替补资格使用次数	0	0
程序运行分钟数(分钟)	74	82

对于A算例的子问题1求解，CPLEX求得的精确解与改进的遗传算法求得的解相同，没有不满足机组配置的航班数，满足机组配置的航班数为206。机组人员总体乘机次数为8，有8个人需要搭乘航班到达其他机场执行飞行任务。没有人使用替补资格完成飞行任务，CPLEX求解器程序运行时间为74分钟，而改进的遗传算法求解时间为82分钟。改进的遗传算法可以在有限的时间里得到与CPLEX求解器几乎相同的解，证明本文设计的模型的正确性与算法的有效性。

#### 4.3.2 B组算例求解

B组算例求解结果如下表所示，由于CPLEX无法对大规模数据进行求解，因此通过改进的遗传算法算例结果如下：

结果指标	子问题1
不满足机组配置航班数	16
满足机组配置航班数	13938
机组人员总体乘机次数	4238
替补资格使用次数	829
程序运行分钟数(分钟)	191

对于B算例的子问题1求解，只有16航班不能满足最低配置，其中机组人员总体乘机次数为4238，替补资格使用次数为829次。

### 4.4 算法有效性和复杂性分析

遗传算法（GA）作为一种智能优化搜索算法，具有扩展性、并行性、鲁棒性好以及

---

简单通用等特征。在一个编码后的解集合内，遗传算法按照一定的规则随机对种群中的个体进行各种遗传操作，高效搜索，最终得到所求问题的最优解。在使用过程中，遗传算法能够普遍适用于各种优化问题的求解，拥有较强的自我调整、组织、学习的能力，被广泛应用于各种领域。总的来说，将遗传算法用于航班机组人员分配问题拥有以下特点：

(1)遗传算法不会直接计算搜索对象的实际值，而是先会对航班任务、机组人员进行编码处理，然后对搜索对象的编码进行计算，方便问题的描述与遗传操作的进行。

(2)在搜索最优解的时候，传统优化算法需要依靠目标函数的实际值，通常会受限于目标函数的导数必须存在的条件和连续性的问题。而遗传算法只要得到了个体适应度，就可以确定接下来的搜索范围和方向，拥有较好的搜索效率。

(3)在搜索最优解的时候，遗传算法采用多点并行搜索信息，迭代过程的起点是航班任务与机组人员组成的染色体种群，而不是解空间中某一个可行解，这样获得的航班任务与机组人员匹配信息足够多，提高了搜索效率，避免出现局部最优。

(4)使用概率化搜索方法。在搜索最优解的时候，传统优化算法的搜索方法通常是固定的，点到点之间的移动是有规则的，而遗传算法的都是以一定概率进行遗传操作，搜索起来足够灵活，拥有较强的全局搜索能力。

## 第五章 子问题二模型建立与求解

子问题二在问题一的基础上增加了执勤的概念，因此需要对每次执勤的时间进行约束，对每次执勤的飞行时间进行约束，对执勤的休息时间进行约束，所以对问题二需要在问题一的基础上加上对执勤情况的约束以及新增执勤成本以及执勤的时间尽可能均衡目标。

### 5.1 模型建立

#### 5.1.1 新增目标函数

新增目标函数如下：

$$\begin{cases} f_4 = \min \sum_{i=1}^I \sum_{j=1}^J \sum_{j'=1}^J \sum_{t=1}^T l_{ijj't} ((F_{jt} - S_{jt}) + (S_{j't} - F_{jt})) C_i \\ f_5 = \min \sum_{i=1}^I u_i \end{cases} \quad (19)$$

其中 $f_4$ 表示最小化机组人员总体执勤成本， $f_5$ 表示最小化员工单次执勤时长与平均值之差。

#### 5.1.2 新增约束条件

问题二在问题一的基础上新增约束：

17) 每次执勤的飞行时间最多不超过  $MaxBlk$  分钟

$$\sum_{j=1}^J (F_{jt} - S_{jt}) m_{ij} + \sum_{j'=1}^J (F_{j't} - S_{j't}) n_{ij} \leq MaxBLK \quad \forall i \in \Phi, \forall t \in \Omega \quad (20)$$

