



## “华为杯”第十五届中国研究生 数学建模竞赛

学 校 江西财经大学

---

参赛队号 18104210009

---

1.罗铸堃

---

队员姓名 2.王俊杰

---

3.张皓涵

---

# “华为杯”第十五届中国研究生 数学建模竞赛

题 目                      机场新增卫星厅对中转旅客影响的评估方法

---

## 摘                      要：

本文针对真实的航班-登机口分配调度方案在不同情形下建立了相应的数学优化模型，并利用遗传算法通过 Python 编程来对各个模型进行求解。遗传算法中有一个评价指标称为适应度作为进化机制的参考依据，而三种情形对应的算法流程都是相同的，最大的区别，三种情况下适应度函数的设计不同。针对问题一，根据最后的结果得出模型按照 1:1 的比例成功安排全部航班，共 347 次航班；对于航站楼的登机口分配来说，我们总共使用了固定登机口 67 个，临时机位 1 个。窄体机航班总数为 298，使用临时机位 128 次，固定机位 100 次，宽体机航班总数为 49，使用临时机位 0 次，固定机位 49 次。从飞机机型来统计，总共使用适合宽体机的登机口 45 个，适合窄体机的登机口 22 个。从航站楼-卫星厅的使用情况来统计，总共使用航站楼 T 登机口 28 个，卫星厅 S 登机口 39 个。针对问题二、问题三，我们采用同样的建模方法，根据适应度公式，在遗传算法中选出适应度最大的个体，进而得出问题的最优解。

关键词：航班登机口分配；Python；遗传算法；适应度



# 1 问题重述

## 1.1 问题背景

由于旅行行业的快速发展，某航空公司在某机场的现有航站楼T的旅客流量已达饱和状态，为了应对未来的发展，现增设卫星厅S。但引入卫星厅后，虽然可以缓解原有航站楼登机口不足的压力，但对中转旅客的航班衔接显然具有一定的负面影响。本文试图通过针对不同的情形建立数学模型，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。

## 1.2 问题的提出

在机场的布局中航站楼T有28个登机口，卫星厅S有41个登机口，其中航站楼T具有完整的国际机场航站楼功能，包括出发、到达、出入境和候机，而卫星厅S可以候机，没有出入境功能。T和S之间有捷运线用来快速往来运送国内、国际旅客。假定旅客无需等待，随时可以发车，T与S之间单程一次的往返需要8分钟。登机口属于固定机位，每个登机口具有国内/国际、到达/出发、宽体机/窄体机等属性，在分配过程中飞机转场计划里航班需要与分配到的登机口属性相同，除此之外，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，且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。在整个机场的布局以及登机口分配规则不改变的情况下，真实的航班-登机口分配调度方案会受各种制约因素影响，因此本文针对以下三种情形对航班-登机口分配问题建立数学优化模型。

1. 在不考虑中转旅客换乘且仅考虑航班-登机口分配的情况下，建立数学优化模型，尽可能多地分配航班到合适的登机口，并且在此基础上最小化被使用登机口的数量。
2. 考虑中转旅客最短流程时间。在情形一的基础上加入旅客换乘因素，要求最小化中转旅客的总体最短流程时间，并且在此基础上最小化被使用登机口的数量。
3. 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，数学模型最终需要考虑换乘旅客总体紧张度的最小化，并且在此基础上最小化被使用登机口的数量。可以在情形二的基础上细化，引入旅客换乘连接变量，并把中转旅客的换乘紧张度作为目标函数的首要因素。

# 2 问题分析

## 2.1 航班-登机口分配模型的约束条件

航班-登机口分配问题根据相应的场景确定分配问题的约束条件和优化目标，不同场景在满足的约束条件上不完全相同，有的模型考虑的约束条件多一些，有的相对少一下。但是根据登机口的分配规则通常的约束条件有：

1. 为每个航班分配登机口时，同一个时间段最多只能有一架飞机。

2. 对于每个航班来说，能且仅能分配到一个登机口，若是分配不到固定登机口，则可以分配到简易临时机位。
3. 每架飞机转场的到达和出发两个航班必须在同一登机口机型。
4. 分配在同一登机口的前后两个航班之间要保证空档间隔时间必须大于等于45分钟。
5. 每个登机口的国内/国际、到达/出发、宽体机/窄体机等动能属性要与分配到的航班飞机配置相匹配。

## 2.2 航班-登机口分配模型的优化目标

航班-登机口分配问题根据场景又有如下几种优化目标：

1. 最小化被使用登机口的数量。
2. 最小化中转旅客的总体中转流程时间。
3. 最小化中转旅客的总体紧张度。

## 2.3 问题1

问题一是在只考虑航班-登机口而不考虑中转旅客的条件下，由于航班以及登机口的数量是确定的，以最小化使用登机口数量为优化目标建立航班-登机口优化模型。利用Python编程使用遗传算法对模型进行求解。即可求得最小使用的登机口数量，以及成功分配到登机口的航班数量和比例，使用分别适用宽体机/窄体机机型的登机口数量。

## 2.4 问题2

在问题一的基础上，考虑中转旅客根据航班性质在航站楼-卫星厅之间换乘的时间，以最小化中转旅客的总体最短流程时间以及最小化使用登机口数量为优化目标建立航班-登机口优化模型。同样，利用Python编程使用遗传算法对模型进行求解。即可求得中转旅客的总体最短流程时间，以及最小化使用的登机口数量，同时也可得到换成失败旅客的数量和比率。

## 2.5 问题3

在问题二的基础上，考虑中转旅客问题根据航班性质在航站楼-卫星厅之间换乘坐捷运线以及登机口区域之间的步行时间，结合换乘旅客的航班连接时间以及换乘时间得到换乘旅客的换乘紧张度，并以最小化中转旅客的换乘紧张度以及最小化使用登机口数量为优化目标建立航班-登机口优化模型。同样，利用Python编程使用遗传算法对模型进行求解。即可求得各类型中转旅客的换乘紧张度，以及最小化使用的登机口数量，同时也可得到各个紧张度内的中转旅客比例。

### 3 模型假设

为了使建立的模型能够更好的适合航班-登机口分配问题，明确登机口分配所满足的条件，本文给出如下假设：

1. 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。
2. 本文研究的是20日出发的航班，对于19日到达且并未起飞的航班记为停机状态，并根据条件为其提供初始的登机口分配。
3. 对于需要乘坐捷运线的旅客，无需等待捷运线的时间。
4. 对于旅客的换乘时间，只考虑航站楼与卫星厅之间的流程时间，各个登机口区域之间的步行时间以及捷运时间，不考虑其他逗留时间。

