** **

**“华为杯”第十五届中国研究生**

**数学建模竞赛**

|  |  |
| --- | --- |
| **学 校** | **同济大学** |
| **参赛队号** | **18102470112** |
| **队员姓名** | **1. 于 猛** |
| **2. 万有文** |
| **3. 陈 石** |

**“华为杯”第十五届中国研究生**

**数学建模竞赛**

题 目 航站楼扩增影响评估方法研究

摘 要：

针对机场增建卫星厅后如何优化航班-登机口的分配，分析中转旅客的换乘紧张程度，其本质上是多目标最小优化问题。

在考虑了机场登机口、航班与中转旅客换乘特点的基础上，建立了以分配到临时机位的航班数量最少、换乘旅客总体流程时间最小（问题二）或换乘旅客总体换乘紧张度（问题三）最小、被使用固定登机口数量最少为三个优化目标的数学模型。登机口分配问题（Airport Gate Assignment Problem, AGAP）是典型的具有NP-难度的优化问题，本文采用遗传算法对模型进行优化求解。

问题一，只考虑航班-登机口分配。考虑机型和登机口类型匹配的约束，建立了航班-登机口分配模型，采用遗传算法对结果进行寻优，在保证尽可能多的分配航班到合适的固定登机口前提下，最小化被使用登机口的数量。计算结果显示：整体来看，**有251对（502架）航班可被分配到固定登机口，在此前提下最少使用65个固定登机口。**

问题二，需同时考虑中转旅客最短流程时间。在问题一的基础上，考虑中转旅客换乘因素，加入中转旅客总体最短流程时间最小的目标函数，其优先级次于尽可能多的分配航班到合适的登机口，高于最小化被使用固定登机口数量。在问题一的基础上引入总体旅客流程时间最小的目标函数，使用遗传算法对结果进行寻优。计算结果显示：**502架航班可被分配到固定登机口，在此前提下中转旅客总体最短流程时间为67745分钟，此时被使用登机口数量最少为67个。**

对于问题三，需同时考虑中转旅客的换乘时间。在问题二的模型上进一步细化，旅客换乘时间在最短流程时间的基础上加入中转旅客捷运时间与行走时间。采用遗传算法对模型进行求解，计算结果显示：**502架航班可被分配到固定登机口，在此前提下中转旅客整体换乘紧张度最小为 ，此时被使用登机口数量最少为 。**

本文所建立的模型贴近题目要求，目标函数与约束条件清晰明确。模型求解方法清晰合理，仿真计算也取得了较好的效果。结果合理可行，具有一定的参考价值。

**关键词：**登机口分配；旅客中转；换乘紧张度；遗传算法。

目录

[一、问题重述 3](#_Toc525107726)

[二、问题分析 5](#_Toc525107727)

[2.1 问题一的分析 5](#_Toc525107728)

[2.2 问题二的分析 5](#_Toc525107729)

[2.3 问题三的分析 5](#_Toc525107730)

[三、模型假设 7](#_Toc525107731)

[四、符号说明 8](#_Toc525107732)

[五、模型建立与求解 9](#_Toc525107733)

[5.1 问题一的建模与求解 9](#_Toc525107734)

[5.1.1 模型建立 9](#_Toc525107735)

[5.1.2 模型求解 10](#_Toc525107736)

[5.1.3 结果与分析 14](#_Toc525107737)

[5.2 问题二的建模与求解 17](#_Toc525107738)

[5.2.1 模型建立 17](#_Toc525107739)

[5.2.2 模型求解 17](#_Toc525107740)

[5.2.3 结果与分析 17](#_Toc525107741)

[5.3 问题三的建模与求解 22](#_Toc525107742)

[5.3.1 模型建立 22](#_Toc525107743)

[5.3.2 模型求解 23](#_Toc525107744)

[5.3.3 结果与分析 23](#_Toc525107745)

[六、模型的总结与评价 27](#_Toc525107746)

[七、参考文献 28](#_Toc525107747)

[附录： 29](#_Toc525107748)

# 一、问题重述

为了解决机场旅客流量饱和问题，航空公司正在增设卫星厅。引入卫星厅后，虽然可以缓解原有航站楼登机口不足的压力，但是对中转旅客的航班衔接具有一定的负面影响。本题要求建立数学模型，优化分配登机口，分析中转旅客的换乘紧张程度，为航空公司航班规划的调整提供参考依据。

**航站楼T和卫星厅S的布局和功能。**T具有完整的国际机场航站功能，包括出发、到达、出入境和候机。卫星厅S是航站楼T的延伸，可以候机，没有出入境功能。现统称航站楼T和卫星厅S为终端厅。T和S之间有捷运线相通，可以快速往来运送国内、国际旅客。假定旅客无需等待，随时可以发车，单程一次需要8分钟。航站楼T有28个登机口，卫星厅S有41个登机口。具体配置见数据表格。

**航班-登机口的分配需要考虑如下规则：**

1、T和S的所有登机口统筹规划分配。

2、每个登机口的国内/国际、到达/出发、宽体机/ 窄体机等功能属性事先给定，不能改变，具体配置详见数据部分。飞机转场计划里的航班只能分配到与之属性相吻合的登机口。

3、每架飞机转场的到达和出发两个航班必须分配在同一登机口进行，其间不能挪移别处。

4、分配在同一登机口的两飞机之间的空挡间隔时间必须大于等于45分钟。

5、机场另有简易临时机位，供分配不到固定登机口的飞机停靠。假定临时机位数量无限制。

本题数据中使用到的宽窄飞机型号分别有：

宽体机（Wide-body）：332, 333, 33E, 33H, 33L, 773；

窄体机（Narrow-body）：319, 320, 321, 323, 325, 738, 73A, 73E, 73H, 73L。

**本题研究新建卫星厅对中转旅客的影响。**中转旅客从前一航班的到达至后一航班的出发之间的流程，按国内（D）和国际（I）、航站楼（T）和卫星厅（S）组合成16种不同的场景，各种场景最短流程时间和捷运乘坐次数如表1-1所示。其中每一格的第一个数是最短流程时间（分钟），第二个数是捷运乘坐次数。捷运时间和旅客行走时间不计入最短流程时间。

表1-1 最短流程时间和捷运乘坐次数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 出发  到达 | | 国内出发（D） | | 国际出发（I） | |
| 航站楼T | 卫星厅S | 航站楼T | 卫星厅S |
| 国内到达  （D） | 航站楼T | 15/0 | 20/1 | 35/0 | 40/1 |
| 卫星厅S | 20/1 | 15/0 | 40/1 | 35/0 |
| 国际到达  （I） | 航站楼T | 35/0 | 40/1 | 20/0 | 30/1 |
| 卫星厅S | 40/1 | 45/2 | 30/1 | 20/0 |