18) 每次执勤的时长最多不超过  $MaxDP$  分钟

$$\sum_{j=1}^J l_{ijj't} ((F_{jt} - S_{jt}) + (S_{j't} - F_{jt})) \leq MaxDP \quad \forall i \in \Phi, \forall j' \in \Psi, \forall t \in \Omega, j \neq j' \quad (21)$$

19) 每个机组人员的相邻两个执勤之间的休息时间不小于  $MinRest$

$$x_{ij't} \cdot S_{j'(t+1)} - x_{ijt} \cdot F_{jt} \geq MinRest \quad \forall i \in \Phi, \forall j, j' \in \Psi, \forall t \in \Omega, j \neq j' \quad (22)$$

20) 线性化约束, 计算人员执勤时间与平均执勤时间差值

$$\left( \sum_{t=1}^T \sum_{j=1}^J l_{ijj't} ((F_{jt} - S_{jt}) + (S_{j't} - F_{jt})) \right) - \left( \sum_{i=1}^I \sum_{t=1}^T \sum_{j=1}^J l_{ijj't} ((F_{jt} - S_{jt}) + (S_{j't} - F_{jt})) \right) / I \leq u_i, \forall i \in \Phi, \forall j' \in \Psi, j \neq j' \quad (23)$$

$$\left( \sum_{t=1}^T \sum_{j=1}^J l_{ijj't} ((F_{jt} - S_{jt}) + (S_{j't} - F_{jt})) \right) - \left( \sum_{i=1}^I \sum_{t=1}^T \sum_{j=1}^J l_{ijj't} ((F_{jt} - S_{jt}) + (S_{j't} - F_{jt})) \right) / I \geq -u_i, \forall i \in \Phi, \forall j' \in \Psi, j \neq j' \quad (24)$$



$$u_i \geq 0 \quad \forall i \in \Phi \quad (25)$$

## 5.2 层次分析法计算权重

问题二新增机组人员的总体执勤成本最低、机组人员之间的执勤时长尽可能均衡两个目标，根据上述分析建模可知，该问题与问题一类似，因此本文仍采用层次分析法和遗传算法对问题进行求解，遗传算法与问题一类似，不再赘述。

层次分析法求解各个目标权重：

(1) 通过表 1 标度表以及专家经验得到构造矩阵

$$A = \begin{Bmatrix} 1 & 2 & 4 & 5 & 7 \\ 0.5 & 1 & 2 & 3 & 5 \\ 0.25 & 0.5 & 1 & 2 & 3 \\ 0.2 & 0.3333 & 0.5 & 1 & 2 \\ 0.1429 & 0.2 & 0.5 & 0.5 & 1 \end{Bmatrix}$$

(2) 层次单排序及其一致性检验

$$A = \begin{Bmatrix} 1 & 2 & 4 & 5 & 7 \\ 0.5 & 1 & 2 & 3 & 5 \\ 0.25 & 0.5 & 1 & 2 & 3 \\ 0.2 & 0.3333 & 0.5 & 1 & 2 \\ 0.1429 & 0.2 & 0.5 & 0.5 & 1 \end{Bmatrix} \xrightarrow{\text{列向量归一化}} \begin{Bmatrix} 0.4778 & 0.4959 & 0.5 & 0.4348 & 0.4778 \\ 0.2389 & 0.2479 & 0.25 & 0.2609 & 0.2389 \\ 0.1195 & 0.124 & 0.125 & 0.1739 & 0.1195 \\ 0.0956 & 0.0826 & 0.0625 & 0.087 & 0.0956 \\ 0.0683 & 0.0496 & 0.0625 & 0.0435 & 0.0683 \end{Bmatrix} \xrightarrow{\text{按行求和归一化}} \begin{Bmatrix} 0.4778 \\ 0.2389 \\ 0.1195 \\ 0.0956 \\ 0.0683 \end{Bmatrix} = W$$

$$\lambda = A * W / 5 = 5.1370$$

$$CR = 0.0342$$

不一致程度在允许范围内，所以求出对应的权重系数  $\alpha_1 = 0.4788$ 、 $\alpha_2 = 0.2389$ 、 $\alpha_3 = 0.1195$ 、 $\alpha_4 = 0.0956$ 、 $\alpha_5 = 0.0683$ 。