### 4 符号说明

符号表示	解释说明
$M$	20日出发总的航班数
$N$	航站楼加上卫星厅共有总的登机口数
$x$	剩余适用窄体机的登机口数
$y$	剩余适用宽体机的登机口数
$T'$	航站楼T登机口的使用数目
$S'$	卫星厅S登机口的使用数目
$s_{i,j}$	航班 $i$ 是否被分配到登机口 $j$ (1:是0:否)
$g_{i,j,k}$	航班 $i$ 和航班 $j$ 是同一个登机口 $k$ 的先后两个航班，且航班 $i$ 在前(1:是0:否)
$f_i$	航班 $i$ 对应的飞机型号(1:窄体机2:宽体机)
$d_i$	登机口 $i$ 可以停留的飞机型号(1:窄体机2:宽体机)
$k_i$	登机口 $i$ 被分配使用的次数
$u_a$	登机口的平均使用率
$v_i$	航班 $i$ 对应的出发/到达类型(1:国内2:国际)
$c_i$	登机口 $i$ 可以的航班出发/到达类型(1:国内2:国际)
$A_{j,k}$	航班 $j$ 到达登机口 $k$ 的到达时间
$L_{i,k}$	航班 $i$ 离开登机口 $k$ 的出发时间
$\Delta$	同一登机口两飞机之间最小空档间隔，这里规定为45分钟
$p_i$	编号 $i$ 对应的换乘乘客数
$l_i$	编号 $i$ 对应的换乘乘客中转流程时间(一共16种中转情况)
$r_i$	编号 $i$ 乘客对应的航班连接时间
$m_i$	编号 $i$ 乘客对应的捷运时间
$w_i$	编号 $i$ 乘客对应的行走时间
$h_i$	编号 $i$ 乘客对应的换乘时间
$\epsilon_i$	编号 $i$ 乘客对应的换乘紧张度

## 5 模型建立

### 5.1 问题1

当在只考虑航班-登机口分配且不考虑中转旅客的基础上要最小化登机口的数量，即要使得剩余未使用的登机口数量最大，则可用以下数学模型表示：

$$\max f = x + y \quad (1.1)$$

$$\sum_{j=1}^N y_{i,j} = 1 \quad (1.2)$$

$$f_i = d_j, \forall (i, j) \in \{(i, j) | y_{i,j} = 1, i = 1, 2 \dots M, j = 1, 2 \dots N\} \quad (1.3)$$

$$v_i = c_j, \forall (i, j) \in \{(i, j) | y_{i,j} = 1, i = 1, 2 \dots M, j = 1, 2 \dots N\} \quad (1.4)$$

$$A_{j,k} - L_{i,k} \geq \Delta, \forall g_{i,j,k} = 1, \&\& i, j = 1, 2 \dots M \quad (1.5)$$

在上面各式中，(1.1)是目标函数，即最小化占用适用宽体机和窄体机的登机口数目总和，在航站楼和卫星厅的登机口总数固定的情况下，可将目标转化为剩余未使用登机口数量最大，(1.2)保证每个航班能且只能分配到一个登机口，(1.3)保证航班的飞机类型与登机口所能容纳的飞机类型相匹配，(1.4)保证飞机的到达或是起飞类型与登机口所能容纳的类型相匹配，(1.5)保证分配到同一登机口的两飞机之间的空档间隔时间必须大于等于45分钟。

### 5.2 问题2

在问题一的基础上，加上各换乘旅客根据航班性质在航站楼-卫星厅之间的最短换乘时间，要使得中转旅客的总体流程时间最短，并在此基础上最小化被使用登机口数量，则可用以下数学模型表示：

$$\min f_1 = \sum_{i=1}^M l_i \cdot p_i \quad (2.1)$$

$$\min f_2 = N - (x + y) \quad (2.2)$$

$$\sum_{j=1}^N y_{i,j} = 1 \quad (2.3)$$

$$f_i = d_j, \forall (i, j) \in \{(i, j) | y_{i,j} = 1, i = 1, 2 \dots M, j = 1, 2 \dots N\} \quad (2.4)$$

$$v_i = c_j, \forall (i, j) \in \{(i, j) | y_{i,j} = 1, i = 1, 2 \dots M, j = 1, 2 \dots N\} \quad (2.5)$$

$$A_{j,k} - L_{i,k} \geq \Delta, \forall g_{i,j,k} = 1, \&\& i, j = 1, 2 \dots M \quad (2.6)$$

在上面各式中，(2.1)和(2.2)是目标函数，即最小化中转旅客的总体最短流程时间，以及最小化占用适用宽体机和窄体机的登机口数目总和，(2.3) 保证每个航班能且只能分配到一个登机口，(2.4) 保证航班的飞机类型与登机口所能容纳的飞机类型相匹配，(2.5)保证飞机的到达或是起飞类型与登机口所能容纳的类型相匹配，(2.6)保证分配到同一登机口的两飞机之间的空档间隔时间必须大于等于45 分钟。

### 5.3 问题3

当问题二的基础上，对于各换乘旅客的换乘时间，除了根据航班性质在航站楼-卫星厅之间的最短换乘时间之外还要加上航站楼-卫星厅之间的捷运时间以及各个登机口区域之间的行走时间，从而获得各旅客的换乘紧张度，要使得中转旅客的总体换乘紧张度最小，并在此基础上最小化被使用登机口数量，则可用以下数学模型表示：

$$\min f_1 = \sum_{i=1}^M \epsilon_i \cdot p_i \quad (3.1)$$

$$\min f_2 = N - (x + y) \quad (3.2)$$

$$\sum_{j=1}^N y_{i,j} = 1 \quad (3.3)$$

$$f_i = d_j, \forall (i, j) \in \{(i, j) | y_{i,j} = 1, i = 1, 2 \dots M, j = 1, 2 \dots N\} \quad (3.4)$$

$$v_i = c_j, \forall (i, j) \in \{(i, j) | y_{i,j} = 1, i = 1, 2 \dots M, j = 1, 2 \dots N\} \quad (3.5)$$

$$A_{j,k} - L_{i,k} \geq \Delta, \forall g_{i,j,k} = 1, \&\& i, j = 1, 2 \dots M \quad (3.6)$$

$$h_i = l_i + m_i + w_i \quad i = 1, 2 \dots H \quad (3.7)$$

$$\epsilon_i = \frac{r_i}{h_i} \quad i = 1, 2 \dots H \quad (3.8)$$

在上面各式中，(3.1)和(3.2)是目标函数，即最小化中转旅客的总体换乘紧张度，以及最小化占用适用宽体机和窄体机的登机口数目总和，(3.3)保证每个航班能且只能分配到一个登机口，(3.4)保证航班的飞机类型与登机口所能容纳的飞机类型相匹配，(3.5)保证飞机的到达或是起飞类型与登机口所能容纳的类型相匹配，(3.6)保证分配到同一登机口的两飞机之间的空档间隔时间必须大于等于45分钟，(3.7)求得各类型旅客的换乘时间，(3.8)求得各类型旅客的换乘紧张度。