**问题一：本题只考虑航班-登机口分配。**作为分析新建卫星厅对航班影响问题的第一步，首先要建立数学优化模型，其目标为：（1）尽可能多地分配航班到合适的登机口；（2）在（1）的基础上最小化被使用登机口的数量。本问题不需要考虑中转旅客的换乘，但要求把建立的数学模型进行编程，求最优解。

**问题二：考虑中转旅客最短流程时间。**本问题是在问题一的基础上加入旅客换乘因素，即目标（1）尽可能多地分配航班到合适的登机口；目标（2）最小化中转旅客的总体最短流程时间；目标（3）最小化被使用登机口的数量。本题不考虑旅客乘坐捷运和步行时间，但也要求编程并求最优解。

**问题三：考虑中转旅客的换乘时间。**如前所述，新建卫星厅对航班的最大影响是中转旅客换乘时间的可能延长。因此，数学模型最终需要：（1）尽可能多地分配航班到合适的登机口；（2）考虑换乘旅客总体紧张度的最小化；（3）最小化被使用登机口的数量。本问题可以在问题二的基础上细化，引入旅客换乘连接变量，并把中转旅客的换乘紧张度作为一个目标函数。和前面两个问题一样，本问题也要求把建立的数学模型进行编程，并求最优解。

换乘紧张度定义：

 （1-1）

 （1-2）

 （1-3）

# 二、问题分析

本题属于机场登机口分配问题（Airport Gate Assignment Problem, AGAP），需要根据不同任务要求及约束条件构建模型，然后通过优化算法进行求解寻优。

本题考虑航班类型、登机口类型、停机时间、中转旅客换乘时间等多方面因素，属于NP难度问题。目前，求解该类问题的方法研究有启发式算法、整数规划法、专家系统、网络流、分支定界法等。由于该类问题的计算复杂度高，在求解大规模问题时，采用精确解法进行求解会遇到维数灾难，计算复杂性问题突出。

建立模型时，需要考虑目标函数与约束条件。

**其核心目标是：**

（1）尽可能多的航班分配到固定登机口，即尽可能少地分配航班到临时机位。

（2）最小化问题三的总体紧张度，或者最小化问题二的总体流程时间。

（3）最小化被使用的固定登机口数量。

**问题涉及的约束条件包括：**

（1）航班机型与登机口型号相匹配，即指定机型只能停靠到可容纳该机型的登机口。

（2）任意一个航班必须且仅能被安排到一个登机口或临时机位。

（3）同一登机口在同一时刻最多分配给一个航班。

（4）同一登机口的前后两架飞机间隔时间必须大于等于45分钟。

## 2.1 问题一的分析

本题为仅考虑宽体机/窄体机、国内/国际、到达/出发机型匹配因素的AGAP问题。由于只进行航班-登机口分配，并且宽体机/窄体机与可容纳宽体机/窄体机的登机口需要严格匹配，因此可以将宽体机与窄体机的航班分开单独计算。本题也不涉及航班在航站楼T和卫星厅S的登机口进行分配，故对航站楼T和卫星厅S也不进行区分。由于首要目标是最小化被分配到临时机位的航班数量，因此不能采用“先到先服务”原则，否则不一定能得到全局最优解。因此，即使航班到达时有合适的登机口可以停靠，也有可能为了全局最优而被分配到临时机位。

按照优先级从高到低排序，问题一的优化目标为：

（1）尽可能多的航班分配到固定登机口，即尽可能少地分配航班到临时机位。

（2）最小化被使用的固定登机口数量。

## 2.2 问题二的分析

在问题一的基础上，加入最小化总体流程时间目标。问题一的约束条件仍然适用于本题。本题的换乘时间只考虑中转旅客的最短流程时间，其值可由数据表格查出。由于登机口的航站楼T和卫星厅S属性对最短流程时间有影响，因此，问题二必须要区分T、S终端厅的区别。本题应当考虑换乘失败的乘客，即换乘时间（问题中即为最短流程时间）大于乘客出发航班的出发时刻与到达航班的到达时刻之差。对换乘失败的乘客记录要考虑在总体流程时间上进行惩罚。

按照优先级从高到低排序，问题二的优化目标为：

（1）尽可能多的航班分配到固定登机口。

（2）最小化中转旅客的总体流程时间。

（3）最小化被使用的固定登机口数量。

## 2.3 问题三的分析

本题在问题二的基础上进一步细化，其中换乘旅客的换乘时间除了最短流程时间之外，还要考虑捷运时间和行走时间。行走时间不仅要考虑航站楼T、S的区别，还要细化到各个厅的各个区域，其值可通过题中的表格查询。同时，本题需要考虑换乘失败的乘客，并对其记录进行惩罚以反映在整体换乘紧张度上。

按照优先级从高到低排序，问题三的优化目标为：

（1）尽可能多的航班分配到固定登机口。

（2）最小化中转旅客的整体换乘紧张度。

（3）最小化被使用的固定登机口数量。

# 三、模型假设

1、假设T和S的所有登机口统筹规划分配，同时使用。

2、每个登机口的国内/国际、到达/出发、宽体机/窄体机等功能属性事先给定，不能改变。飞机转场计划里的航班只能分配到与之属性相吻合的登机口。

3、每架飞机转场的到达和出发两个航班必须分配在同一登机口进行，期间不能挪移别处。

4、分配在同一登机口的两飞机之间的空挡间隔时间必须大于等于45分钟。

5、机场另设置有临时机位，供分配不到固定登机口的飞机停靠，且临时机位数量无限制。

6、换乘失败是指没有足够换乘时间的情形。旅客换乘失败会给航空公司造成很大损失，故在此设定其真实换乘时间为6小时。对停靠临时机位的旅客的换乘时间忽略不计，而不算作换乘失败。

7、总体紧张度为单个乘客紧张度之和。

8、问题二中的中转旅客换乘时间即为最短流程时间。

9、采用离散模型，离散时间步长为1分钟。

# 四、符号说明

|  |  |
| --- | --- |
| **符号** | **意义** |
|  | 航班*i*的到达时刻 |
|  | 航班*i*的出发时刻 |
|  | 航班*i*达到登机口k的时刻 |
|  | 航班*i*离开登机口k的时刻 |
|  | 同一登机口内上一次航班离开到下一次航班达到之间的最小时间间隔 |
|  | 航班*i*的机型 |
|  | 登机口*k*的类型 |
|  | 二进制变量，当且仅当航班*i*被分配到登机口*k*时等于1，否则等于0 |
|  | 二进制变量，当且仅当航班*i*和*j*均被分配到登机口*k*，且航班*j*恰好在航班*i*后被分配到登机口*k*时等于1，否则等于0 |
|  | 二进制变量，若有航班分配到固定登机口*k*中时=1，否则=0。 |
|  | 从到达航班*i*转移到出发航班*j*的中转旅客数量 |
|  | 中转旅客从航班*i*换乘到航班*j*的换乘时间 |
|  | 中转旅客从航班*i*换乘到航班*j*的最短流程时间 |
|  | 中转旅客乘坐一次捷运的时间，8分钟 |
|  | 中转旅客从航班*i*换乘到航班*j*乘坐捷运的次数 |
|  | 登机口*k*与登机口*l*之间的行走时间，显然有= |

