

# 2016 年同济大学校内数学建模竞赛



小组成员：康 华 1451292 交运 交通规划 15900586293  
李曼姿 1350148 交运 交通规划 15221600257  
刘东亮 1451348 交运 道路工程 18117440515

# 游乐园疏导方案

## 摘要

人们在游乐园游玩时都希望花费最少的时间玩到最多的项目，从而达到体验度最优。而游乐园适时的引导可以大大提高游客的体验度。本文基于此，对游乐园的引导方案进行探讨。根据游乐园设施条件的不同，分为不可追踪个人轨迹和可以追踪个人轨迹的两种情况分别建模。

对于不可针对到个人的情况，面向排队等待的人群，建立模型一。首先衡量项目点附近等待时间不均衡程度，引入等待时间比和以排队人数加权的方差两种指标来确定何时进行引导。其次，在引导方案的确定上，考虑各项目点间相互影响，从而确定以当前所有人即场馆内人群，等待人群和在路上行走的人引导后消耗时间最少为目标的线性规划（LP）问题，考虑到了引导完成时刻各项目点排队人数的不同。在求解上，将除 A 点外各项目点排队游客离散的等待时间连续化，利用排队论计算平均等待时间避免了取整运算。

对于游玩后出项目点的游客，将项目点划分为五个等级，引入了供需比指标和时间系数指标综合评价，确定了等级划分标准。引入 logit 离散选择模型刻画游客对于信息的选择行为，进而反馈给信息发布层，以便更改信息发布策略。

对于有 RFID 为每个游客提供个性化引导的情况，建立了游客游览全部项目点情况下的时间依赖网络中 TSP 最短路径模型。根据游乐园当前人流状态预测各个项目点的排队人数随时间的动态变化，对非直接联通节点间建立虚拟弧，弧的权值设为两点间最短行程时间，使用 Dijkstra 算法对模型进行求解，最终得到游客的推荐游园路线。

对于某个项目点拥挤人数超出极限极易发生危险急需疏散的情况，建立模型四，将目标函数变为总体花费最短的时间完成疏散，同样转变为为线性规划问题求解。

上述模型分别利用 MATLAB 生成随机状态值，进行了算例的模拟求解。

根据皇家假日酒店历史预订数据信息，结合机器学习的支持向量机算法对各影响因素深入挖掘，建立起酒店房间预定量的模型。首先将包括季节、周末、节假日、暑假在内的影响因素定量化处理，增设了表示预定量突变程度的不确定系数；其次将一年划分为淡季、旺季、普通季，分类建模更精准地把握了变动规律。利用模型的淡季区间即 1 月 1 号至 3 月 10 号对 2016 年 1 月 1 号至 3 月 10 号的预定量进行预测，利用模型的普通季区间对 3 月 11 号至 3 月 31 号的预定量进行预测。

**关键词：**客流疏导 线性规划 logit 选择模型 支持向量机 Dijkstra 算法

## 一、问题重述

Youth 游乐园即将盛大开园，作为本市建有最多过山车的游乐园，受到了青少年的热捧。预计届时园区将迎来每天 1 万的大客流。如何根据客流情况，及时分流人群，为顾客提供游园线路引导，保障游客的游园体验显得尤为重要。

根据园区的整体规划，本文建立数学模型分析研究下面的问题：

- (1) 根据 Youth 乐园的规划图，在保障每位游客体验游乐设施的前提下，建立对每个游乐项目的等候游客进行游览提醒和疏导的模型，以达到游园体验最优。因此本文要根据游乐园设施条件的不同分别建立模型进行常规引导，同时为避免大客流的踩踏事故建立客流超标时的应急预案。

游乐项目	每场容纳游客数	每场持续时间
A	400	33分
B	30	1分15秒
C	50	2分30秒
D	30	2分30秒
E	100	5分
F	50	2分30秒
G	30	2分
H	30	1分30秒
I	20	1分30秒
J	50	2分

表 1. 每个游乐项目的时间安排



图 1. Youth 乐园规划图

(2) 本文根据皇家假日酒店历史预订数据信息,综合考虑影响房间预定量的主要因素(比如季节,工作日/周末,法定假日,暑期等)建立数学模型。并根据酒店 2015 年全年预定数据,预测 2016 年 1 月至 3 月每天预定房间数。

## 二、模型公共假设

1. 假设游客步速为 1m/s。
2. 假设游客对每个项目喜爱程度相同且只想体验一次。
3. 考虑到人数越多同一项目点相邻场次间换场时间越长,假设换场时间与每场容纳人数符合关系式换场时间=3√每场容纳人数。

## 三、符号定义

符号	含义
$V_i(i=1,2,...,11)$	分别代表游乐园的出入口和 $A,B,...,J$ 10 个项目点
$C_i(i=1,2,...,11)$	分别对应上述各点的容纳游客
$T_i(i=1,2,...,11)$	分别对应上述各点每场持续时间
$t_i(i=1,2,...,11)$	分别对应上述各点离下一场开始的时间
$t_0$	当前时间
$N_i(t_0)(i=1,2,...,11)$	当前在场所 $i$ 的等待人数
$x_{ij}(i=1,2,...,11,j=1,2,...,11)$	游客从项目点 $i$ 前往项目点 $j$ 的人数
$t_{ij}(i=1,2,...,11,j=1,2,...,11)$	游客从项目点 $i$ 转移至项目点 $j$ 在路途上耗费的时间
$t_w^i$	项目点 $i$ 排在末尾的游客的当前等待时间
$tt_i(i=1,2,...,11)$	项目点 $i$ 相邻两点的换场时间
$p_{ij}(i=1,2,...,11,j=1,2,...,11)$	体验完项目 $i$ 的游客选择前往项目 $j$ 的概率
$y_{ij}$	判别 $i$ 和 $j$ 是否相连的 0-1 变量
$Q_{ij}(t_0)$	$t_0$ 时刻从 $i$ 前往 $j$ 的路上的人
$\xi$	各项目点的人流量警戒线
$t_j^w$	游客到达 $V_i$ 时, 还需等待的时间
$n_i^a$	游客到达 $V_i$ 时, 到达时排队人数

$t_i^a$	游客到达 $V_i$ 的时间
$a_i$	$V_2$ 项目点活动结束后流出游客对 $V_i$ 等待人数是否产生影响
$Q_i^{(t)}$	第 $t$ 年第 $i$ 天的房间预定数量
$P(d)$	预订提前天数为 $d$ 的概率
$X_{ij}^{(t)}$	到达日期为第 $t$ 年第 $i$ 天的第 $j$ 个需求的房间数量
$N_i^{(t)}$	到达日期为第 $t$ 年第 $i$ 天的需求房间数量
$Y_{ij}^{(t)}$	第 $t$ 年第 $i$ 天第 $j$ 个订单预定的房间数量
$M_i^{(t)}$	第 $t$ 年第 $i$ 天预定的房间数量
$S_i^{(t)}$	第 $t$ 年第 $i$ 天所在季节
$W_i^{(t)}$	第 $t$ 年第 $i$ 天是否为周末
$H_i^{(t)}$	第 $t$ 年第 $i$ 天是否为法定假日
$V_i^{(t)}$	第 $t$ 年第 $i$ 天是否在暑假

表 2. 符号说明表

## 四、问题分析

问题一要求建立对游客的游览提醒和疏导模型。由于游乐园的设施状况未知，因此可分为两种情况考虑，即有 RFID 用户标识的和没有 RFID 用户标识但可采集到各处人流信息的。对于没有游客标识的，无法获知用户已游玩过的地点，因此只能将其同一对待，推荐的地点可能某些游客已游玩过。考虑排队的人群和项目点游玩出来的人群。对于排队人群他们急需知道是否值得继续等待，首先确定何时采取引导策略的标准，由于各项目点间人员流动相互影响，因此引导方案应联动显示，成为以当前整个游乐园的游客在引导结束后损耗时间最少为目标的线性规划，达到总体的最优。对于出场的游客他们急需知道接下来去哪里玩，由于不能有针对性地推荐，因此只能承担信息发布的作用。本文粗略地根据设定的标准将项目点服务水平分为 5 个等级，再结合游客离散选择性为的多元 logit 模型，得出游客对于信息发布的反应和由其带来的影响，将此影响反馈回信息发布层，制定信息发布策略，防止拥堵的转移。

对于有游客标识的情况，可针对每一游客定位，提供个性化的推荐路线。游客进门时即推荐一路线，此为有时间依赖性的旅行商问题，对于系统随时间的变化，根据现有各点的状态和路径上的流率，预测出游客到达时刻的等待时间，到达较远点的时刻要叠加上之前点上花费的时间由此迭代，得出最短时间路径。由于线路中可以一次游玩过所有点的线路只有一条，因此考虑游客可重复经过项目点，而以总游览时间最少为目标。游客依照推荐到达下一地点体验完成后重新搜索路径并推荐。