## 6 模型求解

### 6.1 遗传算法

遗传算法是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法。遗传算法是从一组初始解，成为“种群”，开始搜索，种群的每一个解称为“个体或染色体”，个体是一串编码，个体在后续迭代中不断进化，在每一代中用“适应度”来衡量个体的好坏，生成的下一代个体称为后代。后代是由前一代个体通过交叉和变异运算形成的。从父代种群和后代种群中，根据适应度值的大小选择比较优秀的个体作为新的种群，淘汰差的个体，从而保持群体规模基本不变。这样，经过若干次迭代之后，算法收敛于最优个体，那么该个体很有可能就是航班-登机口分配问题的最优或次优解。遗传算法的基本流程可分为以下几步：

1. 根据所要解决得实际问题确定问题得参数集，并对参数进行遗传编码，随机生成初始的个体；



2. 计算群体中的每个个体的适应值，判断是否符合优化准则；若符合，输出最佳个体及其代表的最优解，并结束计算，否则转向第3步；
3. 依据适应度选择再生个体，适应度高的个体被选中的概率高，适应度低的个体可能被淘汰；
4. 按照一定的交叉概率和交叉方法，生成新的个体；
5. 按照一定的交叉概率和变异方法，生成新的个体；
6. 由交叉和变异产生新一代的种群，返回第2步；
7. 终止运算后把在任一代中出现的最好的个体指定为遗传算法的执行结果，这个结果可以表示问题的一个解（或近似解），以达到优化分配的结果。

标准遗传算法的流程如图1所示，在遗传算法中，适应度是个体评价的一种指标，它与个体的目标值存在一一对应关系；复制操作通常采用比例复制，即复制概率正比于个体的适应度值，意味着适应值高的个体在下一代中复制自身的概率大，从而可提高种群的平均适应度值；交叉操作通过交换两父代个体的部分信息构成后代个体，使得后代继承父代的有效模式，从而有助于产生优良个体；变异操作通过随意改变个体中某些基因而产生新个体，有助于种群的多样性，避免早熟收敛。

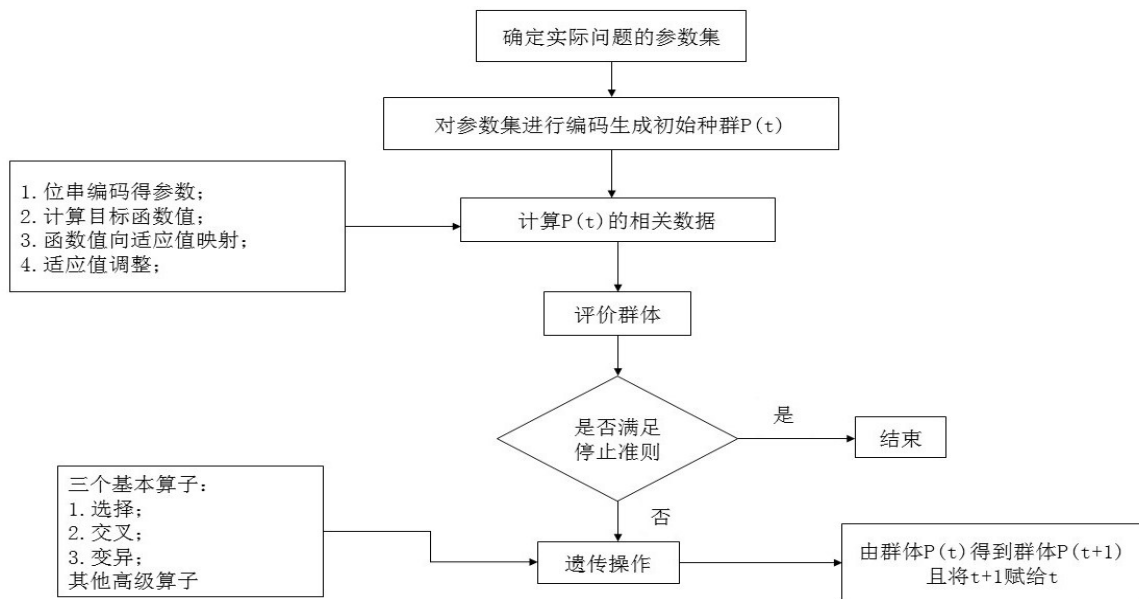


图 1: 标准遗传算法的基本流程

## 6.2 基于航班-登机口模型的遗传算法

针对本文中三种情况的航班-登机口分配优化模型，都可以利用遗传算法该模型进行求解。整个算法包括以下几步：（具体针对三种情形的实现代码详见附录[代码实现]）

1. 根据要解决机位分配的实际问题特性和约束条件进行遗传表达，即进行染色体编码设计；

2. 产生潜在初始解群体的方法设计；
3. 对个体适应度评价函数设计；
4. 进行遗传操作的设计；
5. 设定遗传算法的参数和终止条件。

### 6.3 染色体编码设计

在常规的遗传算法中，编码是把问题的搜索空间中的每个可能的点表示为确定长度的字符串。编码的关键因素有两个，一个是如何将问题的显性空间的特性融入基因空间中，即如何将问题的解以基因的方式表示；另一个是当算法在运行时基因的变化方式。因此，必须要根据问题的实际需要来合适地定义染色体来作为模型运行时最基本的个体。在遗传算法中常用的编码方式有四种：

1. 二进制编码；
2. 整数编码；
3. 各类编码；
4. 浮点数编码。

本文采用整数的编码方式，利用 $M$ 维整数字符串 $(b_1, b_2, \dots, b_M)$ 表示一个分配方案， $M$ 维航班数量。其中每一个编码 $b_i$ 代表将第 $i$ 个航班分配到其可分配的登机口集合 $D_i$ 中的第 $b_i$ 个登机口。设第 $i$ 个航班可停靠的登机口集合为 $D_i = \{1, 2, 3, 4, 5\}$ ，若 $b_i = 2$ 表示将该航班分配到2号机位。本文中的机场登机口分布在航站楼T以及卫星厅S，一共有69个登机口，按照航站楼T在前，卫星厅S在后的顺序按顺序给69个登机口进行编号。将 $b_i$ 设计为第 $i$ 个航班可分配登机口集合的索引，而不直接定义为登机口索引。是出于考虑以下两个方面：