# 五、模型建立与求解

## 5.1 问题一的建模与求解

### 5.1.1 模型建立

问题一是只考虑登机口数量受限的航班-登机口匹配问题。

由于登机口数量有限，首先考虑目标函数，即分配到临时机位的航班数量最少。其次考虑目标函数，即被使用的固定登机口数量最少。为保证分配到临时机位的航班对全局影响最小，是全局最优目标。

本题需要严格考虑登机口-航班的匹配关系。同时，同一个登机口前后两架飞机之间的空档时间必须大于等于分钟。分配不到固定登机口的飞机停靠到临时机位，其中临时机位数量无限制。

假设机场中共有个登机口可供使用，其集合为

同时把临时机位视为第个登机口，且可以停放无限多架航班，而固定登机口同一时刻只能为一架航班服务。在考虑的时间段之内，共有*N*架航班对到达/出发，航班按照到达时间顺序排序，集合为

这样，第个登机口（即临时机位）的航班即为未分配到固定登机口的航班，其数量最小化是最优先的目标。

约束条件分析：

（1）对于任意一个航班必须且仅能被安排到一个登机口（包含临时机位）。设为二进制变量，若航班被分配到登机口中，则，否则。那么，假设登机口数量为，航班数量为，则约束条件(1)为：

因为临时机位被视为第个登机口，且其可以停放无限多架航班，故不受此限制。

（2）登机口独占性原则。即同一登机口在同一时刻最多分配给一个航班。分配到登机口的每个航班的前后均最多只能有一个航班。假设航班被分配给了登机口，则。设为二进制变量，当且仅当航班和均被分配到登机口，并且航班恰好先于航班时等于，否则等于。则约束条件(2)和(3)可以表述为：

（3）登机口类型与机型匹配性原则。机场登机口有宽（W）/窄（N）之分，有到达/出发的国内（D）/国际（I）之分。航班有宽体机（W）/窄体机（N）之分，有到达/出发的国内（D）/国际（I）之分。按照宽/窄、到达的国内/国际、出发的国内/国际可以描述为：航班类型F(W/N,D/I,D/I)，机场登机口类型G(W/N,D/I/DI,D/I/DI)，其中DI的登机口可以容纳D或I的任意类型航班，其他条件必须严格匹配。

（4）满足最短间隔时间要求。同一登机口的两架飞机间隔时间必须大于等于分钟，即同一登机口的前一个航班出发之后45分钟及以上后另一个航班才可以再停靠在该登机口。用表示航班离开登机口的时刻，表示航班达到登机口的时刻，且航班恰好在航班之后停靠在登机口，则有：

临时机位（第个登机口）不受此约束。

目标函数分析：

优化目标是分配到临时机位的航班数量最少，在此基础上最小化被使用的固定登机口数量。首先考虑目标函数，以保证分配到临时机位的航班数量最少。同样，设为二进制变量，若航班被分配到登机口中，则，否则。把临时机位视为第个登机口，若航班被分配到临时机位，则，否则。其次考虑目标函数，其计算中已经考虑了分配航班到临时机位的情形。设为二进制变量，若有航班分配到固定登机口中时，否则。

因此，问题一的目标函数可以表述为：

### 5.1.2 模型求解

利用遗传算法，求得本体条件下模型的最优解。遗传算法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。遗传算法的基本思路是从优化问题的一个种群（一组可行解）开始，按照适者生存与优胜劣汰的原则，逐代进化产生出越来越好的一个种群，即一组可行解。

遗传算法中代表种群中的每个个体差异的表示为遗传染色体。按照上述模型的编码方法，一条遗传染色体代表一组可行解。在本题中，在满足题目约束的条件下，初始化随机产生组可行解作为初始种群。初始种群经过交叉变异等过程逐渐产生最优解。算法的执行过程如下。

1、初始化种群大小，选择概率，交叉概率和变异概率等参数，在要求的各种约束条件下，随机生成初始种群。

2、根据已建模好的优化函数，对应计算种群中每个个体的适应度；

3、按照遗传策略，对第代种群进行选择操作、交叉操作和变异操作，形成下一代的种群；

4、通过种群中个体适应度的最大值，判断算法是否满足停止准则，如果不满足，则返回2；如果满足，则输出种群中的最大适应度值的个体作为最优解,终止计算。具体流程如图所示。

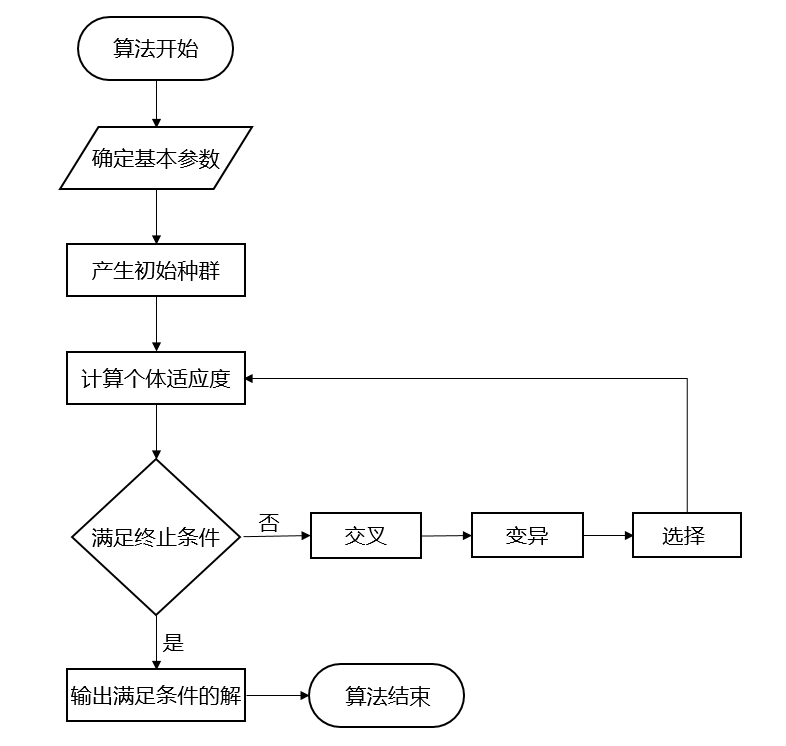


图5-1 遗传算法的基本流程

在本题中，每一个航班的停机口选择作为一个基因节点，这些节点共有个。对每一个基因节点，可能的取值可以有种。在实际求解的过程中可以对模型进行等效的简化，即对每一个节点不使用二值编码（需占用个比特位）而采用整数编码（需占用个比特位）。其中，染色体编码的第位整数编码的值即代表了按时间顺序排序的第个航班选择的登机口编号。例如编码1 3 5 4 2 4 5 2，它表示了8个航班在5个登机口上的分配方案：第1架航班分配到第1号登机口，第2架航班分配到第3号登机口，第3架航班分配到第5号登机口，以此类推。但是，第6架航班到达时，占据第4号登机口的第4架航班已经解除占用，所以它可以被分配到第4号登机口。