对于大规模，大客流量的游乐园，应急疏散方案是必须的，当某项目点的游客数量超过设定阈值时，应当立即将富余游客疏散至相邻点，但要满足相邻点的客流不超阈值。这时的

目标函数将转变为疏散时间最短而非等待时间最短。

因此，问题一的模型建立流程图如下所示：

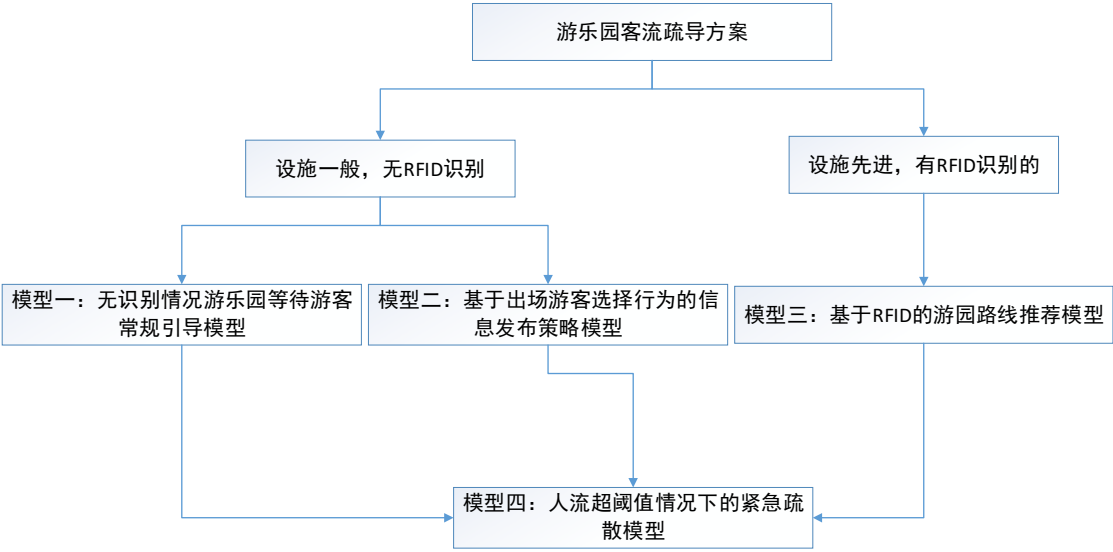


图 2. 问题一模型相互关系

问题二要求根据皇家酒店的历史订单量预测 2016 年的订单量。酒店每日到店量即需求量在本质上可以看作是一个时间序列，可以用时间序列分析的方法进行分析。该分析的目的在于根据该酒店 2015 年的已知数据建立模型，利用该模型来预测 2016 年的每日预订量。首先，酒店房间的预订量一部分呈现出以年为周期的循环变动趋势，其次可能存在发展的上升或下降进程，还有其他不确定因素的影响；此外，酒店房间预定量有明显的淡、旺、普通季的划分，三段时期游客变动的内在规律不同，因此分成三类建模有其合理性。

## 五、模型一：无识别情况游乐园等待游客常规引导模型<sup>[5]</sup>

### 5.1 本问假设

1. 由于本模型基于常规引导，故道路通行能力应足以满足任何情况下的引导方案，不会影响道路上人流的舒适度。
2. 假设符合条件的人群全部听从给出的引导建议。
3. 排队人群在接受到转移的引导信息后，同时从各地出发，不考虑人群流动的先后，且考虑到不同点间转移时间不同，以最远转移距离的人到达时刻作为调度完成时刻，先到达目的地的人默认为等待状态，不再移动。
4. 游乐园具备先进的信息采集设备，能够获取模型所需的参量，如每场剩余时间，等候人数等。
5. 假设在路上有既定走向的人均匀地分布在路上。

### 5.2 模型建立

本模型建立在无法对游客定位识别的基础上，系统无法获知在某地的等待用户的不同属性，即游客已经游玩过哪些地方，哪些地方是待游玩的场所，因而引导牌只能为用户提供通用的建议，若用户没有游玩过建议的场所，则听从引导而前往，若游玩过，则等在原地。利用场馆内的信息采集设备，每隔一定时间对相应信息进行更新。

#### 5.2.1 引导策略发布时间的评定

本模型考虑常规情况下的引导方案，即游乐园可以正常提供服务但是引导后会使得效率增高。这里首先引入指标判别何时需要引导。当各个项目点排队等待时间呈现明显的不均衡状态时，应当采取引导策略。

给出两种是否提供引导策略的评定方法：基于逐个项目点和基于区域的评价标准。

### 1. 基于逐个项目点的评价标准

逐个考察各项目点，只考虑是否应当将人流从该项目点引出。

对于单个项目点  $V_i$ ，与其相邻的项目点集为  $J$ 。只考虑一步式引导策略，即从当前点向相邻点引导。用  $V_i$  点与  $V_j$  点当前时刻排在末尾的人的等待时间  $t_w^i$  和  $t_w^j$  之比作为衡量标准，即

$$r_{ij} = \frac{t_w^i}{t_w^j},$$

显然当  $r < 1$  时，当前点的等待时间较少，服务水平较优。取  $i$  与各相邻项目点的等待时间之比的均值

$$\bar{r} = \frac{1}{n} \sum_{j \in J} r_{ij}$$

表示  $i$  点在相邻区域内等待时间的相对水平。

设定评分标准：

$$mark = 1 - \alpha \bar{r}_i,$$

其中  $\alpha$  为比例系数。显然分数越低，越应当采取措施引导人流从  $i$  转移出去。

### 2. 基于区域的评价标准

将以项目点  $V_i$  为中心，和与它相邻的项目点形成的项目点组合称为一个区域，以中心项目点  $V_i$  进行标定，记为区域  $i$ 。

要考察区域间的等待时间均衡水平，可以采用区域间等待时间均值的方差来衡量。等待时间的均值为

$$\bar{t}_w = \frac{1}{n} \sum_i t_w^i,$$

方差为

$$S_i^2 = \frac{1}{n} \sum_j (t_w^j - \bar{t}_w^i)^2。$$

由于各项目点的排队人数不同，显然排队人数越多，整体损耗的时间就越多，将上述方差改为以等待人数为权重的加权平均，即

$$S_i^2 = \frac{\sum_j N_j (t_w^j - \bar{t}_w^i)^2}{\sum_j N_j}。$$

进而得到

$$mark = 1 - \beta S_i^2,$$

其中  $\beta$  为比例系数。显然, 当分数越低时该区域越应采取引导措施。

### 5.2.2 引导方案

游客具有较好的体验度，即在最短的时间内玩到所有的项目，等待时间的增多会大大削弱游客的游玩热情。因此引导策略的目标就是尽可能减少游客在体验项目之外消耗的时间，包括排队等待时间和不同项目点之间转移的时间。

首先针对某一项目点的等待游客来说,给出的引导应是比较其继续在该项目点等待的时间和前往其他临近地点路程和等待的时间而做出的最优方案。如下面的逻辑语句显示:

```

if 在 i 地等待时间 < 前往相邻的 j 地路程时间与等待时间之和
then 等在 i 地
else
    前往 j 地

```

然而，整个游乐园体系是一个随时间动态变化的系统，各处的引导方案是相互影响的，当前的最佳方案并不代表一段时间后的最佳方案，由于各引导牌相互制约，游客跟从引导到达别处时前面可能又已排了更多的人。因此从整体的角度出发，确定目标函数为：

$$\min \sum \text{每人除体验项目外的损耗时间}$$

将规划图抽象为下述无向图  $G(V, E)$  :

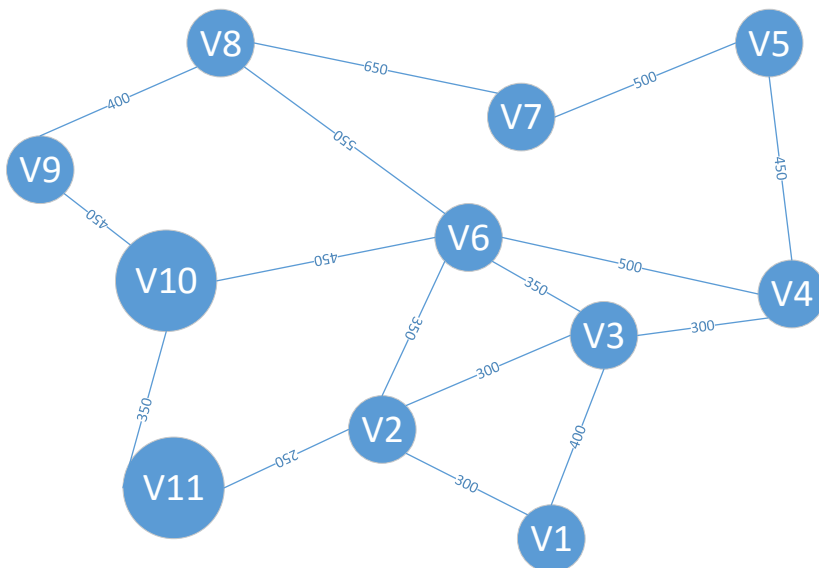


图 3 游乐园路网

邻接矩阵如下

$$Y = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$



其中

$$y_{ij} = \begin{cases} 1 & i \text{ 与 } j \text{ 相连} \\ 0 & \text{其他} \end{cases}$$

先分析相邻两点间的关系：

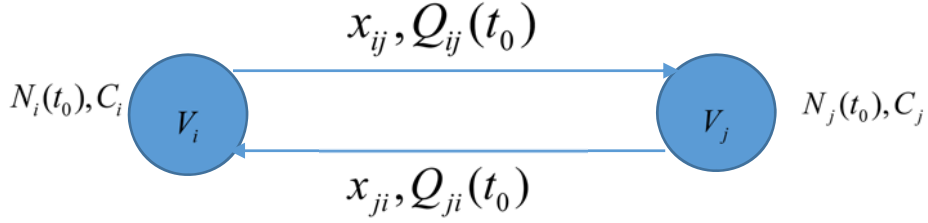


图 4. 相邻两点人流转移示意图

其中  $x_{ij}$  表示从项目点  $i$  前往项目点  $j$  的人数， $x_{ji}$  表示从项目点  $j$  前往项目点  $i$  的人数。 $Q_{ij}(t_0)$  表示在  $t_0$  时刻在项目点  $i$  和  $j$  之间的路上行走的要从  $i$  到  $j$  的人数， $Q_{ji}(t_0)$  表示在  $t_0$  时刻在项目点  $i$  和  $j$  之间的路上行走的要从  $j$  到  $i$  的人数。