1. 每个航班可停靠登机口的编号不一定是连续（如某航班的可停登机口编号为2,4,8），若直接定义为登机口编号则将会出现变异等遗传操作产生无效解的情况
2. 采用这样的二次变换可以方便将模型调整为二进制等其他编码方法。

### 6.4 初始种群的生成

由于航班之间的相互影响和制约，每个航班的可能停机登机口分配状况由当前的登机口空闲状态来决定，而当前的登机口空闲状态，受航班的登机口分配状况的影响。在本文中，20日的航班分配的登机口受19号过夜飞机的影响，体现在编码上就是一个基因的取值，受其之前的基因状况的影响，对其后的基因亦起决定作用。这些影响是由约束条件所决定的，只有满足约束条件的基因取值才是可行解。如果不对染色体中的基因和每代的染色体的产生进行控制，而采取随机取值的话，产生可行解的概率很小，从而大大降低寻优的效率和寻优结果。为了提高遗传算法的收敛速度，从初始群体的建立到每一代子群体的形成，都采用约束控制手段，使其产生的初始解均为可行解。对于初始种群的建立，首先按照航班到达时间顺序初始化 $M$ 个航班，记为集合 $(J_1, J_2, \dots, J_M)$ ；登机口是按照机场的安排方式和开启时间进行初始化，记为集合 $(G_1, G_2, \dots, G_N)$ 。每个航班的数据包括航班的编号、到达时间、作业时间和航班类型，登机口的数据包括登机口的编号、登机口开始空闲时间、登机口类型。初始可行解的产生流程图2所示：

1. 读取航班J和机位G的初始信息；
2. 产生所有航班可停靠机位的集合( $D_1, D_2, \dots, D_M$ )， $D_i$ 表示航班 $i$ 停靠登机口的集合。航班 $i$ 可停靠登机口 $j$ 需要满足的约束包括：航班 $i$ 实际到达时间要晚于登机口 $j$ 开始空闲时间，航班 $i$ 离开时间要早于登机口 $j$ 的结束使用时间，且相邻两航班使用同一机位的时间间隔不小于 $\Delta$ (本文定为45分钟)；航班 $i$ 的机型要在登机口 $j$ 可容纳范围内（即需要满足航班-登机口分配模型的约束条件）；
3. 为一个航班生成一个随机整数 $b_i \in [1, \text{count}(D_i)]$ ，可确定航班 $i$ 在可停登机口集合 $D_i$ 中的登机口 $j$ ，更新登机口 $j$ 的开启时间；若大于边界时间，，则在其他航班可停靠的机位集合中将此登机口 $j$ 删去，更新其他航班可停靠登机口的集合；
4. 判断( $D_{i+1}, D_{i+2}, \dots, D_M$ )， $D_i$ 中是否有空集，如果有则转第2步；如果没有空集，则转第5步；
5. 判断 $i = M$ 是否成立，如果成立则转第6步；如果不成立，则令 $i = i + 1$ 后转第3步；
6. 输出一组可行解。由于每个航班都是随机分配到当前空闲机位，所以每次循环都可以得到一组不同的分配方案。初始种群都是由 $M$ 个染色体构成，循环 $M$ 次即得到种群数量为 $M$ 的一个初始群体。