## 5.3 算例分析

### 5.3.1 A组算例求解

对于A组算例，同样应用商业求解器IBM ILOG CPLEX 12.8和改进的遗传算法进行求解，其中遗传算法迭代次数为1500代，交叉概率为0.8，变异概率为0.1，采用MATLAB语言实现，在AMD Ryzen 5 4600H处理器、16GB内存上的计算机上运行。

A组算例求解结果如下表所示，表中CPLEX表示用求解器求得的精确解，GA表示由改进的遗传算法求得的算法最优解。

结果指标	子问题2	
	CPLEX	GA
不满足机组配置航班数	0	0
满足机组配置航班数	206	206
机组人员总体乘机次数	8	8
替补资格使用次数	0	20
机组总体利用率(%)	75.35	75.35
最小 一次执勤飞行时长	3	3
平均 一次执勤飞行时长	4.1	4.1
最大 一次执勤飞行时长	4.7	4.7
最小 一次执勤执勤时长	3.7	3.7
平均 一次执勤执勤时长	4.8	4.8
最大 一次执勤执勤时长	4.4	4.4
最小 机组人员执勤天数	3	3
平均 机组人员执勤天数	5	5
最大 机组人员执勤天数	10	10
总体执勤成本(万元)	58.15	58.28
程序运行分钟数(分钟)	90	86

对于A算例的子问题2求解，子问题二CPLEX求得的精确解较优于改进的遗传算法求得的解。全部的航班都可以满足最低配置，机组人员总体乘机次数都为8。其中执勤时间与飞行时间都相同，最低配置总体飞行时长为25221分钟，总体执勤时长33471分钟，因此，机组总体利用率为最低配置总体飞行时长 / 总体执勤时长，即75.35%。最小一次执勤飞行时长为3小时，平均一次执勤飞行时长约为4.1小时，最大一次执勤飞行时长为4小时40分钟；最小一次执勤的执勤时间为3小时40分钟，平均一次执勤的执勤时间为4小时50分钟，最大一次执勤的执勤时间为4小时20分钟；最小机组人员执勤天数为3，平均机组人员执勤天数为5，最大机组人员执勤天数为10。而CPLEX求得的精确解中没有人使用替补资格，总体执勤成本为58.15，改进的遗传算法求得的替补资格使用次数为20次，总执勤成本为58.28。因此CPLEX求得的解替补资格使用次数更少，总体执勤成本更低，结果更好。

### 5.3.2 B组算例求解

B组算例求解结果如下表所示，由于CPLEX无法对大规模数据进行求解，因此通过改进的遗传算法算例结果如下：

结果指标	子问题2
不满足机组配置航班数	23
满足机组配置航班数	13931
机组人员总体乘机次数	4352
替补资格使用次数	1067
机组总体利用率(%)	80.64
最小 一次执勤飞行时长	3
平均 一次执勤飞行时长	4.1
最大 一次执勤飞行时长	4.7

---

最小 一次执勤执勤时长	3.7
平均 一次执勤执勤时长	4.8
最大 一次执勤执勤时长	4.4
最小 机组人员执勤天数	3
平均 机组人员执勤天数	11.2
最大 机组人员执勤天数	18
总体执勤成本（万元）	2907.5
程序运行分钟数(分钟)	194

对于B算例的子问题2求解，存在23个航班不可以满足最低配置，其余13931个航班都可以满足最低配置。机组人员总体乘机次数与替补资格使用次数相较于子问题1略微增加，机组总体利用率为80.64%。总体执勤成本为2907.5万元。

## 第六章 子问题三模型建立与求解

问题三在问题二的基础上增加任务环的概念，即由一连串的执勤时间和休息时间组成，机组人员从基地出发并最终回到基地。问题三引入了任务环成本，使机组人员的总体任务环成本最低，需减少任务环的作业时长，即机组休息时间尽可能安排在基地。使机组人员的任务环时长尽可能平衡，需在满足任务环时长的约束的情况下，使机组人员的任务环作业时长均衡。