对于出入点要特殊考虑，其容量设为 0，每场持续时间设为  $M$ ，距离场次结束时间设为  $M$ ，这样即可避免出入口游客的停留，使其迅速疏散到其他点。

对于游客的等待时间呈现出如下的阶梯状增长形式：

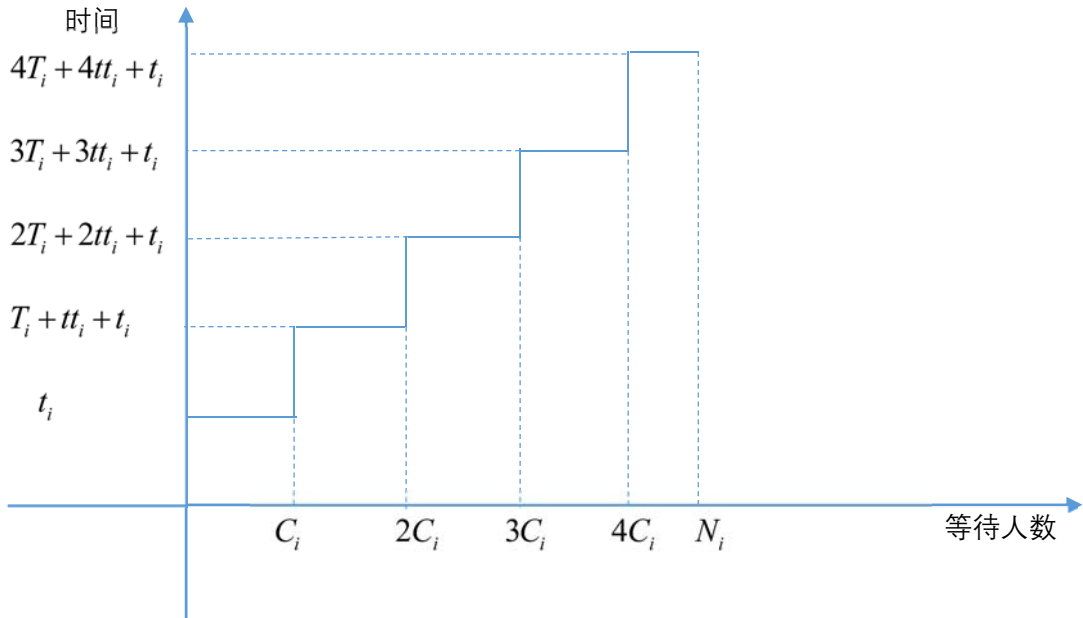


图 5. 游客在项目点排队等待时间规律图

本模型针对当前时刻  $t_0$  游乐园中所有人包括排队等待人群，场馆中游乐人群和在路上行走的人群，使他们的消耗时间最少，而引导策略的针对对象为排队等待人群，实现整个游乐园的等待人群的调度。目标函数取为在实现本次调度后的时刻  $t$  所有人的排队等待时间

最小。最远路径距离为 650 米，故本次调度的完成时刻应为  $t = t_0 + 650$ 。对于当前时刻  $t_0$ ，游玩的游客的等待时间为 0。正在场所内体验的游客从场所出去后随机地到达与其相连的场馆，设项目点  $i$  体验完的人前往项目点  $j$  的概率为  $p_{ij}$ 。调度实现后，当前时刻在路上的人都已到达目的地而接受调度的排队等待人群也已到达目的地，但是从项目点体验完毕的游客可能正在路上行走，由于他们是按照自己的意愿行走的，没有受到调度信息的影响，因此没有达到对他们优化的效果，因此这部分人的排队等待时间不计入目标函数中。

只需考察在调度完成时刻各项目点排队等待的人的总体等待时间最小即可。对于项目点  $i$  来说，其排队长度已发生变化，首先将调度时间  $t_d$  和与其相连各点的演出结束时间  $t_j$  和路程时间  $t_{ij}$  之和相比较，同时将调度时间  $t_d$  与该项目点的距离本场演出结束时间  $t_i$  相比较，项目点  $i$  在调度完成时刻的排队人数可以分下列四种情况计算：

1. 若  $t_d < t_j + t_{ij}$  且  $t_d < t_i$ ，则调度完成的  $t$  时刻，项目点  $i$  的排队人数  $N_i$  为

$$N_i(t_0) + \sum_{j=1}^{11} y_{ji} x_{ji} - \sum_{j=1}^{11} y_{ij} x_{ij} + \sum_{j=1}^{11} y_{ij} [Q_{ji}(t_0) - Q_{ij}(t_0)];$$

2. 若  $t_d < t_j + t_{ij}$  且  $t_d > t_i$ ，则调度完成的  $t$  时刻，项目点  $i$  的排队人数  $N_i$  为

$$N_i(t_0) + \sum_{j=1}^{11} y_{ji} x_{ji} - \sum_{j=1}^{11} y_{ij} x_{ij} - C_i \left( \left[ \frac{t_d - t_i}{T_i + t t_i} \right] + 1 \right) + \sum_{j=1}^{11} y_{ij} [Q_{ji}(t_0) - Q_{ij}(t_0)];$$

3. 若  $t_d > t_j + t_{ij}$  且  $t_d < t_i$ ，设置 0-1 变量  $k_{ij}=1$ ， $k_{ij} = \begin{cases} 1 & \text{符合上述条件} \\ 0 & \text{其他} \end{cases}$ ，而此时与  $i$

相连的项目点  $j$  结束场次的人已到达  $i$  的有  $\left[ \frac{t_d - (t_{ij} + t_j)}{T_j + t t_j + t_{ij}} \right] + 1$  批次的人，以  $n_{ij}$  表示，

则调度完成的  $t$  时刻，项目点  $i$  的排队人数  $N_i$  为：

$$N_i(t_0) + \sum_{j=1}^{11} y_{ji} x_{ji} - \sum_{j=1}^{11} y_{ij} x_{ij} + \sum_{j=1}^{11} \sum_{s=1}^{n_{ij}} k_{ij} y_{ij} C_j p_{ij} + \sum_{j=1}^{11} y_{ij} [Q_{ji}(t_0) - Q_{ij}(t_0)];$$

4. 若  $t_d > t_j + t_{ij}$  且  $t_d > t_i$ ，与上述同理，则调度完成的  $t$  时刻，项目点  $i$  的排队人数  $N_i$  为：

$$N_i(t_0) + \sum_{j=1}^{11} y_{ji} x_{ji} - \sum_{j=1}^{11} y_{ij} x_{ij} + \sum_{j=1}^{11} \sum_{s=1}^{n_{ij}} k_{ij} y_{ij} C_j p_{ij} + \sum_{j=1}^{11} y_{ij} [Q_{ji}(t_0) - Q_{ij}(t_0)] - C_i \left( \left[ \frac{t_d - t_i}{T_i + t t_i} \right] + 1 \right)。$$

调度完成时项目点  $i$  的距离演出结束时间是：

$$t'_i = T_i - (t_d - t_i - [\frac{t_d - t_i}{T_i + tt_i}](T_i + tt_i))(t_d > t_i)$$

$$t'_i = t_i - t_d (t_d < t_i)$$

项目点  $i$  相应的排队总时间为：

$$T_i^{\text{总}} = t'_i * N_i + (T_i + tt_i)([\frac{N_i}{C_i}] - 1 + 1)([\frac{N_i}{C_i}] - 1) + (T_i + tt_i)(N_i - C_i[\frac{N_i}{C_i}])$$

因此可归结为如下的规划问题：

$$\min \sum_{i=1}^{11} T_i^{\text{总}} + 0.5 \sum_{i=1}^{11} \sum_{j=1}^{11} Q_{ij} t_{ij} + \sum_{i=1}^{11} \sum_{j=1}^{11} x_{ij} t_{ij}$$

$$s.t. \sum_{j=1}^{11} y_{ij} x_{ij} + x_{ii} = N_i(t_0)$$

$$N_i(t_0) + \sum_{j=1}^{11} y_{ji} x_{ji} - \sum_{j=1}^{11} y_{ij} x_{ij} + \sum_{j=1}^{11} Q_{ji} - \sum_{j=1}^{11} Q_{ij} \geq 0$$

$$x_{ij} \geq 0, i = 1, 2, \dots, 11, j = 1, 2, \dots, 11$$

$$y_{ij} \text{ 为 } 0-1 \text{ 变量}$$

相应的流程图如下：

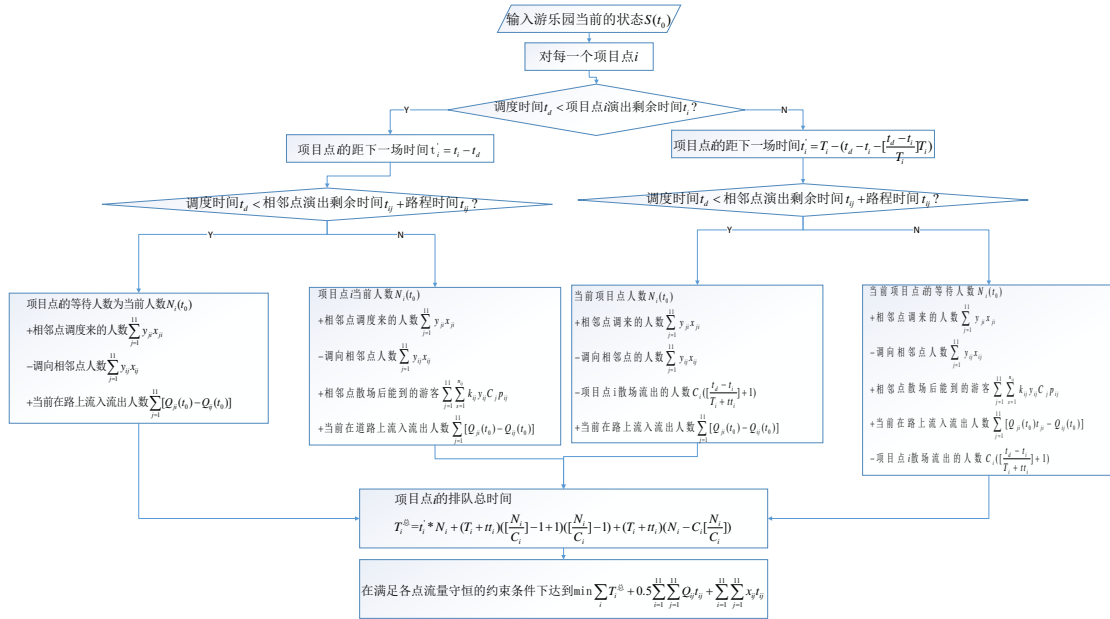


图 6. 模型一计算流程图

由于计算时间时涉及到对于人数的取整计算，难以求解。因此，需要进行合理的假设。由于 A 点的每场持续时间过长，当等待人数超过 400 人时，等待时间必定大幅增加，因此可确定 A 点在调度完成时的  $[\frac{N_2}{C_2}] = 0$ 。对于其他项目点，由于其持续周期短，可以将离散化的等待时间变为连续曲线。

根据排队论的相关知识，假设某一项目点人群的到达间隔服从负指数分布，平均到达率

为  $\lambda_i$ ，项目点的平均服务率为  $\mu_i$ ， $\mu_i = \frac{C_i}{T_i + tt_i}$ ，服务强度  $\rho_i = \frac{\lambda_i}{\mu_i}$ 。根据相关公式的推