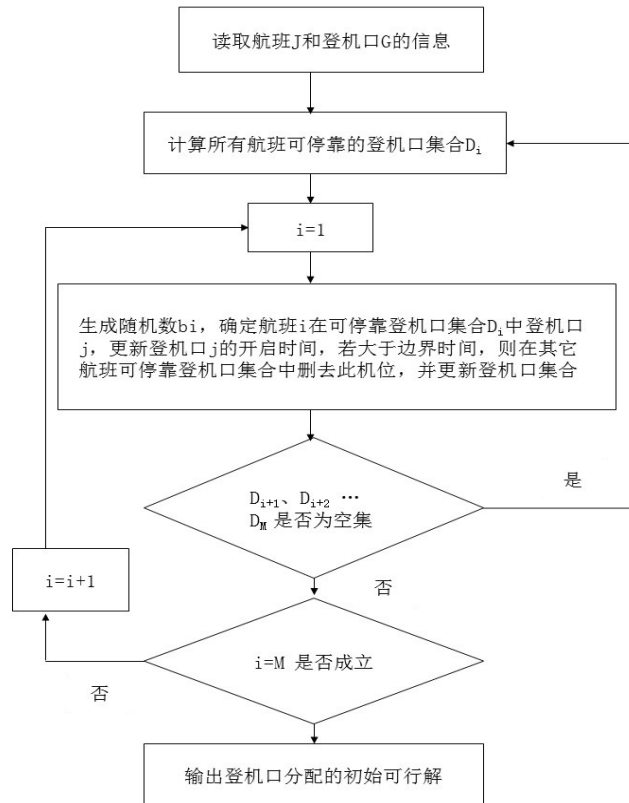


图 2: 初始可行解搜索流程图

本文将每个航班作为一个个体，然后根据约束条件按照遗传算法的要求初始化了一个种群，并将其保存在附件[1]中，其中部分内容如下图所示：

```

1 第0个体: ['T24', 'S30', 'S30', 'T24', 'S30', 'S29', 'S29', 'T24', 'T25', 'T24', 'S30', 'T70', 'T24', 'T23', 'T24', 'T24', 'T70', 'S29', 'T23', 'T23', 'T70', 'T24', 'T70',
2 第0次航班分配情况: {'T24': 45, 'S30': 23, 'S29': 22, 'T25': 5, 'T70': 56, 'T23': 46, 'T22': 4, 'T26': 2, 'T2': 1, 'T1': 2, 'S11': 3, 'T7': 9, 'T6': 7, 'S13': 1, 'S34': 1, 'T
3 第1个体: ['S30', 'T70', 'T70', 'S30', 'T70', 'S29', 'T70', 'T25', 'T23', 'T24', 'T24', 'T70', 'T5', 'S29', 'S30', 'T70', 'S29', 'S29', 'S30', 'T23', 'T25', 'S30', 'T70',
4 第1次航班分配情况: {'S30': 22, 'T70': 52, 'S29': 28, 'T25': 12, 'T23': 47, 'T24': 33, 'T5': 8, 'T6': 7, 'T9': 1, 'S37': 1, 'T28': 1, 'S12': 4, 'S13': 4, 'T1': 5, 'T22': 2, 'T
5 第2个体: ['S30', 'S30', 'S29', 'S30', 'S29', 'T70', 'T23', 'T23', 'T24', 'S29', 'T24', 'T25', 'T70', 'S30', 'S29', 'T70', 'S30', 'S30', 'T70', 'T70', 'T23', 'S29', 'T24',
6 第2次航班分配情况: {'S30': 26, 'S29': 26, 'T70': 49, 'T23': 39, 'T24': 38, 'T25': 9, 'T7': 6, 'T9': 2, 'T28': 2, 'T2': 2, 'S13': 4, 'T22': 4, 'T1': 4, 'S12': 4, 'T6': 5, 'T2
7 第3个体: ['T24', 'T24', 'T70', 'T23', 'S29', 'S30', 'T70', 'T25', 'T70', 'T23', 'T23', 'T70', 'T24', 'S29', 'T23', 'T70', 'S30', 'S29', 'T70', 'T25', 'T24', 'S30',
8 第3次航班分配情况: {'T24': 43, 'T70': 49, 'T23': 47, 'S29': 22, 'S30': 19, 'T25': 11, 'T22': 6, 'T6': 3, 'T3': 4, 'T26': 3, 'S11': 5, 'T1': 4, 'S12': 7, 'T21': 1, 'T7': 7, 'T
9 第4个体: ['T23', 'T24', 'S29', 'T24', 'T24', 'T24', 'T70', 'T24', 'T70', 'T24', 'T70', 'T25', 'T5', 'T23', 'T70', 'T23', 'T23', 'T24', 'S29', 'T5', 'T25', 'S29', 'S30', 'T
10 第4次航班分配情况: {'T23': 41, 'T24': 43, 'S29': 19, 'T70': 47, 'T25': 14, 'T5': 14, 'S30': 24, 'S11': 1, 'T9': 1, 'S39': 5, 'T22': 8, 'S12': 2, 'T6': 6, 'S13': 2, 'T7': 1,
11 第5个体: ['S29', 'T23', 'T24', 'S30', 'T24', 'T23', 'T70', 'T25', 'T70', 'T23', 'T23', 'T23', 'T23', 'S30', 'T23', 'T24', 'S29', 'T24', 'T70', 'T5', 'T70', 'T24', 'S29',
12 第5次航班分配情况: {'S29': 14, 'T23': 50, 'T24': 39, 'S30': 19, 'T70': 57, 'T25': 10, 'T5': 8, 'T1': 8, 'T22': 7, 'S39': 2, 'S11': 4, 'T7': 6, 'T6': 3, 'S12': 2, 'S33': 2, 'T
13 第6个体: ['S29', 'T70', 'T70', 'S30', 'T24', 'T70', 'T23', 'T25', 'T70', 'T23', 'T23', 'T25', 'T5', 'T24', 'T23', 'T25', 'T24', 'T23', 'T24', 'T24', 'T24', 'T70', 'T23',
14 第6次航班分配情况: {'S29': 25, 'T70': 41, 'S30': 16, 'T24': 44, 'T23': 45, 'T25': 16, 'T5': 12, 'T6': 7, 'T22': 8, 'S12': 6, 'S13': 2, 'T7': 8, 'T9': 1, 'S11': 1, 'S38': 1,
15 第7个体: ['T70', 'S29', 'T70', 'T24', 'S29', 'S30', 'S29', 'T70', 'T25', 'T23', 'T70', 'T25', 'T5', 'T23', 'T23', 'T23', 'S30', 'T70', 'S29', 'T23', 'T70', 'S29', 'T23',
16 第7次航班分配情况: {'T70': 57, 'S29': 26, 'T24': 37, 'S30': 22, 'T25': 11, 'T23': 36, 'T5': 9, 'T1': 4, 'T22': 8, 'S40': 2, 'S33': 2, 'S11': 2, 'T20': 1, 'T6': 5, 'S32': 1,
17 第8个体: ['S30', 'S30', 'T24', 'S30', 'T23', 'T70', 'T24', 'T25', 'T24', 'T23', 'T25', 'T23', 'S30', 'T24', 'T23', 'T70', 'T24', 'S30', 'T24', 'T70', 'T23', 'T24',
18 第8次航班分配情况: {'S30': 21, 'T24': 40, 'T23': 52, 'T70': 33, 'T25': 11, 'S11': 7, 'T9': 1, 'S40': 2, 'S39': 2, 'S29': 23, 'S12': 5, 'S13': 3, 'T6': 7, 'T1': 3, 'T7': 2, 'T
19 第9个体: ['T23', 'S30', 'T24', 'T23', 'T24', 'S30', 'S30', 'T24', 'T70', 'T24', 'T70', 'T24', 'T70', 'T23', 'T70', 'T25', 'T23', 'S29', 'T24', 'T24', 'T70', 'S29', 'T70',
20 第9次航班分配情况: {'T23': 39, 'S30': 20, 'T24': 43, 'T70': 56, 'T25': 8, 'S29': 26, 'T22': 7, 'T7': 8, 'S36': 1, 'S11': 3, 'S13': 3, 'T6': 5, 'T2': 3, 'T1': 1, 'S12': 4, 'S
21 第10个体: ['T23', 'S30', 'T24', 'T23', 'T23', 'T70', 'T24', 'T24', 'T25', 'T24', 'S30', 'T23', 'T70', 'S30', 'T70', 'T24', 'S30', 'T24', 'T23', 'T5', 'T70', 'S29', 'S30',
22 第10次航班分配情况: {'T23': 34, 'S30': 25, 'T24': 43, 'T70': 44, 'T25': 12, 'T5': 9, 'S29': 26, 'T1': 2, 'T9': 1, 'S35': 3, 'S32': 3, 'S13': 3, 'T6': 11, 'T20': 1, 'T7': 7,

```

图 3: 初始化航班种群

## 6.5 适应度评价函数的设计

为了评价群体中个体的优劣，遗传算法必须导入一个评价指标作为进化机制的参考依据，为此评价指标为适应度。依据投入变量的不同，将适应度估算方式定义为适应度函数，此函数是评价个体是否能适应模型目标函数的指标，也是个体是否能继续进化的标准，通常与模型的目标函数相关。为保证算法收敛速度，本文定义如下规则来计算适应度：