### 6.1 模型建立

#### 6.1.1 新增目标函数

新增目标函数：

$$\begin{cases} f_6 = \text{Min} \sum_{i=1}^I ((\sum_{j=1}^J \sum_{t=1}^T G_{ih} K_{jh}^F x_{ijt} - \sum_{j=1}^J \sum_{t=1}^T G_{ih} K_{jh}^S x_{ijt}) CC_i) \\ f_7 = \text{Min} \sum_{i=1}^I \sum_{t'=1}^T |(G_{ih} K_{jh}^F x_{ijt} - \sum_{j=1}^J G_{ih} K_{jh}^S x_{ijt}) - (\sum_{j=1}^J G_{i'h} K_{jh}^S x_{i'jt} - \sum_{j=1}^J G_{i'h} K_{jh}^S x_{i'jt})| \end{cases} \quad (26)$$

其中 $f_6$ 是最小化总任务环成本， $f_7$ 是最小化任务环时间差值以达到任务环时间均衡。

#### 6.1.2 新增约束条件

新增约束条件：

- 1) 每个机组人员每个排班周期的任务环总时长不超过MAXTAFB分钟

$$\sum_{j=1}^J \sum_{t=1}^T G_{ih} K_{jh}^F x_{ijt} - \sum_{j=1}^J \sum_{t=1}^T G_{ih} K_{jh}^S x_{ijt} \leq \text{MaxTAFB}, \forall i \in \Phi, h \in \Lambda \quad (27)$$

- 2) 相邻任务环之间的间隔

$$(\sum_{j=1}^J \sum_{t=1}^T G_{ih} K_{jh}^F x_{ijt} - \sum_{j=1}^J \sum_{t=1}^T G_{ih} K_{jh}^S x_{ijt}) + (1 - L_{ijj'})M \geq \text{MinVacDay}, \forall i \in \Phi, h \in \Lambda \quad (28)$$

- 3) 每个机组乘员的连续执勤天数不超过MaxSuccon天

$$\sum_{j=1}^J \sum_{t=1}^{T-1} \sum_{t'=t+1}^T (\sum_{t=1}^{T-1} G_{ih} \cdot F_{jt} \cdot KF_{jh} \cdot x_{ijt} + \sum_{t'=t+1}^T G_{ih} \cdot S_{jt} \cdot KS_{jh} \cdot x_{ijt'}) \leq \text{MaxSuccon}, \forall i \in \Phi \quad (29)$$

### 6.2 层次分析法计算权重

问题三算法同前两问一致，层次分析法求解各个目标权重：

- (1) 通过标度表以及专家经验得到构造矩阵

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0.5 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0.3333 & 0.5 & 1 & 2 & 3 & 4 & 5 \\ 0.25 & 0.3333 & 0.5 & 1 & 2 & 3 & 4 \\ 0.2 & 0.25 & 0.3333 & 0.5 & 1 & 2 & 3 \\ 0.1667 & 0.2 & 0.25 & 0.3333 & 0.5 & 1 & 2 \\ 0.1429 & 0.1667 & 0.2 & 0.25 & 0.3333 & 0.5 & 1 \end{pmatrix}$$

## (2) 层次单排序及其一致性检验

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0.5 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0.3333 & 0.5 & 1 & 2 & 3 & 4 & 5 \\ 0.25 & 0.3333 & 0.5 & 1 & 2 & 3 & 4 \\ 0.2 & 0.25 & 0.3333 & 0.5 & 1 & 2 & 3 \\ 0.1667 & 0.2 & 0.25 & 0.3333 & 0.5 & 1 & 2 \\ 0.1429 & 0.1667 & 0.2 & 0.25 & 0.3333 & 0.5 & 1 \end{pmatrix} \xrightarrow{\text{列向量归一化}} \begin{pmatrix} 0.3857 & 0.4494 & 0.4119 & 0.3609 & 0.3158 & 0.2791 & 0.2500 \\ 0.1928 & 0.2247 & 0.2746 & 0.2707 & 0.2526 & 0.2326 & 0.2143 \\ 0.1286 & 0.1124 & 0.1373 & 0.1805 & 0.1895 & 0.1860 & 0.1786 \\ 0.0964 & 0.0749 & 0.0686 & 0.0902 & 0.1263 & 0.1395 & 0.1429 \\ 0.0771 & 0.0562 & 0.0458 & 0.0451 & 0.0632 & 0.0930 & 0.1071 \\ 0.0643 & 0.0449 & 0.0343 & 0.0301 & 0.0316 & 0.0465 & 0.0714 \\ 0.0551 & 0.0375 & 0.0275 & 0.0226 & 0.0211 & 0.0233 & 0.0357 \end{pmatrix} \xrightarrow{\text{按行求和归一化}} \begin{pmatrix} 0.3504 \\ 0.2375 \\ 0.1590 \\ 0.1056 \\ 0.0696 \\ 0.0462 \\ 0.0318 \end{pmatrix} = W$$