导可以得出：

$$\text{项目点的平均人数 } \bar{n}_i = \frac{\rho_i}{1 - \rho_i} = \frac{\lambda_i}{\mu_i - \lambda_i} ;$$

$$\text{项目点的平均消耗时间为 } \bar{d}_i = \frac{\bar{n}_i}{\lambda_i} ;$$

$$\text{排队中的平均等待时间为 } \bar{w}_i = \bar{d}_i - \frac{1}{\mu_i} .$$

由此，上述除 A 点外每个项目点的总的等待时间可表示为下式  $T_i^{\text{总}} = N_i * (\frac{1}{\mu_i - \lambda_i} - \frac{1}{\mu_i})$ 。

A 点的等待时间还按离散时间计算：

$$T_2^{\text{总}} = t_2' * N_2 .$$

由此再计算相应的目标函数。

### 5.3 算例求解

由于游乐场的状态是实时变化的，只能根据具体的算例来进行计算模拟。

用 MATLAB 生成随机数的方法，产生当前的状态，从而减少人为给出算例的主观性。

$$t_0 \text{ 时刻，路径上的流量矩阵 } Q = \begin{pmatrix} 0 & 91 & 22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 77 & 0 & 80 & 0 & 0 & 66 & 0 & 0 & 0 & 0 & 62 \\ 92 & 47 & 0 & 92 & 0 & 35 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 & 69 & 39 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 89 & 0 & 59 & 0 & 0 & 0 & 0 \\ 0 & 76 & 58 & 69 & 0 & 0 & 0 & 98 & 0 & 75 & 0 \\ 0 & 0 & 0 & 0 & 84 & 0 & 0 & 77 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 91 & 33 & 0 & 58 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 60 & 0 & 23 & 0 \\ 0 & 0 & 0 & 0 & 0 & 22 & 0 & 0 & 25 & 0 & 28 \\ 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 26 & 0 \end{pmatrix} ;$$

当前距离下场开始的时间（出入口处的时间设为一个很大的值，这里取为 10000）

$$t = (10000 \ 1668 \ 66 \ 26 \ 109 \ 171 \ 166 \ 88 \ 85 \ 47 \ 61)^T ;$$

$$\text{每个点的容纳量 } C = (0 \ 400 \ 30 \ 50 \ 30 \ 100 \ 50 \ 30 \ 30 \ 20 \ 50)^T ;$$

每个项目点每场持续时间：

$$T = (10000 \ 1980 \ 75 \ 150 \ 150 \ 300 \ 150 \ 120 \ 90 \ 90 \ 120)^T ;$$

每个项目点相邻两场之间的换场时间：

$$tt = (0 \ 60 \ 16 \ 21 \ 16 \ 30 \ 21 \ 16 \ 16 \ 13 \ 21)^T ;$$

各项目点间的距离矩阵为:

$$D = \begin{pmatrix} 0 & 300 & 400 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 300 & 0 & 300 & 0 & 0 & 350 & 0 & 0 & 0 & 0 & 250 \\ 400 & 300 & 0 & 300 & 0 & 350 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 300 & 0 & 450 & 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 450 & 0 & 0 & 500 & 0 & 0 & 0 & 0 \\ 0 & 350 & 350 & 500 & 0 & 0 & 0 & 550 & 0 & 450 & 0 \\ 0 & 0 & 0 & 0 & 500 & 0 & 0 & 650 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 550 & 650 & 0 & 400 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 400 & 0 & 450 & 0 \\ 0 & 0 & 0 & 0 & 0 & 450 & 0 & 0 & 450 & 0 & 350 \\ 0 & 250 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 350 & 0 \end{pmatrix}$$

每点当前的等待人数向量:

$$N = (51 \ 330 \ 8 \ 20 \ 16 \ 78 \ 125 \ 72 \ 5 \ 24 \ 79)^T$$

每个项目点出来的人数选择从  $i$  向  $j$  的概率矩阵:

$$p = \begin{pmatrix} 0 & 0.5112 & 0.4888 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.6038 & 0 & 0.1026 & 0 & 0 & 0.0652 & 0 & 0 & 0 & 0 & 0.2285 \\ 0.1001 & 0.2475 & 0 & 0.1716 & 0 & 0.4809 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5379 & 0 & 0.0308 & 0.4313 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3889 & 0 & 0.6111 & 0 & 0 & 0 & 0 \\ 0 & 0.1716 & 0.2953 & 0.1308 & 0 & 0 & 0 & 0.3079 & 0 & 0.0944 & 0 \\ 0 & 0 & 0 & 0 & 0.5127 & 0 & 0 & 0.4873 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2832 & 0.3667 & 0 & 0.3501 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5819 & 0 & 0.4181 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8306 & 0 & 0 & 0.1423 & 0 & 0.0271 \\ 0 & 0.3889 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6111 & 0 \end{pmatrix}$$

每个项目点的平均服务率为:

$$\mu = (0 \ 0.2174 \ 0.3297 \ 0.2924 \ 0.1807 \ 0.3030 \ 0.2924 \ 0.2206 \ 0.2830 \ 0.1942 \ 0.3546)^T ;$$

每个项目点的平均到达率为:

$$\lambda = (0 \ 0.2081 \ 0.1600 \ 0.2340 \ 0.0256 \ 0.1278 \ 0.2678 \ 0.1748 \ 0.2716 \ 0.1273 \ 0.2820)^T ;$$

出入口点的平均等待时间直接设为很大的数 10000, 则得到各项目点的平均等待时间向量为

$$\bar{w} = (10000 \ 103 \ 3 \ 14 \ 1 \ 2 \ 37 \ 17 \ 84 \ 10 \ 11)^T .$$

而在计算 A 点的等待时间时按照离散时间进行计算。  $t_2' = t_2 - t_d = 1018$  。

得到修正后的各点等待时间  $w = (10000 \ 1018 \ 3 \ 14 \ 1 \ 2 \ 37 \ 17 \ 84 \ 10 \ 11)^T$  。

编程求解可得调度完成后各点的人数, 由于概率为小数, 人数不是整数, 可在最后取整

$$\begin{pmatrix} -x_{12} - x_{13} + x_{21} + x_{31} + 121.6628, \\ x_{11,2} + x_{12} - x_{21} - x_{23} - x_{26} - x_{2,11} + x_{32} + x_{62} + 316.0185, \\ x_{13} + x_{23} - x_{31} - x_{32} - x_{34} - x_{36} + x_{43} + x_{63} - 2291.3, \\ x_{34} - x_{43} - x_{46} + x_{64} - 2169.9, \\ x_{45} - x_{54} - x_{57} + x_{75} - 1210, \\ x_{10,6} + x_{26} + x_{36} + x_{46} - x_{62} - x_{63} - x_{64} - x_{68} - x_{6,10} + x_{86} - 2218.5, \\ x_{57} - x_{75} - x_{78} + x_{87} - 1578.6, \\ x_{68} + x_{78} - x_{86} - x_{87} - x_{89} + x_{98} - 1514.5, \\ x_{10,9} + x_{89} - x_{98} - x_{9,10} - 1949.2, \\ -x_{10,6} - x_{10,9} - x_{10,11} + x_{11,10} + x_{6,10} + x_{9,10} - 1158.3, \\ x_{10,11} - x_{11,2} - x_{11,10} + x_{2,11} - 2630.8 \end{pmatrix}$$

总的损耗时间为

$$\begin{aligned} \text{totaltime} = & 442x_{10,6} + 524x_{10,9} + 351x_{10,11} + 1257x_{11,2} + 349x_{11,10} + 318x_{12} - 597x_{13} \\ & + 282x_{21} - 715x_{23} - 666x_{26} - 757x_{2,11} + 1397x_{31} + 1315x_{32} + 311x_{34} + 349x_{36} \\ & + 289x_{43} + 437x_{45} + 488x_{46} + 463x_{54} + 536x_{57} + 1366x_{62} + 351x_{63} + 512x_{64} + 565x_{68} \\ & + 458x_{6,10} + 464x_{75} + 630x_{78} + 535x_{86} + 670x_{87} + 467x_{89} + 333x_{98} + 376x_{9,10} + 486960 \end{aligned}$$

结合约束条件，可利用 MATLAB 求解线性规划问题，得到最优解：

$$\begin{aligned} x_{11} = 40, x_{21} = 33, x_{31} = 8, x_{12} = 48, x_{32} = 19, x_{62} = 27, x_{11,2} = 71, x_{13} = 37, x_{23} = 72, \\ x_{33} = 36, x_{43} = 30, x_{63} = 45, x_{34} = 54, x_{44} = 46, x_{64} = 58, x_{45} = 22, x_{55} = 28, x_{75} = 3, \\ x_{26} = 178, x_{36} = 36, x_{46} = 35, x_{66} = 42, x_{86} = 61, x_{10,6} = 7, x_{57} = 73, x_{77} = 51, x_{87} = 44, x_{78} = 123, \\ x_{98} = 93, x_{89} = 78, x_{10,9} = 40, x_{6,10} = 61, x_{10,10} = 47, x_{2,11} = 44, x_{10,11} = 41, x_{11,11} = 74 \end{aligned}$$

总损耗时间为 38600s。

## 六、模型二：基于出场游客选择行为的信息发布策略模型<sup>[5]</sup>

### 6.1 本问假设

1. 假设所有景区的硬件设施水平相同，在确定服务等级时只考虑动态变化信息的影响。

### 6.2 模型建立

对于在项目点  $i$  游玩过后出场的游客，给予他们及时的游玩引导是必要的。由于不知其已游玩过的场所，游乐园扮演的角色只能局限于各场馆当前状况的信息发布者。但是发布信息的策略是值得研究的。由于各场馆的排队等待时间是实时变化的，这里可以简单地将各场馆的状况划分为 5 个等级，即 A, B, C, D 和 E，分别用绿，黄，橘黄，红，暗红来表示以显示其服务水平。景区的信息可以是真实的信息发布，也可以是有引导和偏向性的行为。如果仅限于对真实信息的发布，对某场馆显示为绿色则可能导致周围场馆的游客全部汇入而导致拥挤转移到该地。因此，游乐园的信息发布策略应考虑人们对于信息的选择行为及其反馈的影响。