1. 在计算过程中，如果基因编码对应的分配方案与约束条件冲突，则适应度为0；
2. 分配优化的目的是为了实现目标函数 $f$ 的最优，直接采用目标函数作为适应度函数，不同的分配结果取值相差很大，收敛速度过慢，影响算法效率。因此引入模拟退火的思想，假设 $N$ 个登机口分别被使用次数记为 $(k_1, k_2, \dots, k_N)$ ，临时停靠机位被使用的次数记为 $o$ ，重新改进并设计适应度函数为：

$$Fitness = \max o + \sum_{i=1}^N ((k_i - \frac{w_i^2}{k_i}) \cdot p_i)$$

另外，适应度函数与目标函数的单调性相反，当目标函数取到最大值时，适应度函数取值最小，也即适应度值越高越能满足目标函数的要求。同时，在这里就避免了将目标函数最小化再转化最大化来求解。

## 6.6 遗传操作设计

### 6.6.1 精英保留策略

遗传算法根本上是一种随机搜索算法，交叉、变异等遗传算子的操作可能会将优良的染色体破坏，而只保留优良染色体的部分基因，最后所得到的优化解很可能并不是所能搜索到的最优解，而是整个搜索过程中的次优解。并且在实际运营中的停机位分配仅得到一个最优方案时不可行的，必须有一定数量的备选方案。鉴于此，采取精英保留策略，在每次遗传算子开始之前，我们从种群染色体中选择一定比例的优良染色体并记录。而又考虑到可能出现的“早熟问题”，为保持群体的多样性，与一般精英保留策略不同，遗传操作产生新的众群后，并不讲其与新的种群一起进行下次遗传运算。每代如此操作，在遗传运算结束后，便得到一组最优的分配方案。在实际操作中，机位分配问题要求实时性和即时性，采用精英保留策略，即使由于时间不允许最终得到最优解，所得到的亦是所能搜索到的分配方案中的一组较优方案。

## 6.6.2 遗传算子的设计

### 1.选择

选择算子是按照适应性强的染色体保留，而淘汰适应性较差的染色体的原则进行，即优胜劣汰的原则。它是遗传算法优化搜索的动力所在，选择算子的使用才能使优化过程不断向前。现在已经有多种有效得算子，如比例选择，轮盘赌选择，随机联赛选择等。而比例选择方法是最常用得选择算子，是一种回放式随机采样得方法。其思想是：各个个体被选中得概率与其适应度大小成正比。由于比例选择直接反映了适应度函数对选择概率的影响，适应度越高得个体被选中得概率也越大，所以本文采用比例选择方法。设群体大小为 $Q$ ，个体 $i$ 的适应度为 $F_i$ ，则个体 $i$ 被选中的概率为 $P_i$ ：

$$P_i = \frac{F_i}{\sum_{i=1}^Q F_i} \quad (i = 1, 2, 3, \dots, Q)$$

### 2.变异

变异是指个体染色体编码串中的某些基因座上的基因值用该基因座的其他等位基因来替换，从而形成一个新的个体。交叉运算决定了遗传算法的全局搜索能力，而变异运算则决定了遗传算法的局部搜索能力，提供了一种跳出目前搜索空间避免落入局部最优的进化方式。变异操作主要有以下两个目的：

1. 改变遗传算法局部搜索能力；
2. 维持群体的多样性，防止出现早熟现象。在进行变异操作前一般先设定一个变异频率 $P_m$ ，一般会设为一个较小值，以免过分破坏由选择、交叉所产生的优良染色体。

## 6.6.3 参数设定与算法的终止条件

遗传算法的参数包括群体规模、选择、交叉、变异频率以及终止条件。控制参数的选取对于算法的收敛有着很大的影响，最后得到的结果也会有较大的差异，参数的设定有必要对实际数据进行多次实验，选择参数的最优设置。首先，关于群体规模大小的设定，群体规模太小是，可能会过早的收敛于局部最优解；群体规模选取越大，收敛到全局最优解的概率也越大；但如果群体规模过大会造成计算量过大，收敛的速度也越慢，不能被接受。

## 6.7 问题1

针对问题一，假设 $N$ 个登机口分别被使用次数记为 $(k_1, k_2, \dots, k_N)$ ，临时停靠机位被使用的次数记为 $o$ ，设计的适应度函数可为：

$$Fitness_1 = \max \sum_{i=1}^N (k_i - \frac{o^2}{k_i})$$

将该适应度函数带入算法模型中，根据遗传算法对优化目标进行搜索，最后得出以下结果：

1. 航班总数为：347，成功安排全部航班数：347，比例为1:1，其中窄体机航班数为：298，宽体机航班数为：49。

2. 航站楼T和卫星厅S登机口总数为：69，总共使用固定登机口数为：67，其中使用适合宽体机的登机口数为：45，使用适合窄体机的登机口数为：22，使用临时机数为：1。
3. 窄体机的航班总数为：298，其中临时机位使用次数为：128，固定机位使用次数为：100，但是由于各航班使用时间可以错开只使用一个临时机位。
4. 宽体机的航班总数为：49，其中固定机位使用次数为：49，临时机位使用次数为：0。
5. 航站楼T登机口总数为：28，使用个数为：28，卫星厅S登机口总数为：41，使用个数为39。

针对于问题1，具体的分配情况如下，详细内容见附件[问题1登机口分配情况]，其中起飞登机口空白处意味着和起飞登机口到达登机口相同，无需改变， $t_{70}$  表示临时停机位，因此到达登机口使用临时机位的，起飞时需要为其分配满足条件的登机口。

	A	B	C				
1	转场号	到达登机口	起飞登机口				
2	PK062	T10		131	PK479	t70	T11
3	PK072	T11		132	PK480	t70	T13
4	PK102	T12		133	PK482	t70	T12
5	PK117	T13		134	PK484	t70	T12
6	PK136	T14		135	PK486	S28	
7	PK145	T15		136	PK488	t70	T12
8	PK147	T16		137	PK489	S22	
9	PK150	T17		138	PK490	S24	
10	PK155	T18		139	PK491	S25	
11	PK168	T19		140	PK493	t70	T11
12	PK170	S1		141	PK494	t70	T12
13	PK171	S2		142	PK496	S27	
14	PK173	S3		143	PK497	t70	T12
15	PK174	S4		144	PK498	t70	T11
16	PK175	S5		145	PK500	S26	
17	PK179	S6		146	PK501	t70	T11
18	PK180	S7		147	PK062	T1	
19	PK181	S8		148	PK072	S11	
20	PK182	S9		149	PK102	S12	
21	PK184	S10		150	PK117	S13	
22	PK187	S14		151	PK136	t70	S13
23	PK188	S15		152	PK145	t70	S13
24	PK191	S16		153	PK147	t70	S12
25	PK192	S17		154	PK150	t70	S12
26	PK193	S18		155	PK155	t70	S12
27	PK194	S19		156	PK168	t70	S12
28	PK195	S20		157	PK170	t70	S12
29	PK196	S21		158	PK171	t70	S12
30	PK197	S22		159	PK173	t70	S12
				160	PK174	t70	S12