$$\lambda = A * W / 5 = 7.1973$$

$$CR = 0.0249$$

不一致程度在允许范围内，所以  $\alpha_1 = 0.3504$ 、 $\alpha_2 = 0.2375$ 、 $\alpha_3 = 0.1590$ 、 $\alpha_4 = 0.1056$ 、 $\alpha_5 = 0.0696$ 、 $\alpha_6 = 0.0462$ 、 $\alpha_7 = 0.0318$

## 6.3 算例分析

### 6.3.1 A组算例求解

对于A组算例，同样应用商业求解器IBM ILOG CPLEX 12.8和改进的遗传算法进行求解，其中遗传算法迭代次数为1500代，交叉概率为0.8，变异概率为0.1，采用MATLAB语言实现，在AMD Ryzen 5 4600H处理器、16GB内存上的计算机上运行。

A组算例求解结果如下表所示，表中CPLEX表示用求解器求得的精确解，GA表示由改进的遗传算法求得的算法最优解。

结果指标	子问题3	
	CPLEX	GA
不满足机组配置航班数	108	112
满足机组配置航班数	98	94

机组人员总体乘机次数	0	0
替补资格使用次数	0	0
机组总体利用率（%）	81.16	80.40
最小一次执勤飞行时长	2.3	2.3
平均一次执勤飞行时长	3.5	3.4
最大一次执勤飞行时长	4.7	4.7
最小一次执勤执勤时长	2.9	2.8
平均一次执勤执勤时长	4.2	4.2
最大一次执勤执勤时长	5.3	5.2
最小机组人员执勤天数	4	4
平均机组人员执勤天数	5	5
最大机组人员执勤天数	5	5
一天任务环数量分布	8	8
二天任务环数量分布	6	6
三天任务环数量分布	6	6
四天任务环数量分布	8	8
总体执勤成本（万元）	51.2	50.4
总体任务环成本（万元）	0.8	0.8
程序运行分钟数(分钟)	130	90

对于A算例的子问题3求解，CPLEX求得的精确解较同样优于改进的遗传算法求得的解。CPLEX求解结果中，108个航班不可以满足最低配置，而98个航班可以满足最低配置；改进的遗传算法求解结果中，只有94个航班可以满足最低配置；两种求解方式中，乘机次数与替补资格使用次数都为0。CPLEX求解结果中最低配置总体飞行时长为905分钟，总体执勤时长1115分钟，因此，机组总体利用率为81.15%，而改进的遗传算法求解结果中最低配置总体飞行时长为795分钟，总体执勤时长985分钟，因此，机组总体利用率为80.40%。两种求解方式的结果中，飞行时间、执勤时间、执勤天数及任务环数量都相近。由于CPLEX求得的精确解中满足最低配置的航班多于改进的遗传算法结果，因此精确解的执勤成本于任务环成本都略高于算法结果，尽管如此，但精确解中满足最低配置的航班更多，因此结果好于改进的遗传算法求得的解。

### 6.3.2 B组算例求解

B组算例求解结果如下表所示，由于CPLEX无法对大规模数据进行求解，因此通过改进的遗传算法算例结果如下：

结果指标	子问题3
不满足机组配置航班数	59
满足机组配置航班数	13895
机组人员总体乘机次数	4590
替补资格使用次数	1115
机组总体利用率（%）	81.77
最小一次执勤飞行时长	5.8

---

平均 一次执勤飞行时长	7.3
最大 一次执勤飞行时长	9.5
最小 一次执勤执勤时长	6.3
平均 一次执勤执勤时长	8.8
最大 一次执勤执勤时长	11.5
最小 机组人员执勤天数	3
平均 机组人员执勤天数	3.7
最大 机组人员执勤天数	4
一天 任务环数量分布	28
二天 任务环数量分布	3498
三天 任务环数量分布	5923
四天 任务环数量分布	4446
总体执勤成本（万元）	2525.8
总体任务环成本（万元）	40
程序运行分钟数(分钟)	199