本模型的基本流程图如下：

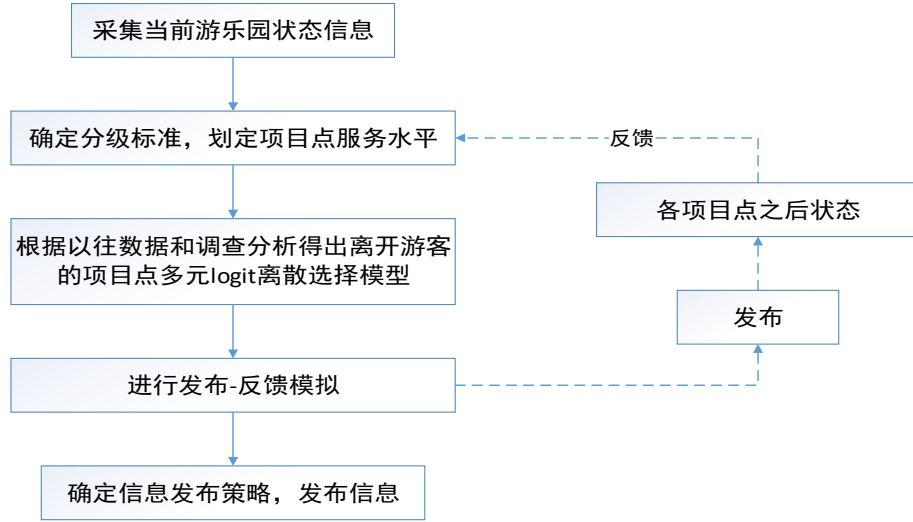


图 7. 模型二基本结构

### 6.2.1 服务水平的划分

不同景区的服务水平关系到游客的体验度。首先对于各项目点服务水平的等级划分，可根据每个项目点的状态确定。

首先引入类供需比指标，定义为  $\frac{C_i}{N_i}$ ，它相当于项目点的可容纳量即供给水平与排队等

待人数即需求水平形成的供需比，显然此指标越大意味着项目点的服务能力越大，服务水平越好。

另外，项目点的服务水平还与等待时间的大小有关，引入时间系数指标，定义为  $\frac{t_i}{T_i + tt_i}$ ，即等待时间占一个周期持续时间（包括每场持续时间和换场时间）的比例，它反

映了场次可提供服务的准备时间，显然，此指标越大，服务水平越低。

对于景区服务水平的划定，若根据上述指标的具体数值对不同的场馆分别定界则显得繁复，因此，本文将上述指标综合起来，将其映射到  $[0, 1]$  区间上，确定景区评级的基本计算公式如下：

$$mark = 1 - \frac{1}{(\alpha \frac{C_i}{N_i} + \beta \frac{t_i}{T_i + tt_i}) + 1};$$

上面的式子将两个指标整合在一起，权重  $\alpha, \beta$  显示了它们所占的比重。

将上述指标的组合  $(\alpha \frac{C_i}{N_i} + \beta \frac{t_i}{T_i + tt_i})$  定义为服务系数，用  $\varepsilon$  表示。

上述公式可简化为  $mark = 1 - \frac{1}{1 + \varepsilon}$ ；

其图像如下图：

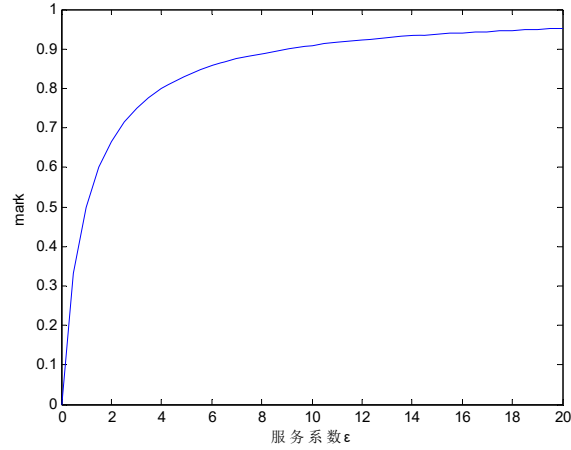


图 8. 评分与服务系数曲线图

由上图可以看出，随服务系数的减小，分数的减小逐渐加快，因而在横坐标上表现的区间范围越来越小，因此对于服务等级由高到低变化时，分数划分区间应当越来越大。可以粗略地将景区的等级划分为：

Rank	Mark
A	0.95~1
B	0.85~0.95
C	0.7~0.85
D	0.3~0.7
E	<0.3

表 3. 评分与等级划分

对于不同的景区也有特殊的情况，对于 A 景区来讲，由于其每场的间隔时间过长，因此当等待的游客人数超过 400 人则把其等级评定为 E 级，因为继续排队等待的人至少要等待 34 分钟，会造成大量的时间浪费。

### 6.2.2 人群选择行为

根据上述等级划分的方法，对于每个场馆的等级可以得到一真实水平，但是单纯对于真实水平的发布有可能会当导致当前畅通的场馆在之后的时间人们蜂拥而至。因此要充分考虑人们对于信息的选择行为。

实际情况下，人的行为是复杂的，对于某刺激做出反应的机制可以简化为

$$Response = \lambda \times Stimulus ;$$

其中  $\lambda$  是比例因子，反映人对于刺激的反应强度系数或灵敏度。

由此依照离散选择行为的 logit 模型[1-4]，它建立在随机效用最大化的理论上，其随机效用方程为：

$$U_{nj} = V_{nj} + \varepsilon_{nj} ;$$

$U_{nj}$  表示第  $n$  个决策者选择第  $j$  个替代品的效用。其中  $V_{nj}$  是可观测到的效用， $\varepsilon_{nj}$  是不能观测到的效用，通常假设其为独立同分布的随机变量。经推导可得出：



$$p_{ni} = \frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}} ;$$

可根据上面刺激-反应关系式设定效用的可观测部分  $V_{ni}$  是解释变量  $x_{ni}$  的线性关系式，比例系数向量为  $\beta$  。因此，得到多元 logit 模型的普遍应用形式：

$$p_{ni} = \frac{e^{\beta' x_{ni}}}{\sum_j e^{\beta' x_{nj}}}。$$

由此，对于游客游玩过本项目后对于不同路径的选择概率可以确定人员的大致走向。游客的选择集为与当前项目点相连的项目点，解释变量  $x_{nj}$  可以包括项目点的情况即分级指标，游客自身属性如年龄，景区熟悉度，对信息的信任度等。在通过 SP 调查得到相应数据后，可以通过极大似然估计得到相应的系数向量  $\beta$  ，由此得到人群选择的大致走向。

### 6.2.3 信息发布策略

信息发布策略可通过发布-反馈模拟实验的方法来修改探究。

首先确定每个项目点信息发布的频率不同，由项目点项目的循环周期决定，每场游客出馆后都应当对引导信息进行更新。

设游乐园当前的状态为  $S(t_0)$  ，利用信息采集设备得到当前拥堵情况的真实分级状况后，探究此信息对人们选择行为的影响，在项目点  $i$  对于每个游客  $n$  选择项目点  $j$  的概率为  $p_{nj}$  ，若项目点  $i, j$  不相连，则  $p_{nj}=0$  ，  $\sum_j p_{nj}=1$ 。将所有游客的选择某路径的概率叠加起来可得到总体选择该路径的人数期望  $E_{ij} = \sum_n p_{nj}$  。则对于总体来说，选择项目

点  $j$  的人数占比可取为  $p_{ij} = \frac{E_{ij}}{n} = \frac{E_{ij}}{C_i}$  ，由此可得到状态转移矩阵 P。

转移人数为  $C_i \begin{pmatrix} p_{i1} \\ \vdots \\ p_{ij} \\ \vdots \\ p_{i11} \end{pmatrix}$  。除游玩出来人群外其他人自然的走动变化记为  $D(t_0)$  ，可得

到  $t$  时刻的状态：

$$S(t) = S(t_0) + D(t_0) + \begin{pmatrix} C_1 & \cdots & C_i & \cdots & C_{11} \end{pmatrix} \begin{pmatrix} p_{11} & \cdots & p_{1,11} \\ & \ddots & \\ & & p_{ij} \\ & & & \ddots \\ p_{11,1} & \cdots & & p_{11,11} \end{pmatrix}。$$

根据预测的  $t$  时刻的状态对信息的发布进行调整。

在此，本文仅给出一种粗略的判别方法，在进行发布-反馈模拟后，如果发现某项目点的等级下降两级及以上或达到 E 级，则在发布其信息时就在真实水平的基础上降低一个等级，使得该景点对于游客的吸引程度降低，从而减少到达量，以防拥堵的转移。