初始解的生成过程是一个有约束的随机过程，即随机的把每个航班安排到一个可行的登机口上。这里的可行的登机口是指满足国内/国际约束和机型约束等条件的登机口。为了在初始阶段能产生较为理想的个体，在随机产生初始解的过程中需要进行筛选，将随机产生的初始解中适应度较差的一部分解丢弃一部分。

在算法的执行期间，不仅是在初始解的生成过程中，也包括在遗传迭代过程中的交叉和变异操作，都将确保所有约束一定满足。交叉操作又叫做基因重组，是指把两个父代染色体的部分结构加以替换重组而生成新的染色体的操作。在本题中，染色体的交叉操作是得到新染色体的主要方法。通过染色体的交叉可以得到解空间中其他的可行解，并保留当前解中不一定能显式表示的优良特征。对两个长度为的染色体来说，交叉操作是将某个随机位置之后的基因进行交换，如图所示。

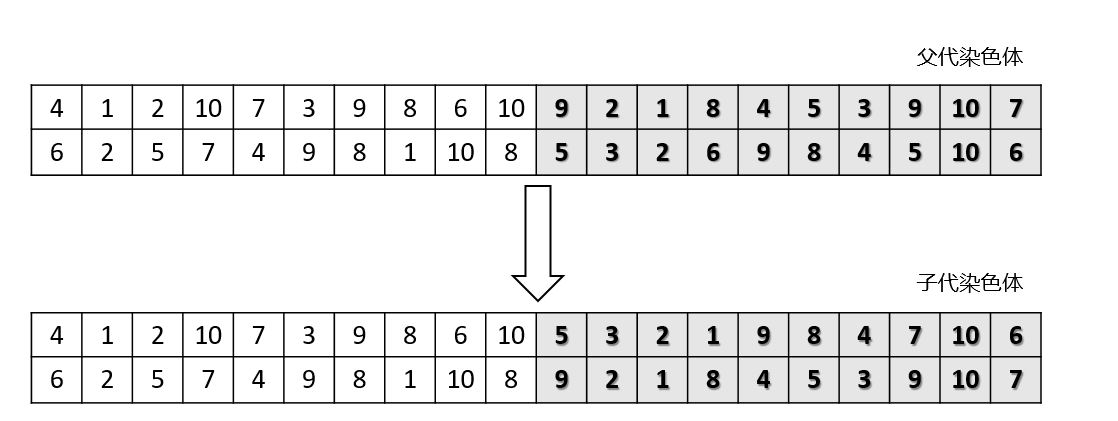


图5-2 染色体交叉

被部分交换的两个染色体被称为父代染色体，交叉产生的新染色体称为子代染色体。此时，即使父代染色体是一种满足约束条件的排列，子代染色体的排列顺序也未必能满足约束。因此，对于每对进行了交叉操作的染色体和，都需要对其进行约束检查。具体是在交叉过程中记录下交叉的位置，因为原父代染色体是满足约束条件的，因此对位置在到的基因排列，是已经满足约束的。对于经过交换的位置大于等于的那些基因，需要按顺序进行约束检查。如图所示。

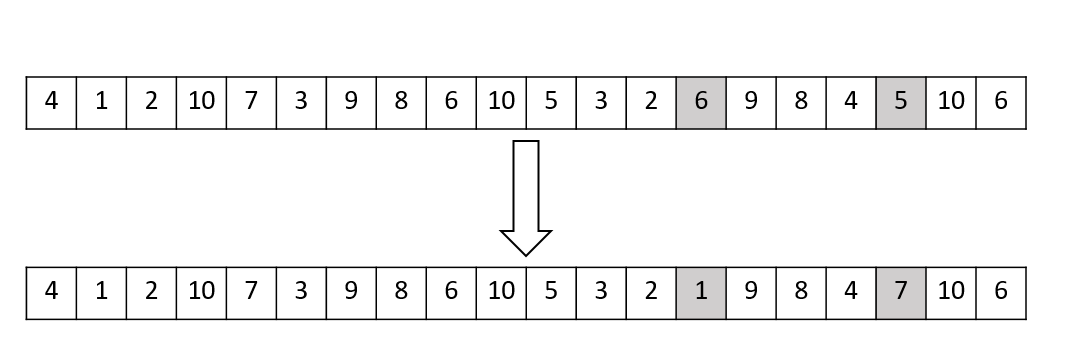


图5-3 染色体变异与约束检查

对每个进行检查的基因，若该基因满足约束条件，则保留原值，若不满足约束条件，则在对应的状态条件和约束条件下，按照登机口分配的原则，给该位置上的基因赋值，直到染色体结束。在上图中，从第12个基因位置开始检查，第12个航班被分配到第5号登机口，由于该登机口之前未被分配航班，检查出发/到达类型后其满足条件，故其值可以保持不变而不违反约束条件。依次向右检查，检查到第15个基因位置时，发现在当前的状态条件下，6号登机口已经被之前的飞机占用且尚未离开，因此该基因不满足条件，应考虑在当前条件下所有可以被分配的满足条件的登机口，随机选择一个登机口分配给第15个航班，图中分配了1号登机口。之后继续检查，直到染色体结束。交叉操作的后一部分的约束条件检查其实就是变异操作。交叉和变异是自然界遗传和进化过程中的重要部分，通过这两个流程，加以选择，后代越来越可能得到适应度高的个体。

**算法的具体步骤**

1、定义算法的基本参数

2、建立初始种群

初始化航班集合，其包含所有的航班，同时初始化登机口集合，其包含所有的登机口。

判断当前航班集合是否为空，若不为空，则随机选取一个航班，根据该航班的到达时刻，出发时刻，到达类型和出发类型，确定满足该航班约束条件的登机口集合。若航班集合为空，转。