对于B算例的子问题3求解，存在59个航班不可以满足最低配置，其余13895个航班都可以满足最低配置。机组总体利用率为81.77%。总体执勤成本为2525.8万元，总体任务环成本为40万元。

---

## 第七章 模型的评价

### 7.1 模型评价

首先基于上述问题中所构建的模型和模型运算的结果，通过对模型求解的算法进行分析和回顾，找出改进办法提高算法的效率。然后结合所建模型的参数，通过改变和调整参数，选取对求解航班任务人员分配影响较大的参数，对参数进行调整，从而有效提高模型和算法的求解能力。

#### 7.1.1 模型的优点

- 1、建立了线性优化模型，明确表达飞行时间、执勤时间、休息时间等约束，把航班直接分配给机组人员；
- 2、层次分析法计算权重，对不同目标进行权重分析，适合求解多目标优化问题；
- 3、模型中的决策变量均是采用0-1变量来表示分配情况，简洁明了；
- 4、本文基于线性加权组合的思想，将问题一、二、三中的多个目标转化为单目标，极大的降低到了模型的复杂度，提高了计算效率。

#### 7.1.2 模型的缺点

- 1、本文所有模型对应的都是一个 NP 难问题，对精确方法求解做出了挑战，庆幸的是进化算法是当前一种解决此类问题的有效方法。
- 2、对于机组员工分配给航班的编码方式中，在航班多的情况下编码规模会变的很大，算法不易求解。



---

## 参考文献

- [1]向杜兵. 航空机组排班计划一体化优化研究[D].北京交通大学,2020.
- [2]Zeighami Vahid,Saddoune Mohammed,Soumis François. Alternating Lagrangian Decomposition for Integrated Airline Crew Scheduling Problem[J]. European Journal of Operational Research,2020(prepublish):
- [3]Lei Zhou,Zhe Liang,Chun-An Chou,Wanpracha Art Chaovaitwongse. Airline planning and scheduling: Models and solution methodologies[J]. Frontiers of Engineering Management,2020,7(5):
- [4]Chutima Parames,Arayikanon Kanokbhorn. Many-objective low-cost airline cockpit crew rostering optimisation[J]. Computers & Industrial Engineering,2020,150:
- [5]吕建飞. 航空公司飞行机组人员分配问题研究[D].中国民用航空飞行学院,2015.

---

## 附录：

在MATLAB2018a环境下，使用yalmip调用cplex进行求解模型代码如下：

```
tic
clc;
clear;
yalmip('clear');
%% 参数
% ai=csvread('data_a_person.csv',1,5,'F2..F22');
% bi=load('Captain.mat');
load('ai.mat');
load('bi.mat');
load('pj.mat')
M=1000000000;
ct=40;
alpha=0.5;
beta=0.3;
gamma=0.2;
blk=600;
dp=660;
rest=720;
tafb=14400;
H=1;
I=21;
J=206;
T=1;
pj=pj(1:J,:);
dh=5;
%% 变量
xijt = binvar(I,J,T,'full');
yj = binvar(1,J,'full');
zij=binvar(I,J,'full');
kij=binvar(I,J,'full');
mij=binvar(I,J,'full');
nij=binvar(I,J,'full');
lijjt=binvar(I,J,J,T,'full');
ui=intvar(1,I,'full');

%%子问题1 目标函数及约束
%% 目标函数
sum_yj=0;
sum_kij=0;
sum_zij=0;
```