## 七、模型三：基于 RFID 的个性化游园路线推荐模型<sup>[5]</sup>

随着科技的发展，智慧游乐园逐渐兴起。可利用用户使用的 APP 对用户进行实时定位，记录其已游玩过的项目点，为用户提供完善的个性化服务。

### 7.1 模型假设

- 1、假设一段时间内各个项目点间流率近似不变
- 2、该游客通过非直达路线时，在路过的节点不作停留
- 3、假设各个项目点的活动时间为固定周期，不因人数而改变

### 7.2 模型建立

#### 7.2.1 初始路线

设当前该游客当前在  $V_i$ ，从  $V_i$  出发，前往  $V_j$

此时，各个点间流率可以根据各个项目点间人数计算得，例： $V_m$  前往  $V_n$  路途随时间的流率为  $\mu_{mn}$ ，

$$\mu_{mn} = \frac{x_{mn}}{t_{mn}}$$

在游客从  $V_i$  移动到  $V_j$  过程中用时  $t_{ij}$ ，则游客到达  $V_j$  的时间

$$t_j^a = t_{ij}$$

这段时间内，游客从  $V_m$  向  $V_n$  移动了  $\mu_{mn} t_j^a$ ，则对于  $V_j$  来说流出的人数为

$$\sum_{m=1}^n \mu_{jm} t_j^a$$

进入的人数为

$$\sum_{m=1}^n \mu_{mj} t_j^a$$

所以，对于  $V_j$  来说，当该游客从  $V_i$  移动到  $V_j$  时， $V_j$  处的排队人数为

$$n_j^a = n_j - \sum_{m=1}^n \mu_{jm} t_j^a + \sum_{m=1}^n \mu_{mj} t_j^a$$

对于各个项目点的活动结束而流出的游客，除  $V_2$  点以外，由于其活动持续时间较短，其活动结束流出的人流可认为对于各通道的流率随时间的变化没有影响。

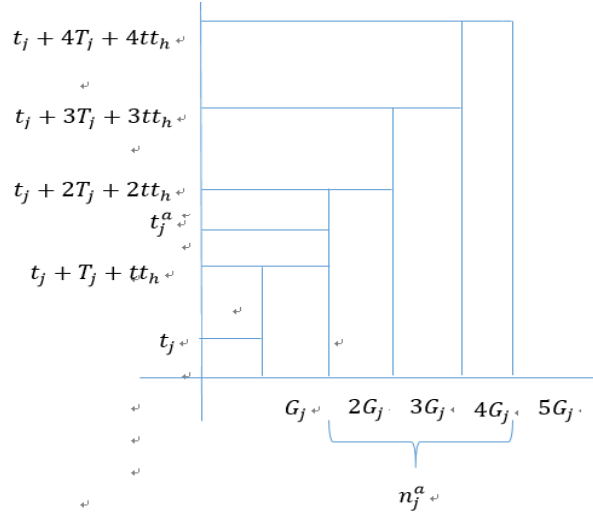


图 9.  $V_j$ 点时间轴图

所以，该游客将处于自到达以后的第 $\left[\frac{n_j^a}{G_j}\right] + 1$ 场，则该游客在 $V_j$ 的等待时间为

$$t_j^w = \left( \left[ \frac{n_j^a}{G_j} \right] + 1 \right) * (T_j + tt_j) - \left[ t_j^a - \left( \left[ \frac{t_j^a - t_j}{T_j + tt_j} \right] * (T_j + tt_j) + t_j \right) \right]$$

简化后

$$t_j^w = \left( \left[ \frac{n_j^a}{G_j} \right] + \left[ \frac{t_j^a - t_j}{T_j + tt_j} \right] + 1 \right) * (T_j + tt_j) - t_j^a + t_j$$

### 7.2.2 重复计算

游客从 $V_j$ 前往 $V_h$ ，则该游客到达 $V_h$ 的时间为

$$\begin{aligned} t_h^a &= t_a^j + t_j^w + t_{jh} + (T_j + tt) \\ &= \left( \left[ \frac{\sum_{m=1}^n \mu_{jm} t_j^a - \sum_{m=1}^n \mu_{mj} t_j^a + n_j}{C_i} \right] + \left[ \frac{t_j^a - t_j}{T_j + tt_j} \right] + 2 \right) * (T_j + tt_j) + t_j + t_{jh} \end{aligned}$$

对于 $V_h$ 来说在 $t_h^a$ 时间内流出的人数为

$$\sum_{m=1}^n \mu_{hm} t_h^a$$

进入的人数为

$$\sum_{m=1}^n \mu_{mh} t_h^a$$

所以，对 $V_h$ 来说， $n_h^a$ 时的排队人数为

$$n_h^a = n_j - \sum_{m=1}^n \mu_{hm} t_h^a + \sum_{m=1}^n \mu_{mh} t_h^a$$

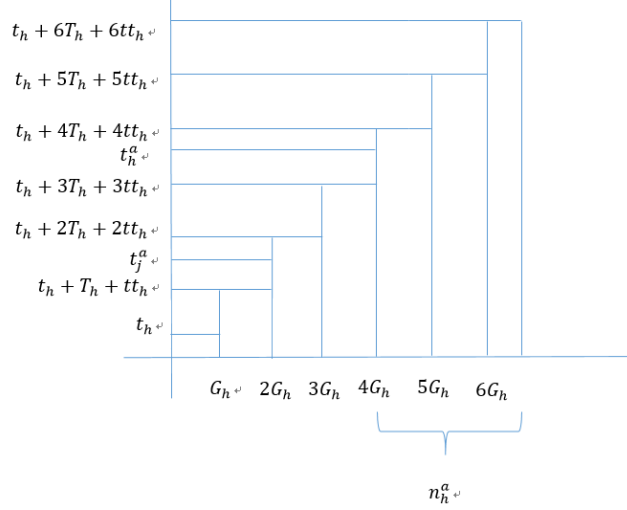


图 10.  $V_h$ 点时间轴图

所以，该游客将处于自到达以后的第 $\left\lceil \frac{n_h^a}{G_h} \right\rceil + 1$ 场，则该游客在 $V_h$ 的等待时间为

$$t_h^w = \left( \left\lceil \frac{n_h^a}{G_h} \right\rceil + 1 \right) * (T_h + tt) - \left[ t_h^a - \left( \left\lceil \frac{t_h^a - t_h}{T_h + tt_h} \right\rceil * (T_h + tt_h) + t_h \right) \right]$$

简化后

$$t_h^w = \left( \left\lceil \frac{n_h^a}{G_h} \right\rceil + \left\lceil \frac{t_h^a - t_h}{T_h + tt_h} \right\rceil + 1 \right) * (T_h + tt_h) - t_h^a + t_h$$

### 7.2.3 $V_2$ 点结束客流影响

特别地，对于到达 $V_2$ 点，由于其项目间隔时间大，容纳人数多， $V_2$ 通往各个项目点的流率随时间的变化受其影响过大，所以需单独考虑。

当 $T_2 - t_2 \geq \max\{t_{2i}, i = 1 \dots n\}$  时可认为上一场结束流出的人流已经各自前往目的地并走完与 $V_2$ 相连的通道，对流率没有影响，所以流率可按照当前流率计算，对于 $t_2^a$ 时间内 $V_2$ 活动结束流出的人进行附加， $t_2^a$ 时间内流出的游客为 $\sum_{m=1}^n \mu_{m2} t_2^a + \left( \left\lceil \frac{t_2^a - t_2}{T_2 + tt_2} \right\rceil + 1 \right) * G_2$ ，

该情况下， $t_2^a$ 时 $V_2$ 的排队人数为：

$$n_2^a = n_2 + \sum_{m=1}^n \mu_{m2} t_2^a - \sum_{m=1}^n \mu_{m2} t_2^a - \left( \left\lceil \frac{t_2^a - t_2}{T_2 + tt_2} \right\rceil + 1 \right) * G_2$$

当 $T_2 - t_2 \leq \min\{t_{2i}, i = 1 \dots n\}$  时可认为上一场结束流出的人流正走在与 $V_2$ 相连的通道上，此时上一场流出的游客对流率产生影响。从 $V_2$ 流向各方向的流率可计算为

$$= \frac{x_{2m} - C_2 \frac{x_{2m}}{\sum_{s=1}^n x_{2s}}}{t_{2m}} = \mu_{2m} - C_2 \frac{x_{2m}}{t_{2m} \sum_{s=1}^n x_{2s}}$$

所以，在 $t_2^a$ 时间内，所有流出的游客为 $\sum_{m=1}^n \mu_{m2} t_2^a + \left( \left\lceil \frac{t_2^a - t_2}{T_2} \right\rceil + 1 \right) G_2$ 。该情况下， $t_2^a$ 时 $V_2$ 的排队人数为：

$$n_2^a = n_2 + \sum_{m=1}^n \mu_{2m} t_2^a - \sum_{m=1}^n \mu_{m2} t_2^a - \sum_{m=1}^n \frac{x_{2m} c_2 t_2^a}{t_{2m} \sum_{s=1}^n x_{2s}} - \left( \left\lfloor \frac{t_2^a - t_2}{T_2 + t t_2} \right\rfloor + 1 \right) C_2$$

当  $T_2 - t_2 > \min\{t_{2i}, i = 1 \dots n\}$  且  $\left\lfloor t_2^a - \left( \left\lfloor \frac{t_2^a - t_2}{T_2 + t t_2} \right\rfloor * T_2 + t_2 \right) \right\rfloor < \max\{t_{2i}, i = 1 \dots n\}$  时，部分线路的流率受上一场  $V_2$  活动流出游客的影响，部分不受上一场  $V_2$  活动流出游客的影响。

设  $a_i$  为  $x_{2i}$  的人流是否受  $V_2$  上一场活动流出游客的影响，1 为有影响，0 为无影响，

$T_2 - t_2 > t_{2i}$  表明上一场流出的前往该路线的游客还在路线上，对该路线流率产生影响  
 $T_2 - t_2 < t_{2i}$  表明上一场流出的前往该路线得游客已经走出该路线，对该路线流率不产生影响。

$$a_i = \begin{cases} 1 & i \in \{i \in [1, n] | T_2 - t_2 > t_{2i}\} \\ 0 & i \in \{i \in [1, n] | T_2 - t_2 < t_{2i}\} \end{cases}$$

从  $V_2$  流向各方向的流率可计算为

$$= \frac{x_{2m} - a_i c_2 \frac{x_{2m}}{\sum_{s=1}^n x_{2s}}}{t_{2m}} = \mu_{2m} - \frac{a_i c_2 x_{2m}}{t_{2m} \sum_{s=1}^n x_{2s}}$$

所以，在  $t_2^a$  时间内，所有流出的游客为  $\sum_{m=1}^n \mu_{m2} t_2^a + \left( \left\lfloor \frac{t_2^a - t_2}{T_2 + t t_2} \right\rfloor + 1 \right) C_2$ 。该情况下， $t_2^a$  时  $V_2$  的排队人数为：

$$n_h^a = n_j + \sum_{m=1}^n \mu_{2m} t_2^a - \sum_{m=1}^n \mu_{m2} t_2^a - \sum_{m=1}^n \frac{a_i c_2 x_{2m} t_i^a}{t_{2m} \sum_{s=1}^n x_{2s}} - \left( \left\lfloor \frac{t_2^a - t_2}{T_2 + t t_2} \right\rfloor + 1 \right) * C_2$$

同样，对于  $V_2$  以外的  $V_i$ ， $V_2$  流向其的流率也需要另外计算

$$= \begin{cases} \mu_{2i} & T_2 - t_2 \geq t_{2i} \\ \mu_{2i} - \frac{c_2 x_{2i}}{t_{2i} \sum_{s=1}^n x_{2s}} & T_2 - t_2 < t_{2i} \end{cases}$$