判断是否为空集，若不为空，则随机从中选取一个登机口分配给该航班，若为空，则将临时登机口分配给该航班，同时将该航班从待分配的航班集合中移除，转。

记录当前航班分配方案的染色体中的数据，并以此计算出该方案所占用的临时登机口数量和空闲的固定登机口数量，这些是为计算适应度函数所需的变量。

按的方法，重复产生个个体，选取其中适应度最好的个体加入种群，并按照这种方式建立个体个数为的初始种群。

3、交叉与变异

按照已确定的交叉参数，在当前种群集合中随机选择待交叉的个体集合。

将中的个体随机两两组合，对每组个体，在随机产生的位置，进行染色体交叉操作，交换两个染色体种位置在之后的基因。

对每条经过交叉操作的染色体，逐个检查在交换位置之后的基因是否满足约束条件，若满足，则保留原值，若不满足，则在对应的状态条件和约束条件下重新分配，直到染色体结束。

按照已确定的变异参数，在当前种群集合中随机选择待交叉的个体集合。

对中的每个个体，查询所有基因位中有可能有其他取值的基因位集合，并且在该集合中随机产生一个位置，进行变异操作，在该位置的基因随机变异成其他的取值，而非原来的值。

对每条经过变异操作的染色体，逐个检查在变异位置之后的基因是否满足约束条件，类似。

4、选择与进化

采用确定性的选择策略，即选择目标函数值最大的一部分个体进化到下一代。按照已确定的选择概率，在当前种群中选择适应度最低的一部分个体，将其淘汰。

经过上述过程，种群已经进化成了新的一代。在达到最大进化代数之后，可以选择出最优秀，即适应度最高的个体，作为本题的解。

问题一的求解，需要用到原始数据中Pucks表和Gates表中的数据，Pucks表中的一行记录代表一次飞机转场记录。包含有该飞机的到达日期及时刻，到达航班号，到达类型，飞行型号，出发日期及时刻，出发航班号，出发类型等具体数据。由题意，飞机转场计划和中转旅客信息包含2018年1月19、20、21三天，但本次竞赛只需要对20日到达或20日出发的航班和旅客进行分析。因此，经过时间筛选后的航班转场记录表共包含303条记录。Gates表中包含每一个固定登机口的数据，包含有该登机口所能接受的飞机的到达类型，出发类型，终端厅类型（航站楼/卫星厅），机体类别和所在区域等。本题的求解过程，需要在这两张表所关联形成的约束条件下进行。

### 5.1.3 结果与分析

计算结果显示：**有251对（502架）航班可被分配到固定登机口，在此前提下最少使用65个固定登机口。**换言之，有52对（104架）航班被分配到临时机位，有4个固定登机口空闲。具体分配方式见附件表格。

**对于宽体机（W）来说**，最多有49对（98架）航班被分配到固定登机口，成功分配到固定登机口的航班所占比例为100%。最少被使用固定登机口的数量为20个，即空闲了4个固定登机口。

**对于窄体机（N）来说，**最多有202对（404架）航班被分配到固定登机口，成功分配到固定登机口的航班所占比例为79.53%。最少被使用登机口的数量为45个，即没有空闲登机口。

**注意：**这里的统计结果是按航班数量计算的，即每个飞机转场记录对应两个航班，及1对航班就是2架航班。

如图5-4所示，给出了成功分配到登机口的航班数量及其比例，并按照宽、窄体机进行了区分。

图5-4 成功分配到登机口的航班数量和比例

或表示为图5-5所示的柱状图。

|  |  |
| --- | --- |
|  |  |

图5-5 成功分配到登机口的航班数量和比例

**注意：**这里的统计结果是按航班数量计算的，即每个飞机转场记录对应两个航班。

由于求得的最优解存在多种航班-登机口分配方式，这里展示如下几种示例结果。

**1、T-S登机口的使用数目分别为28-37：**

（1）宽体机停靠在T登机口和S登机口的数目分别为11、9，窄体机停靠在T登机口和S登机口的数目分别为17、28（第1种分配方式）。

**此时平均使用率：T-S登机口分别为60.79%-62.15%**。

（2）宽体机停靠在T-S登机口的数目为11-9，窄体机停靠在T登机口和S登机口的数目为17、28（第2种这样的分配方式，虽然与上面数量相同，但是具体分配到哪一个登机口的方式不同）。

**此时平均使用率：T-S登机口分别为63.06%-60.25%**。

**2、T-S登机口的使用数目分别为26-39：**

（1）宽体机停靠在T-S登机口的数目为9-11，窄体机停靠在T-S登机口的数目为17-28（第1种分配方式）。

**此时平均使用率：T-S登机口分别为59.19%-63.15%。**

（2）宽体机停靠在T-S登机口的数目为9-11，窄体机停靠在T-S登机口的数目为17-28（第2种分配方式，解释同上）。

**此时平均使用率：T-S登机口分别为61.63%-61.34%。**

用表格表示如下所示。

表5-1 登机口使用数量

|  |  |  |
| --- | --- | --- |
| 登机口使用数量 | 分配方式一 | 分配方式二 |
| T登机口 | 28 | 26 |
| S登机口 | 37 | 39 |

表5-2 登机口平均使用率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 平均使用率 | 结果一 | 结果二 | 结果三 | 结果四 |
| T登机口 | 60.79% | 63.06% | 59.19% | 61.63% |
| S登机口 | 62.15% | 60.25% | 63.15% | 61.34% |

以上T-S登机口的分配结果如图所示。

图5-6 两种分配方式下T-S登机口的使用数量

图5-7 四种分配结果下T-S登机口的平均使用率

结果显示，问题二的最优航班-登机口分配方式下，T-S登机口数量的分配方式多于一种。但不管哪种分配方式，T-S登机口的平均使用率均在60%左右。

## 5.2 问题二的建模与求解

### 5.2.1 模型建立

问题二在问题一的基础上，加入中转旅客整体流程时间最少的目标。

**按照优先级自高到低排序，问题二的优化目标为：**

（1）：尽可能多的航班分配到固定登机口。

（2）：最小化中转旅客的总体流程时间。

（3）：最小化被使用的固定登机口数量。

与问题一中一样，设为二进制变量，若航班被分配到登机口中，则，否则。用表示从航班换乘到航班的换乘时间，问题二中即为最短流程时间。用表示从到达航班转移到出发航班的旅客数量，则表示：有个乘客从航班换乘到航班，每个乘客换乘时间为*，*并且航班分配到登机口中，航班分配到登机口中。所有的总和，即为最短流程时间。

约束条件分析：约束条件和问题一相同。

目标函数分析：

问题二的目标函数，在问题一的基础上，加入了换乘时间最短的目标函数，其优先级在问题一的两个目标函数的优先级之间。

### 5.2.2 模型求解

问题二的模型和问题一的模型大致相同，只是目标函数的数量和优先级有所变化。因此，在总体上可以延续问题一的求解思路。即利用遗传算法，在交叉、变异、选择和进化的过程中得到最优的解。值得注意的是，由于引入了乘客换乘这一个因素，需要用到原数据中的Tickets表。Tickets表中的每一行代表一位乘客的换乘经历。因此，本小题的求解的第一步，就是通过Tickets表和Pucks表和Gates表的关系，获取所需的数据。