(a) 问题1登机口分配(1)

(b) 问题1登机口分配(2)

图 4: 问题1登机口分配

根据以上分配情况，按宽体机/窄体机类型分别统计出各个登机口的使用次数，如下两张图所示：

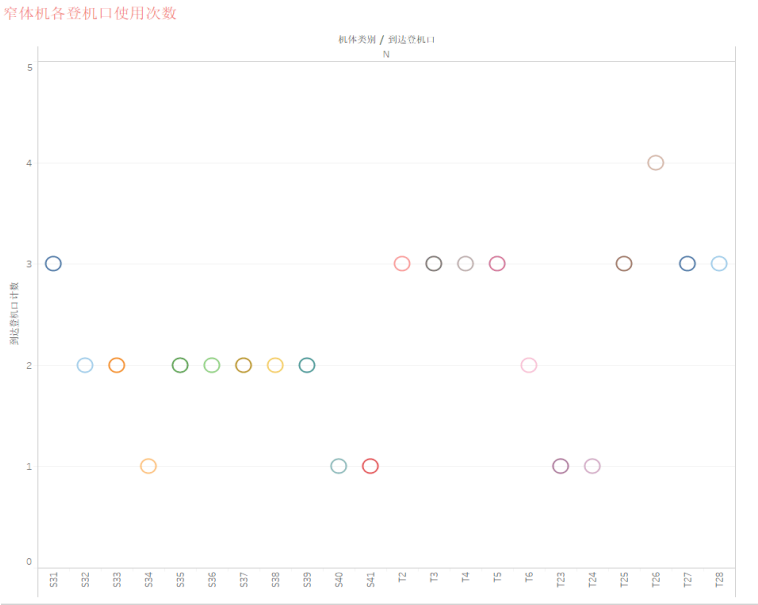


图 5: 问题1窄体机各登机口使用次数

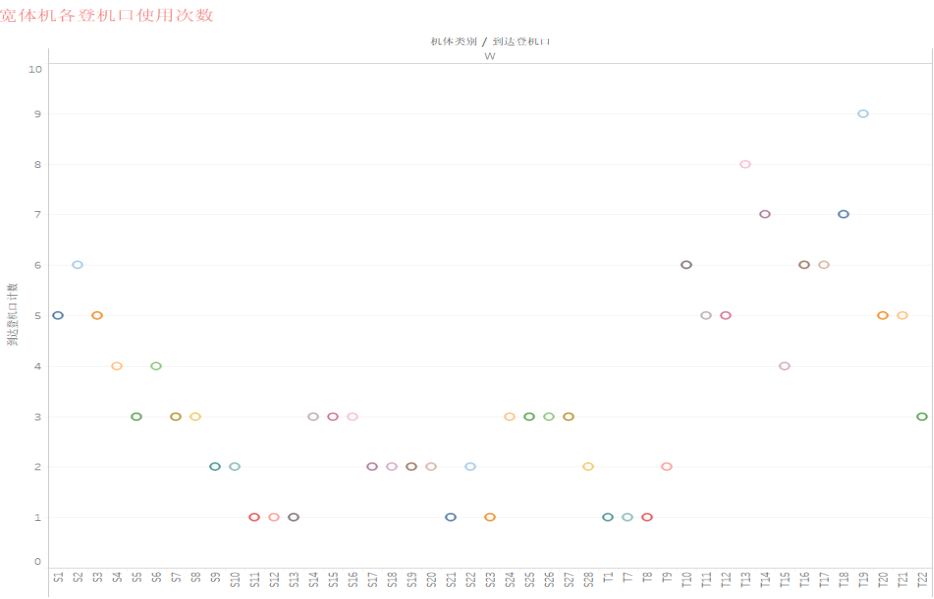


图 6: 问题1宽体机各登机口使用次数

为了更直观地感受各个时间段各登机口的使用情况，按航站楼S以及卫星厅T两种类型统计出各个时刻的使用情况，如下两张图可以感受到各登机口一天的使用密集度：



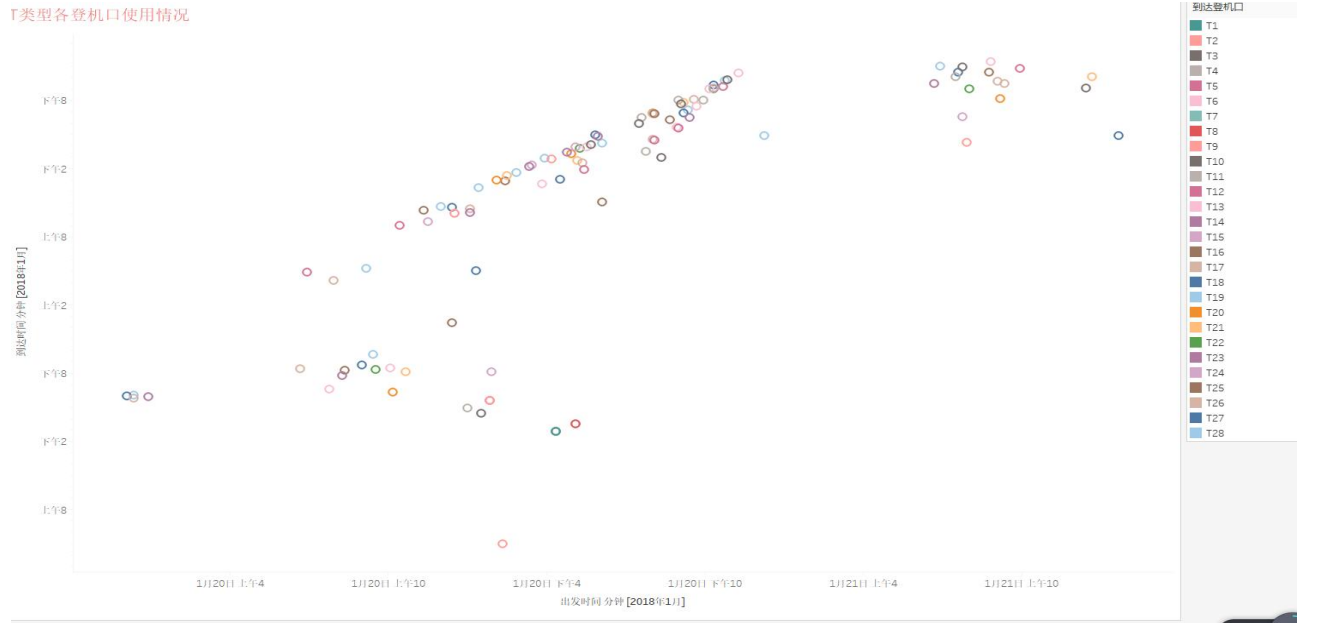


图 7: 问题1航站楼T各登机口使用情况

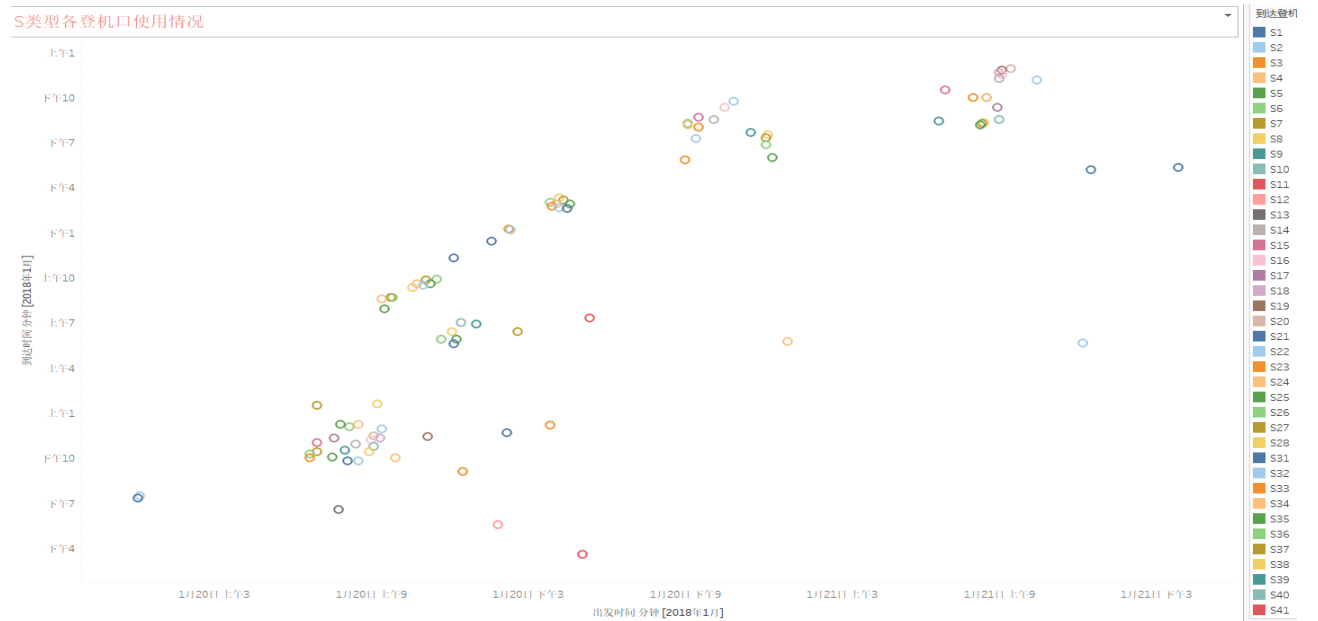


图 8: 问题1卫星厅S各登机口使用情况

## 6.8 问题2

针对问题二，具体的分配情况，详细内容见附件[问题2登机口分配情况]，假设 $N$ 个登机口分别被使用次数记为 $(k_1, k_2, \dots, k_N)$ ，临时停靠机位被使用的次数记为 $o$ ，各换乘航班的旅客数为 $(p_1, p_2, \dots, p_M)$ ，对应的行走时间 $(w_1, w_2, \dots, w_M)$ ，考虑中转旅客在航站楼-卫星厅之间的最短中转流程时间，此时设计的适应度函数可在问题1中的适应度函数的基础上改进为：

$$z_i = k_i - \frac{o^2}{k_i}$$



$$Fitness_2 = \max \sum_{i=1}^N (z_i - \frac{w_i^2}{z_i})$$

## 6.9 问题3

针对问题三，具体的分配情况，详细内容见附件[问题3登机口分配情况]假设 $N$ 个登机口分别被使用次数记为 $(k_1, k_2, \dots, k_N)$ ，临时停靠机位被使用的次数记为 $o$ ，各换乘航班的旅客数为 $(p_1, p_2, \dots, p_M)$ ，对应的行走时间 $(w_1, w_2, \dots, w_M)$ ，对应的换乘紧张度为 $\epsilon_1, \epsilon_2, \dots, \epsilon_M$ ，考虑中转旅客在航站楼-卫星厅之间的最短中转流程时间，以及航站楼-卫星厅乘坐捷运线时间，各个登机口区域之间的步行时间，此时设计的适应度函数可改进为：