---

```

for j=1:J
    sum_yj = sum_yj+yj(j);
end
for i=1:I
    for j=1:J
        sum_kij=sum_kij+kij(i,j);
    end
end
for i=1:I
    for j=1:J
        sum_zij=sum_zij+zij(i,j);
    end
end
f=alpha*sum_yj-beta*sum_kij-gamma*sum_zij;
%% 约束条件
F=[];
%约束1
for i = 1 : I
    for j = 1:J
        F= [F,kij(i,j)+mij(i,j)+nij(i,j)==sum(xijt(i,j,:))];
    end
end
%约束2
for i = 1 : I
    for j = 1:J
        F= [F,ai(i)*bi(i)*nij(i,j)==zij(i,j)];
    end
end
%约束3
for j =1:J
    sumxijt = 0;
    for i=1:I
        sumxijt= sumxijt+sum(xijt(i,j,:));
    end
    F= [F,sumxijt<=yj(j)*M];
    F= [F,(ai(:,1))*mij(:,j)==yj(j)];
    F= [F,(bi(:,1))*nij(:,j)==yj(j)];
end
%约束4
for j =1:J
    F= [F,sum(kij(:,j))<=dh];
end
%约束5
for j=1:J

```

---

```

for jj=1:J
    if j~=jj
        for i = 1:I
            for t=1:T
                F=[F,lijjt(i,j,jj,t)*(pj(j,1)-pj(jj,2))==0];
            end
        end
    end
end
end
%约束6
for j=1:J
    for jj=1:J
        if j~=jj
            for i = 1:I
                for t=1:T
                    F=[F,lijjt(i,j,jj,t)*(pj(j,4)+ct)<=pj(jj,3)];
                end
            end
        end
    end
end
%约束7
for j=1:J
    for jj=1:J
        if j~=jj
            for i = 1:I
                for t=1:T
                    F=[F,2*lijjt(i,j,jj,t)<=xijt(i,j,t)+xijt(i,jj,t)];
                end
            end
        end
    end
end
%约束8
for j=1:J
    for i = 1:I
        for t=1:T
            F=[F,sum(lijjt(i,j,.,t))<=1];
        end
    end
end
%约束9
for j=1:J

```

---

```

    for i = 1:I
        for t=1:T
            F=[F,sum(lijzt(i,:,j,t))<=1];
        end
    end
end
%约束10
for i = 1:I
    for t=1:T
        sum2j=0;
        time2j=0;
        sumj2=0;
        timej2=0;
        for j=1:J
            if pj(j,1)==2
                sum2j=sum2j+sum(lijzt(i,j,:,t));
                time2j=time2j+reshape(lijzt(i,j,:,t),1,[])*pj(:,3);
            end
            if pj(j,2)==2
                sumj2=sumj2+sum(lijzt(i,:,j,t));
                timej2=timej2+reshape(lijzt(i,:,j,t),1,[])*pj(:,3);
            end
        end
        F=[F,sum2j==sumj2];
        F=[F,time2j<=timej2];%从基地出发的航班出发时间和小于回到基地的航班的出发
时间和
    end
end
%约束5
for i = 1:I
    F=[F,sum(xijzt(i,:,,:),3)*pj(:,1)==sum(xijzt(i,:,,:),3)*pj(:,2)];
end
%约束
for j = 1:J
    for i=1:I
        F=[F,ai(i)>=mij(i,j)];
        F=[F,bi(i)>=nij(i,j)];
    end
end

ops = sdpsettings('solver','cplex','verbose',1);
ops.cplex.display='on';
ops.cplex.timelimit=18000;

```

---

```

ops.cplex.mip.tolerances.mipgap=0.001;
% 诊断求解可行性
disp('开始求解')
diagnostics=optimize(F,-f,ops);
if diagnostics.problem==0
    disp('Solver thinks it is feasible')
elseif diagnostics.problem == 1
    disp('Solver thinks it is infeasible')
    pause();
else
    disp('Timeout, Display the current optimal solution')
end

%%子问题2目标函数及新增约束
%% 目标函数
sum_liijt=0;
sum_ui=0;
for i=1:I
    for j=1:J
        for jj=1:J
            for t=1:T
                sum_liijt = sum_liijt+liijt(i,j,jj,t)*(pj(j,4)-pj(j,3)+pj(jj,3)-pj(j,4))*ci(i);
            end
        end
    end
end
for i=1:I
    for j=1:J
        sum_kij=sum_kij+kij(i,j);
    end
end
for i=1:I
    sum_ui=sum_ui+uu(i);
end
f=alpha*sum_yj-beta*sum_kij-gamma*sum_zij-sum_liijt-sum_ui;

%%子问题2新增约束
for i=1:I
    for t=1:T
        F=[F,mij(i,:)*(pj(:,4)-pj(:,3))+nij(i,:)*(pj(:,4)-pj(:,3))<=blk];
    end
end
for i=1:I
    for jj=1:J