对于问题二的优化目标函数来说，和都已经在问题一中综合考虑进行求解。由于优化目标有着明确的优先级的区别，而这些优化的目标函数都可以显式的表示为适应度函数。因此对问题二的求解，另一个重要步骤是对问题一求解的适应度函数进行改进，使其考虑目标函数，其余的求解过程与问题一的求解过程类似。

### 5.2.3 结果与分析

计算结果显示：**502架航班可被分配到固定登机口，在此前提下中转旅客总体最短流程时间为67745分钟，此时被使用固定登机口数量最少为67个。**

如图所示，给出了成功分配到登机口的航班数量及其比例，并按照宽、窄体机进行了区分。

结果显示，航班分配的情况如下：宽体机成功分配到固定登机口的航班数量为98架，占比100%；窄体机成功分配到固定登机口的航班数量为404架，占比仅为79.53%。

图5-8 成功分配到登机口的航班数量和比例

图5-9 登机口使用数量与平均使用率

登机口使用情况方面，航站楼T的登机口使用数量为28个，其平均使用率为61.95%；卫星厅S的登机口使用数量为39个，其平均使用率为58.16%。

图5-10 换乘失败乘客的数量和比率

由于问题二的换乘时间仅考虑最短流程时间，其最大值也仅为45分钟。结果显示换乘失败乘客的数量为0，即没有人换乘失败。

总体旅客换乘时间方面，15分钟以下的占比为0%，45分钟以内换乘的比例为100%。具体如图5-11、5-12所示。

图5-11 总体旅客换乘时间分布图（堆积图）

上图含义为：横轴数据的时间以内，中转旅客的比率为纵轴坐标值。如，20分钟以内的中转旅客比率为22.21%。或者用下图表示，即用每个时间段内中转旅客的比率来展现。

图5-12 总体旅客换乘时间分布图（分段图）

上图含义为：对应横轴数据的每个时间段，其中转旅客的比率为纵轴坐标值。如，换乘时间在15~20分钟的中转旅客展全部中转旅客数量的比率为21.46%。

图5-13 中转旅客换乘紧张度分布（堆积图）

中转旅客换乘紧张度方面，绝大多数中转旅客的换乘紧张度处于0.1~0.3之间，并且全部都小于0.5。具体情况如图5-13、5-14所示。

图5-14 中转旅客换乘紧张度分布（分段图）

## 5.3 问题三的建模与求解

### 5.3.1 模型建立

问题三在问题二的基础上，又考虑了中转旅客的换乘紧张度。

**按照优先级排序，问题三的优化目标为：**

（1）*f1*：尽可能多的航班分配到固定登机口。

（2）*f2*：最小化中转旅客的整体换乘紧张度。

（3）*f3*：最小化被使用的固定登机口数量。

与问题一中一样，设为二进制变量，若航班被分配到登机口中，则，否则。用表示从航班换乘到航班的换乘时间，问题三中的旅客换乘时间为最短流程时间、捷运时间、行走时间之和，公式表述为：

其中，分别是到达和出发的登机口，捷运时间等于单次捷运时间（本题中为8分钟）乘以捷运次数，即

并且，遇到换乘失败的情况，其换乘时间假设为6小时。用表示从到达航班转移到出发航班的旅客数量，则表示：有个乘客从航班换乘到航班，每个乘客换乘时间为*，*并且航班分配到登机口中，航班分配到登机口中。乘客的换乘紧张度定义为

约束条件分析：约束条件和问题一相同。

目标函数分析：

其中，分钟，且

### 5.3.2 模型求解

与问题二的模型相比，问题三的模型的目标函数有所不同。同时，由于多考虑了换乘中影响时间的的各种因素，使得在优化过程中换乘的时间相对延长了，因此，可能存在换乘失败的问题，即对于某一位乘客来说，他乘坐的到达航班的到达时间和他要乘坐出发航班的出发时间之间的差值大于其换乘时间，即换乘紧张度小于1时，换乘成功，反之换乘失败，作为惩罚，在这种情况下，换乘时间置为6小时。对于问题三的求解，在总体上可以延续问题二的求解思路。即利用遗传算法，在交叉、变异、选择和进化的过程中得到最优的解。

### 5.3.3 结果与分析

计算结果显示：**502架航班可被分配到固定登机口，在此前提下中转旅客总体紧张度为0.3103，此时被使用固定登机口数量最少为67个。**

如下图所示，给出了成功分配到登机口的航班数量及其比例，并按照宽、窄体机进行了区分。

结果显示，航班分配的情况如下：宽体机成功分配到固定登机口的航班数量为98架，占比100%；窄体机成功分配到固定登机口的航班数量为404架，占比仅为79.53%。

图5-8 成功分配到登机口的航班数量和比例

图5-9 登机口使用数量与平均使用率

登机口使用情况方面，航站楼T的登机口使用数量为28个，其平均使用率为61.95%；卫星厅S的登机口使用数量为39个，其平均使用率为58.16%。

图5-10 换乘失败乘客的数量和比率

由于问题二的换乘时间仅考虑最短流程时间，其最大值也仅为45分钟。结果显示换乘失败乘客的数量为0，即没有人换乘失败。

总体旅客换乘时间方面，15分钟以下的占比为0%，45分钟以内换乘的比例为100%。具体如图5-11、5-12所示。

图5-11 总体旅客换乘时间分布图（堆积图）

上图含义为：横轴数据的时间以内，中转旅客的比率为纵轴坐标值。如，20分钟以内的中转旅客比率为22.21%。或者用下图表示，即用每个时间段内中转旅客的比率来展现。

图5-12 总体旅客换乘时间分布图（分段图）

上图含义为：对应横轴数据的每个时间段，其中转旅客的比率为纵轴坐标值。如，换乘时间在15~20分钟的中转旅客展全部中转旅客数量的比率为21.46%。

图5-13 中转旅客换乘紧张度分布（堆积图）

中转旅客换乘紧张度方面，绝大多数中转旅客的换乘紧张度处于0.1~0.3之间，并且全部都小于0.5。具体情况如图5-13、5-14所示。

图5-14 中转旅客换乘紧张度分布（分段图）

# 六、模型的总结与评价

本文建立了航班-登机口分配的模型。考虑了机型与登机口匹配、登机口停靠航班受限、航班最小时间间隔等诸多约束条件，以分配到固定登机口航班数量最多为首要目标，以换乘乘客的整体换乘紧张度或整体流程时间最少为次要目标，以被使用的固定登机口数量最少为再次目标，建立了航班-登机口分配的数学模型，并使用改进的遗传算法对其求解。通过编程计算，所得结果合理有效。