$$v_i = z_i - \frac{w_i^2}{z_i}$$

$$Fitness_3 = \max \sum_{i=1}^N (v_i - \frac{\epsilon_i^2}{v_i})$$

# 7 模型的评价

## 7.1 模型的优点

本文主要使用遗传算法，针对不同的场景，不同的优化目标，通过对适应度函数的调整来对算法稍作修改，然后利用Python进行编程，基本上可以很好的实现模型，具体优点有如下几点：

1. 登机口的基本分配规则中的约束条件都可以很好的满足。
2. 本文中的优化模型可以最终可以得到最优解或者次优解。
3. 由于机场存在临时机位，该模型不仅能满足最小化使用登机口数量，同时也能最小化使用临时机位的个数。

## 7.2 模型的缺点

本文中的模型虽然基本上能很好的根据约束条件及优化目标得到结果，但是仍然存在一些不足，具体有如下几点：

1. 结果的优化程度随遗传算法迭代时间的增加而增加，因此若是迭代时间过短，则搜索范围较小，因此可能只能得到局部最优解，因此最终得到的可能是次优解而不是最优解。
2. 模型在遗传算子的设计中，由于时间关系，主要使用了选择和变异，交叉没有使用，后期在算法优化中可以将交叉也考虑进去。
3. 本文中的模型是根据不同的约束条件，进而通过修改不同的适应度函数来对算法进行修改，因此可能并不够全面。

## 参考文献

- [1] 戴顺南. 机场机位分配模型构建及算法实现[D].北京交通大学,2008.
- [2] 蒋延军. 机场停机位分配优化问题的研究[D].天津大学,2010.
- [3] Liu S, Chen W, Liu J. Optimizing airport gate assignment with operational safety constraints[C]// International Conference on Automation and Computing. IEEE, 2014:61-66.
- [4] Ghaleb M A, Suryahatmaja U S, Alharkan I M. Modeling and simulation of Queuing Systems using arena software: A case study[C]// International Conference on Industrial Engineering and Operations Management. IEEE, 2015:1-7.