所以，在  $t_2^a$  时间内，所有流出的游客为  $\sum_{m=1}^n \mu_{2i} t_2^a - \left( \left\lfloor \frac{t_2^a - t_2}{T_2} \right\rfloor + 1 \right) C_2$ 。 $t_2^a$  时  $V_2$  的排队人数为：

$$n_i^a = n_i - \sum_{m=1}^n \mu_{im} t_i^a + \sum_{m=1}^n \mu_{mi} t_i^a + \sum_{m=1}^{n, T_2 - t_2 < x_{2i}} \frac{c_2 x_{2i} t_i^a}{t_{2i} \sum_{s=1}^n x_{2s}} + \left( \left\lfloor \frac{t_2^a - t_2}{T_2 + t t_2} \right\rfloor + 1 \right) \frac{x_{2i}}{\sum_{s=1}^n x_{2s}} C_2$$

#### 7.2.4 最终模型

最终模型可表达为

$$\min f = \sum_{i=1, j=1}^{i \leq n, j \leq n} k_{ij} y_{ij} + \sum_{i=1}^n t_i^w$$

$$\sum_{i=1}^n y_{ij} = 1, \quad j = 1 \dots n$$

$$\sum_{j=1}^n y_{ij} = 1, \quad i = 1 \dots n$$

$$y_{ij} \in \{0,1\}$$

符号说明,  $k_{ij}$ 表示游客从 $V_i$ 到 $V_j$ 的路程消耗时间 $t_{ij}$ , 若 $V_i$ 与 $V_j$ 并不直接连通, 则 $k_{ij}$ 为两点间最短距离, 若  $i = j$ , 则有 $k_{ij} = M$ ;  $y_{ij}$ 为 0-1 决策变量, 取值 1 表示游客线路中包括路线, 取值 0 表示旅行线路中不包括路线;  $t_i^w$ 为游客在 $V_i$ 点排队等待时间。  
对于已经游览过一些景点的游客, 可增设限制条件, 添加  $S$  为游客已经游览过点的路线集合, 则可增设约束

$$y_{ij} = 1 \quad (i,j) \in S$$

### 7.3 算例求解

本算例使用经典的 Dijkstra 算法进行计算。

用 MATLAB 生成随机数的方法, 产生当前的状态, 从而减少人为给出算例的主观性。

$$t_0 \text{ 时刻, 路径上的流量矩阵 } Q = \begin{pmatrix} 0 & 50 & 22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 77 & 0 & 80 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 42 \\ 92 & 47 & 0 & 72 & 0 & 35 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 & 29 & 39 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 39 & 0 & 59 & 0 & 0 & 0 & 0 \\ 0 & 6 & 58 & 69 & 0 & 0 & 0 & 78 & 0 & 45 & 0 \\ 0 & 0 & 0 & 0 & 44 & 0 & 0 & 77 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 91 & 33 & 0 & 58 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 60 & 0 & 23 & 0 \\ 0 & 0 & 0 & 0 & 0 & 22 & 0 & 0 & 25 & 0 & 28 \\ 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 26 & 0 \end{pmatrix};$$

当前距离下场开始的时间 (出入口处的时间设为一个很大的值, 这里取为 10000)

$$t = (10000 \ 1668 \ 66 \ 26 \ 109 \ 171 \ 166 \ 88 \ 85 \ 47 \ 61)^T;$$

每个点的容纳量

$$C = (0 \ 400 \ 30 \ 50 \ 30 \ 100 \ 50 \ 30 \ 30 \ 20 \ 50)^T;$$

每个项目点每场持续时间

$$T = [10000 \ 1980 \ 75 \ 150 \ 150 \ 300 \ 150 \ 120 \ 90 \ 90 \ 120]^T;$$

每个项目点相邻两场之间的换场时间

$$tt = (0 \ 60 \ 16 \ 21 \ 16 \ 30 \ 21 \ 16 \ 16 \ 13 \ 21)^T;$$

各点当前的等待人数向量

$$N = [51 \ 530 \ 80 \ 120 \ 116 \ 178 \ 225 \ 172 \ 105 \ 124 \ 179]^T;$$

各项目点间的路径权值矩阵为

$$k = \begin{pmatrix} 0 & 300 & 400 & 700 & 1150 & 650 & 1650 & 1200 & 1300 & 850 & 550 \\ 300 & 0 & 300 & 600 & 1050 & 350 & 1550 & 900 & 1050 & 600 & 250 \\ 400 & 300 & 0 & 300 & 750 & 350 & 1250 & 900 & 1250 & 800 & 550 \\ 700 & 600 & 300 & 0 & 450 & 500 & 950 & 1050 & 1400 & 950 & 850 \\ 1150 & 1050 & 750 & 450 & 0 & 950 & 500 & 1150 & 1550 & 1400 & 1300 \\ 650 & 350 & 350 & 500 & 950 & 0 & 1150 & 550 & 900 & 450 & 600 \\ 1650 & 1550 & 1250 & 950 & 500 & 1150 & 0 & 650 & 1050 & 1500 & 1800 \\ 1200 & 900 & 900 & 1050 & 1150 & 550 & 650 & 0 & 400 & 850 & 1150 \\ 1300 & 1050 & 1250 & 1400 & 1550 & 900 & 1050 & 400 & 0 & 450 & 800 \\ 850 & 600 & 800 & 950 & 1400 & 450 & 1500 & 850 & 450 & 0 & 350 \\ 550 & 250 & 550 & 850 & 1300 & 1800 & 1150 & 800 & 800 & 350 & 0 \end{pmatrix}$$

由 $V_2$ 前往其他项目点的分布

$$(0.2702 \ 0 \ 0.2807 \ 0 \ 0 \ 0.2316 \ 0 \ 0 \ 0 \ 0.2175)^T$$

由 $V_i$ 通向 $V_j$ 的流率

$$\mu = \begin{pmatrix} 0 & 0.3033 & 0.0550 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2567 & 0 & 0.2667 & 0 & 0 & 0.1886 & 0 & 0 & 0 & 0 & 0.2480 \\ 0 & 0.1567 & 0 & 0 & 0.3067 & 0.1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2000 & 0 & 0.0009 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1180 & 0 & 0 & 0 & 0 \\ 0 & 0.2171 & 0.1657 & 0.1380 & 0 & 0 & 0 & 0.1782 & 0 & 0.1667 & 0 \\ 0 & 0 & 0 & 0 & 0.1680 & 0 & 0 & 0.1185 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1655 & 0.0508 & 0 & 0.1450 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1500 & 0 & 0.0511 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0489 & 0 & 0 & 0.0556 & 0 & 0.0800 \\ 0 & 0.1440 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0743 & 0 \end{pmatrix}$$

使用经典的 Dijkstra 算法进行计算。首先,  $k=1$  时,

$V_j \backslash K$	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$	$V_7$	$V_8$	$V_9$	$V_{10}$	$V_{11}$
1	0*	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

表 4.  $k=1$  时 Dijkstra 算法计算图

$K=2$  时, 自入口到各个点, 所需时间为:

$$[+\infty \ 300 \ 400 \ 700 \ 1150 \ 650 \ 1650 \ 1200 \ 1300 \ 850 \ 550]$$

则到达各个项目点时, 各个项目前排队的人数分别为:

$$[+\infty \ 540 \ 64 \ 145 \ 156 \ 0 \ 163 \ 231 \ 104 \ 159 \ 195]$$

自入口到各个点时, 该游客进入的场次分别为:

$$[+\infty \ 2 \ 3 \ 3 \ 6 \ 1 \ 4 \ 8 \ 4 \ 8 \ 4]$$

该游客到达下个项目开始游玩的时间减去入口处初始时间为:

$V_j \backslash K$	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$	$V_7$	$V_8$	$V_9$	$V_{10}$	$V_{11}$
1	0*	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
		3708	612*	1052	2101	831	2218	2264	1675	1592	1048

表 5.  $k=2$  时 Dijkstra 算法计算图

重复以上步骤，最终得到：

$\begin{matrix} V_j \\ K \end{matrix}$	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$	$V_7$	$V_8$	$V_9$	$V_{10}$	$V_{11}$
1	0*	$+\infty$ 3708 3708 3708 5748 7788 9828 13908 15948 20028*	$+\infty$ 621*	$+\infty$ 1052 1565 2420*	$+\infty$ 2101 2267 3263 4425 6085 9239 11895 14717*	$+\infty$ 831 1161*	$+\infty$ 2218 2389 2731 2928*	$+\infty$ 2264 2672 2944 5392 6616 8112 11648 14912 20216 26608*	$+\infty$ 1675 2205 2523 4537 5491*	$+\infty$ 1592 2416 2622 5094 7360 7978*	$+\infty$ 1048 1753 2599 4432 7393 8098 10354*

表 6. Dijkstra 算法计算图

所以该模型算得的结果的游览顺序为：

$$V_1 V_3 V_6 V_4 V_7 V_9 V_{10} V_{11} V_5 V_2 V_8$$

即为：

$$\text{入口} \rightarrow B \rightarrow E \rightarrow C \rightarrow F \rightarrow H \rightarrow I \rightarrow J \rightarrow K \rightarrow D \rightarrow A \rightarrow G \rightarrow \text{出口}$$

## 八、模型四：人流超阈值情况下的紧急疏散模型<sup>[5]</sup>

### 8.1 模型假设

1. 仅考虑单个项目点人流量超阈值的情况，不考虑多处流量均爆发的情况，其它项目点处于正常服务水平。
2. 假设除 A 点以外其他点散场后出来的游客和与其相连点散场游客汇入量大致相等。

### 8.2 模型建立

由外滩踩踏事件等带给我们的教训和反思，游乐园应当设置完善的预警系统。在各个项目点都应设置人流量的警戒线，一旦超过则极易发生踩踏事故。由于出入口是一个流动性场所人们不会在此停留，故不考虑出入口的警戒人数。设各项目点的人流量警戒量为

$\xi = (0 \quad \xi_2 \quad \cdots \quad \xi_{11})^T$ 。项目点可维持正常服务水平的人流量为  $0.9\xi$ 。因此若某项目点

$V_k$  的人数  $N_k > \xi_k$ ，则必须实行人员的调度转移以避免危险的发生，此时目标函数就从

排队等待时间最小转变为人员疏散时间最小。此时项目点  $V_k$  和与其相连的项目点  $V_j$  之间的路径上只允许单向的从  $k$  向  $j$  人转移，而不允许人再去往扩散源点。因此对于与扩散点相连的点来说，若其他点仍按原来的流量向其汇入，而其缺少了一条可疏散的向事故点的通路，同时事故点的游客又会向其转移，这样极易造成  $V_j$  的人流激增甚至突破警戒