所建立的模型贴近情境需求，是科学合理的。本文所建立的登机口分配模型，约束条件考虑周全，目标函数明确。随着问题的逐渐深入，在之前模型基础上对约束条件和目标函数进行适当修改，即可解决新的问题，这对问题的建模和求解是十分有利的。

由于本文使用改进的遗传算法对模型进行求解，当输入的航班规模有较大增长时，问题的复杂度显著增大，求解时间明显增长。在求解计算时，算法的求解速度相对较快，基本能够满足实时性的需求。

本题所求得的结果是可行的，但未必是最优的。由于情境中限制条件较多，各种因素交错，模型在建立时已经尽可能多的考虑了约束条件。使用遗传算法求解时，初始种群数量、迭代次数有限，并且有可能陷入局部最优。因此结果是可行的、有参考价值的，但不一定是最优的。

本文所建立的模型对约束条件考虑周全，目标函数清晰合理，模型具有一定的可扩展性，求解过程合理且速度较快，所得结果具有可行性，总体来说，本文建立的模型较好的解决了目标问题，有一定的参考价值。

# 七、参考文献

[1] Abdelghani Bouras, Mageed A. Ghaleb, Umar S. Suryahatmaja, and Ahmed M. Salem, The Airport Gate Assignment Problem: A Survey, The Scientific World Journal, Volume 2014.

[2] Dong Zhang, Diego Klabjan. Optimization for gate re-assignment, Transportation Research Part B: Methodological, Volume 95, January 2017.

[3] Shuo Liu, Wenhua Chen, Jiyin Liu, Optimizing airport gate assignment with operational safety constraints, 20th International Conference on Automation and Computing, 2014.

[4] H. Ding, A. Lim, B. Rodrigues. Aircraft and Gate Scheduling Optimization at Airports, Proceedings of the 37th Hawaii International Conference on System Sciences – 2004.

[5] Deng-W. Study on an improved adaptive PSO algorithm for solving multiobjective gate assignment, Applied Soft Computing 59 (2017) 288–302.

[6] Andreas Drexl and Yury Nikulin, Multicriteria airport gate assignment and Pareto simulated annealing, IIE Transactions (2008) 40, 385–397.

[7] 卫东选. 基于改进遗传算法的机场停机位分配问题研究[D]. 中国民用航空学院,2006.

[8]文军.机场停机位分配问题的遗传算法[J].科学技术与工程,2010,10(01):135-139.

[9]卫东选,刘长有.机场停机位分配问题研究[J].交通运输工程与信息学报,2009,7(01):57-63+69.

[10]王志清,商红岩,宁宣熙.机场登机口优化调度算法及实证[J].南京航空航天大学学报,2007(06):819-823.

[11]文军,孙宏,徐杰,梁志杰.基于排序算法的机场停机位分配问题研究[J].系统工程,2004(07):102-106.

[12] 郑攀,胡思继,张晨.机门指派模型建立与启发式算法设计[J].系统工程学报,2011,26(01):127-131.

# 附录：

附录：程序和数据结构。

clc;

clear;

%% F:AGAP Problem

%% 1. 载入数据===========================================

Gate\_nums = 69; % 总登机口数量

Flight\_nums = 303; % 总航次数量

%% 航班表 303行 6列：

%{

1.到时间(1~4320) 2.走时间(1~4320)

3.到类型(I(0) / D(1)) 4.走类型(I(0) / D(1))

5.机型(W(1) / N(0)) 6.编号index(1~303)

%}

Flight\_table = xlsread('data\Flight\_table.xlsx');

%% 登机口表 6行 69列：

%{

1.到类型(I(0) / D(1) / I/D(2))

2.走类型(I(0) / D(1)/ I/D(2))

3.所在终端厅(T(0) / S(1))

4.支持机型(W(1) / N(0))

5.区域位置(North=0 Center=1 South=2 East=3)

6.编号index(1~69)

%}

Gate\_table = xlsread('data\Gate\_table.xlsx');

%% 机票乘客时间表 1649行 5列：

%{

1.乘客数量(1~2) 2.到达航次ID(1~303)

3.出发航次ID(1~303) 4.时间差

5.编号index(1~1649)

%}

Ticket\_table = xlsread('data\Ticket\_table.xlsx');

%% 登机口流程时间表 4行(到达) DT(10) DS(11) IT(00) IS(01) 4列(出发) DT(10) DS(11) IT(00) IS(01)

Progress\_time\_table = xlsread('data\Progress\_time\_table.xlsx');

%% 捷运时间 4\*4

Trans\_time\_table = xlsread('data\Trans\_time\_table.xlsx');

%% 行走时间

Work\_time\_table = xlsread('data\Work\_time\_table.xlsx');

%% 2. 算法参数设置 ================================

%% 最优个体

best\_Temp\_gate\_num= 1000; % 最优解临时口使用数量越少越好优先级最大

best\_Progress\_time = 10000000; %最优解 乘客 过程时间越少越好 次

best\_Change\_time = 100000000;% 最优解乘客中转时间越少越好 次

best\_time\_or\_tension = 1000000; % 紧张程度 次

best\_Gate\_num = 0; % 最优解 固定口 使用数量 越低越好 最低

% best\_Score = -10000; % 得分

%% 种群个体数量 为参数 的5~10倍 500个

I\_NP = 500;

%% 进化最大代数 100次

I\_itermax = 100;

%% 变异概率 Variation

V\_weight = 0.4;

%% 交叉概率 crossover 交叉后需要检查，使用变异来将不可行解变成可选解

C\_weight = 0.4;

%% 选择 Sele 留下的概率 0.8

S\_weight = 0.8;

%% 3. 产生初始种群 ========================

% 初始化 初始种群中的个体最优解

% 初始化

best\_Temp\_gate\_num\_init=1000;%最优解临时口使用数量越少好优先级最大

best\_Progress\_time\_init=10000000;% 最优解乘客过程时间 次

best\_Change\_time\_init=100000000; %最优解乘中转时间 次

best\_time\_or\_tension\_init =1000000; % 紧张程度========== 次

best\_Gate\_num\_init = 10000; % 最优解 固定口使用数量越多越好 最低

% best\_Score\_init = -10000; % 得分

NP = zeros(Flight\_nums, 2, I\_NP);

% 303\*2 初始化解的群体 第一列:分配的登机口 第二列：可变异否

max\_itr = 200; % 200次迭代 中 选一个好的 个体 作为种群中的个体

for n =1:I\_NP

str1 = sprintf('generate %d individual...',n);

disp(str1)

for i=1:max\_itr

% 输入：航次表 登机口表 机票表Ticket\_table 乘客时间表Progress\_time\_table

% 捷运时间表 Trans\_time\_table 行走时间表 Work\_time\_table

% 输出：一个解 临时口使用数量 过程时间/紧张程度 固定登机口使用数量

[Solution, Temp\_gate\_num\_init, time\_or\_tension\_init, Gate\_num\_init] = three\_init2( Flight\_table, Gate\_table,Ticket\_table, Progress\_time\_table, Trans\_time\_table, Work\_time\_table );

if((Temp\_gate\_num\_init < best\_Temp\_gate\_num\_init )||...