```

---

```

        for t=1:T
            F=[F,(pj(:,4)-pj(:,3)+pj(jj,3)-pj(jj,4))*lijjt(i,.,jj,t)<=dp];
        end
    end
end
for i=1:I
    for j=1:J
        for jj=1:J
            for t=1:T
                F=[F,xijjt(i,jj,t)*pj(jj,3)-xijjt(i,j,t)*pj(j,4)>=rest];
            end
        end
    end
end
%%
%%子问题3目标函数及新增约束
%% 目标函数
sum_c=0;
sum_balance=0;
for i=1:I
    for j=1:J
        for t=1:T
            sum_c = sum_c+xijjt(i,j,t)*(gih(i)kjh(j)-xijjt(i,j,t)*gih(i)*kjh(j))*cci(i);
        end
    end
end
for i=1:I
    for j=1:J
        sum_balance=sum_balance+kij(i,j);
    end
end
for i=1:I
    sum_ui=sum_ui+uu(i);
end
f=alpha*sum_yj-beta*sum_kij-gamma*sum_zij-sum_lijtt-sum_ui-sum_c-sum_balance;

%%子问题3新增约束
for i=1:I
    for h=1:H
        F=[F,xijjt(i,j,t)*gih(i)kjh(j)-xijjt(i,j,t)*gih(i)*kjh(j)<=tafb];
    end
end
double(f)
toc

```

---

MATLAB遗传算法部分代码如下：

主程序：

```
clear
clc
close all
%遗传参数
popsize1=100;
MAXGEN1=500;
crewNum=21;           % 机组人员数量
chromlength1=206;      % 航班数量
Pc1=0.5;
Pm1=0.1;
hcxx=20;
hczx=20;
ACrew=xlsread('机组排班Data A-Crew.xlsx');
AFlight=xlsread('机组排班Data A-Flight.xlsx');
CC= xlsread('参数集合.xlsx');
%% 生成初始种群
pop1=initGroup1(popsize1,chromlength1, crewNum);
%% 初始化数据;
gen1=0;
zuijia=[];
zymbz=[];
zyjuzhen=[];
while gen1<MAXGEN1
    %% 交叉
    pop1=crossover1(pop1,Pc1);
    %% 变异
    pop1=Mutate1(pop1,Pm1);
    %% 计算适应度
    [objvalue1,pop1]=calobj1(pop1,chromlength1,fztask,hczx);
    zpop1=pop1;%用于最后迭代
    %% 选择
    fitvalue1=zeros(size(objvalue1,1),2);%一行目标函数， 另一方编号
    for i=1 : size(objvalue1,1)
        fitvalue1(i,1)=i;
        fitvalue1(i,2)=1/objvalue1(i,3);
    end
    %计算适应度函数
    fitvalue1=sortrows(fitvalue1,2,'descend');
    zyjuzhen(1,:)=pop1(fitvalue1(1,1)*2-1,:);
    zyjuzhen(2,:)=pop1(fitvalue1(1,1)*2,:);
    newpop=[];
    for i=1:popsize1
        newpop(i*2-1,:)=pop1(fitvalue1(i,1)*2-1,:);
```



---

```
newpop(i*2,:)=pop1(fitvalue1(i,1)*2,:);
end
pop1=newpop;
%% 重插入子代
gen1=gen1+1;
objv=sortrows(objvalue1,3,'ascend');
mdzy=min(objvalue1(:,3));
if gen1==1
    his_zuiyou(gen1,1)=gen1;
    his_zuiyou(gen1,2)=objv(1,1);
    his_zuiyou(gen1,3)=objv(1,2);
    his_zuiyou(gen1,4)=objv(1,3);
    zuijia(gen1*2-1,:)=zyjuzhen(1,:);
    zuijia(gen1*2,:)=zyjuzhen(2,:);
else
    if mdzy>his_zuiyou(gen1-1,2)
        his_zuiyou(gen1,1)=gen1;
        his_zuiyou(gen1,2)=his_zuiyou(gen1-1,2);
        his_zuiyou(gen1,
```