((Temp\_gate\_num\_init == best\_Temp\_gate\_num\_init ) && ( time\_or\_tension\_init < best\_time\_or\_tension\_init))||...

((Temp\_gate\_num\_init == best\_Temp\_gate\_num\_init ) && ( time\_or\_tension\_init == best\_time\_or\_tension\_init) && ( Gate\_num\_init < best\_Gate\_num\_init)))

% 临时口更少 或 临时口使用数量相同, 且紧张度小 或 临时口使用数量 和紧张度相同, 且固定口使用数量少

best\_Solution\_init = Solution;

best\_Temp\_gate\_num\_init = Temp\_gate\_num\_init; % 最少的 临时口 使用数量

best\_time\_or\_tension\_init = time\_or\_tension\_init; % 最少的 过程时间/紧张程度

best\_Gate\_num\_init = Gate\_num\_init; % 最少的 固定口 使用数量

end

end

NP(:,:,n) = best\_Solution\_init ;% 初始个体

end

%% 4. 开始迭代=============================================

% 5 -> 6 -> 7 -> 8 -> 5 -> 6 -> 7 -> 8 -> ...

for ll =1:I\_itermax

str1 = sprintf(' %d iter...',ll);

disp(str1);

%% 5. 交叉 后 使用变异来 将不可行解变成可行解===================

str1 = sprintf('begin crossover ...');

disp(str1);

NP = three\_crossover\_mutation(NP, Flight\_table, Gate\_table, C\_weight);

%% 6. 变异 =================================================

V\_num = floor(V\_weight \* I\_NP); % 变异个体数量

str1 = sprintf('variation individual num %d ', V\_num);

disp(str1);

c = randperm(numel(1:I\_NP));

V\_NP = c(1:V\_num);% 变异个体序列

for m =1:V\_num

[NP(:,:,V\_NP(m))]=three\_variation(NP(:,:,V\_NP(m)),Flight\_table, Gate\_table);

end

%% 7. 选择 ===============================================

str1 = sprintf('begin select.');

disp(str1)

% 产生 适应度值

NP\_OBJ = zeros(I\_NP,4);

% 存储每个 个体 的适应度值 ( 即 临时口使用数量 、 过程时间/紧张程度 和 固定口使用数量 ) 和 在种群中的id

for k = 1:I\_NP

[T, P, G] = three\_score(NP(:,:,k), Flight\_table, Gate\_table, Ticket\_table, Progress\_time\_table, Trans\_time\_table, Work\_time\_table);

NP\_OBJ(k,1:3) = [T, P, G];

NP\_OBJ(k,4) = k;% 在种群中的id

end

% 按 第一例 第二列 第三列 排序

NP\_OBJ = sortrow(NP\_OBJ);

% 选择最好的个体

if((NP\_OBJ(1,1) < best\_Temp\_gate\_num )||...

((NP\_OBJ(1,1) == best\_Temp\_gate\_num ) && ( NP\_OBJ(1,2) < best\_time\_or\_tension))||...

((NP\_OBJ(1,1) == best\_Temp\_gate\_num ) && ( NP\_OBJ(1,2) == best\_time\_or\_tension) && ( NP\_OBJ(1,3) < best\_Gate\_num)))

% 临时口更少 或 临时口使用数量相同,且紧张度小 或 临时口使用数量 和紧张度相同,且固定口使用数量少

best\_Solution = NP(:,:,NP\_OBJ(1,4)); % 解

best\_Temp\_gate\_num = NP\_OBJ(1,1); % 最少的 临时口 使用数量

best\_time\_or\_tension = NP\_OBJ(1,2); % 最少的 过程时间/紧张程度

best\_Gate\_num = NP\_OBJ(1,3); % 最少的 固定口 使用数量

end

%% 8. 产生新的个体 ===========================================

died\_num = ceil((1-S\_weight) \* I\_NP) ; % 需要淘汰的 个体数量

died\_id = NP\_OBJ(I\_NP - died\_num + 1 : I\_NP, 4);% 淘汰后面不好的个体

str1 = sprintf('begin generate %d new individuals', died\_num);

disp(str1)

% 产生新的

best\_Temp\_gate\_num\_new=1000;%最优解临时口使用数量越少好优先级最大

best\_Progress\_time\_new=10000000; % 最优乘客过程时间 次

best\_Change\_time\_new=100000000;% 最优解乘客中转时间 次

best\_time\_or\_tension\_new = 1000000;% 紧张程度

best\_Gate \_num\_new = 10000; %最优解固定口使用数量越少越好最低

% best\_Score\_new = -10000; % 得分

max\_itr = 200; % 200次迭代 中 选一个好的 个体 为为种群中的个体

for l = 1: died\_num

str1 = sprintf('generate %d th new individual',l);

disp(str1)

for i=1:max\_itr

% 输入：航次表 登机口表 机票表 Ticket\_table 乘客时间表 Progress\_time\_table捷运时间表 Trans\_time\_table 行走时间表 Work\_time\_table

% 输出：一个解 临时口使用数量 过程时间/紧张程度 固定登机口使用数量

[Solution, Temp\_gate\_num\_new, time\_or\_tension\_new, Gate\_num\_new]

=three\_init2(Flight\_table, Gate\_table, Ticket\_table, Progress\_time\_table, Trans\_time\_table, Work\_time\_table); %求解1次

if((Temp\_gate\_num\_new < best\_Temp\_gate\_num\_new )||...

((Temp\_gate\_num\_new==best\_Temp\_gate\_num\_new)&& ( time\_or\_tension\_new < best\_time\_or\_tension\_new))||...

((Temp\_gate\_num\_new==best\_Temp\_gate\_num\_new)&& ( time\_or\_tension\_new == best\_time\_or\_tension\_new) && ( Gate\_num\_new < best\_Gate\_num\_new)))

% 临时口更少 或 临时口使用数量相同,且紧张度小或 临时口使用数量 和紧张度相同,且固定口使用数量少

best\_Solution\_new = Solution;

best\_Temp\_gate\_num\_new = Temp\_gate\_num\_new; %最少的临时口使用数量

best\_time\_or\_tension\_new = time\_or\_tension\_new; % 最少的 过程时间

best\_Gate\_num\_new = Gate\_num\_new; % 最少的 固定口 使用数量

end

end

NP(:,:,died\_id(l)) = best\_Solution\_new ; % 新个体替换淘汰的个体

end

end