线。所以紧急情况发生时应当是整个园区的各项项目点联动控制，使人员尽快疏散且各项目点保持在正常服务水平。



与模型一类似，但是项目点  $V_k$  有  $(N_k - 0.9\xi_k)$  是必须转移量，而其接收其他点的转移人群能力为 0。

可以假定，对于除 A 点之外的项目点，散场后的离开游客  $C_j$  和其他各点汇入其的人数大致相等。

A 点则不然，若拥挤发生于与 A 点相邻的点，而 A 点刚好结束，则从 A 点散去的人  $C_2$  将失去到项目点  $k$  的通路，假定原先选择此通路的人的概率为  $p_{2k}$ ，人数为  $C_2 * p_{2k}$ ，

假定这部分人将平均地去往与 A 相连的其他点，则  $N'_j = N_j + \frac{1}{2} p_{2j} * C_2$ 。而对于 A 点自身来说，由于其他点持续时间短，将时间离散化，可以看做其他点出来的游客近似均匀地来到 A 点，使 A 得排队人数增加，当 A 散场时又有大量人群离开，可以假定

$C_2 = \int_{T_2} \lambda_{j2} dt$ ，即 A 点持续时间内其他点出来的游客累积到达数与 A 点散场时散去的人数

相等，因此 A 点本身的排队人数  $N_2$  不变。若 A 点场次持续进行，则假设扩散的短时间内其他点出来的人流入量很少可忽略。

对于待疏散的项目点  $k$ ，若正好有场次结束，则其他点结束人群不可汇入，而  $k$  点可去往其他点，但  $k$  点等待人群依旧会进去相等的人数，一次不必考虑散场的人  $C_k$  对邻近点的影响。

与模型一同理，实行一次调度生效的时间间隔为  $t_d = 650s$ 。

定义超警戒点为  $V_k$ ，与其邻近点集为  $J$ 。

目标函数是使得疏散过程最短，且较远处人员流动较少，二者可以统一为总的疏散过程花费时间最少，即

$$\min \sum x_{ij} t_{ij}$$

首先设置 0-1 变量  $z_1$

$$z_1 = \begin{cases} 1 & t_2 < t_d \\ 0 & else \end{cases}$$

分以下两种情况建立模型：

1. 人流过大情况发生于 A 点邻近点

$$\begin{aligned}
& \min x_{ij}t_{ij} \\
& s.t. \sum_{j=1}^{11} y_{kj}x_{kj} = N_k - 0.9\xi_k \\
& N_j + \frac{1}{2}z_1p_{2j}y_{2j}C_2 + \sum_i x_{ij}y_{ij} - \sum_{i \in V - \{k\}} x_{ji}y_{ji} \leq 0.9\xi_j \\
& x_{ij} \geq 0 (i=1,2,\dots,11, j=1,2,\dots,11) \\
& y_{ij} (i=1,2,\dots,11, j=1,2,\dots,11) \text{ 为0-1变量, } z_1 \text{ 为0-1变量}
\end{aligned}$$

式（1）表示从密集地  $k$  向各地转移的人数总和与待转移等待人数相等。

式（2）表示其他点  $j$  流入流出人后人员仍在正常范围内。

2. 人流过大情况不发生于 A 点邻近点

$$\begin{aligned}
& \min x_{ij}t_{ij} \\
& s.t. \sum_{j=1}^{11} y_{kj}x_{kj} = N_k - 0.9\xi_k \\
& N_j + \sum_i x_{ij}y_{ij} - \sum_{i \in V - \{k\}} x_{ji}y_{ji} \leq 0.9\xi_j \\
& x_{ij} \geq 0 (i=1,2,\dots,11, j=1,2,\dots,11) \\
& y_{ij} (i=1,2,\dots,11, j=1,2,\dots,11) \text{ 为0-1变量}
\end{aligned}$$

式（1）表示从密集地  $k$  向各地转移的人数总和与待转移等待人数相等。

式（2）表示其他项目点  $j$  流入流出人后人员仍在正常范围内。

### 8.3 算例求解

1. 人流过大情况发生于 A 点邻近点

此时又有 A 点是否演出完毕两种情况，仅以刚好有场次结束为例。

假定将紧急情况的发生点设在 E 点。

各点的警戒阈值为  $\xi=[0 \ 1200 \ 200 \ 300 \ 200 \ 500 \ 300 \ 200 \ 200 \ 150 \ 300]^T$  ；

密集地  $E$  点的等待人数  $N_6=600$ ，需要转移人群。

各点间的距离矩阵为

$$D = \begin{pmatrix} 0 & 300 & 400 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 300 & 0 & 300 & 0 & 0 & 350 & 0 & 0 & 0 & 0 & 250 \\ 400 & 300 & 0 & 300 & 0 & 350 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 300 & 0 & 450 & 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 450 & 0 & 0 & 500 & 0 & 0 & 0 & 0 \\ 0 & 350 & 350 & 500 & 0 & 0 & 0 & 550 & 0 & 450 & 0 \\ 0 & 0 & 0 & 0 & 500 & 0 & 0 & 650 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 550 & 650 & 0 & 400 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 400 & 0 & 450 & 0 \\ 0 & 0 & 0 & 0 & 0 & 450 & 0 & 0 & 450 & 0 & 350 \\ 0 & 250 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 350 & 0 \end{pmatrix};$$

每点当前等待人数向量为

$$N = (96 \ 904 \ 90 \ 198 \ 135 \ 600 \ 266 \ 96 \ 177 \ 131 \ 219)^T;$$

从 A 点散场游客概率选择向量可与模型一中算例保持一致,

$$p_{2\bullet} = (0.6038 \ 0 \ 0.1026 \ 0 \ 0 \ 0.0652 \ 0 \ 0 \ 0 \ 0 \ 0.2285);$$

将决策变量整合为一个列向量  $XX = [X(:,1); X(:,2); \dots; X(:,j); \dots X(:,11)]$  ;

将目标函数的系数矩阵变为一行向量  $TT = [D(1,:), \dots, D(i,:), \dots D(11,:)]$  ;

由此目标函数为  $\min TT * XX$  。

把约束条件化为  $\begin{cases} A * XX \leq b \\ Aeq * XX = beq \\ XX \geq lb \end{cases}$  的形式, 其中  $b = 0.9 * \xi - N - 0.5 * 400 * p_{2\bullet}$ 。

利用 MATLAB 求解, 由于概率向量为小数, 具体人员调度方案只采用普通线性规划问题求解再取整的方法, 满足这类粗略的估计精度。

$$\begin{aligned} x_{21} &= 68, x_{31} = 64, x_{12} = 110, x_{22} = 287, x_{32} = 79, x_{62} = 40, x_{11,2} = 85, x_{13} = 49, x_{23} = 72, \\ x_{43} &= 86, x_{63} = 48, x_{34} = 65, x_{54} = 72, x_{64} = 31, x_{45} = 56, x_{75} = 67, x_{57} = 54, x_{87} = 44, \\ x_{68} &= 28, x_{78} = 62, x_{98} = 64, x_{89} = 70, x_{10,9} = 48, x_{6,10} = 44, x_{9,10} = 103, x_{11,10} = 106, \\ x_{2,11} &= 75, x_{10,11} = 59 \end{aligned}$$

2. 人流过大情况不发生于 A 点邻近点

假定将紧急情况的发生点设在 I 点。

各点的警戒阈值为  $\xi = [0 \ 1200 \ 200 \ 300 \ 200 \ 500 \ 300 \ 200 \ 200 \ 150 \ 300]^T$  ;

密集地 I 点的等待人数  $N_{10} = 300$  , 需要转移。

每点当前等待人数向量为

$$N = (96 \ 904 \ 90 \ 198 \ 135 \ 446 \ 266 \ 96 \ 177 \ 300 \ 219)^T;$$

将决策变量整合为一个列向量  $XX = [X(:,1); X(:,2); \dots; X(:,j); \dots X(:,11)]$  ;

将目标函数的系数矩阵变为一行向量  $TT = [D(1,:), \dots, D(i,:), \dots, D(11,:)]$ ;

由此目标函数为  $\min TT * XX$ 。

把约束条件化为  $\begin{cases} A * XX \leq b \\ Aeq * XX = beq \\ XX \geq lb \end{cases}$  的形式, 其中  $b = 0.9 * \xi - N$

利用 MATLAB 求解并取整后得到;

$$x_{22} = 1063, x_{62} = 12, x_{11,2} = 157, x_{63} = 151, x_{10,9} = 3, x_{10,11} = 162$$

的调配方案, 其中较小的数值可免去调度。

## 九、模型五、六：基于支持向量机的房间预定量模型

### 9.1 模型假设

- 1、附件中做给的数据能准确描述酒店预定量的波动规律
- 2、在其他条件暂无变化的情况下, 酒店每天的预定量或者实际需求量以年为周期呈现一定的规律性

### 9.2 数据预处理

#### ■ 原始数据的分类汇总

每天的订单随机产生, 随着样本容量的增大, 最终每天总预定房间数量呈现一定的数学统计规律。将每日单个的订单按预定日期累加求得日房间预定数量, 将每日到店的游客房间需求量按到店日期累加求得日房间实际需求量, 以便分析房间预定量随二者的变动趋势。

#### ■ 影响因子的定量化

为分析季节、周末、法定假日、暑假等定性化因素对预定房间数量的影响, 对数据进行定量化处理。

$$Si = \begin{cases} 1 & (\text{日期在春季}) \\ 2 & (\text{日期在夏季}) \\ 3 & (\text{日期在秋季}) \\ 4 & (\text{日期在冬季}) \end{cases}$$

$$Wi = \begin{cases} 0 & (\text{日期是工作日}) \\ 1 & (\text{日期是周末}) \end{cases}$$

$$Vi = \begin{cases} 0 & (\text{日期在暑假}) \\ 1 & (\text{日期不在暑假}) \end{cases}$$

其中, 2015 年的假日中, 对预定量有明显影响的有: 4 月 4 日至 6 日清明节放假、5 月 1 日至 5 月 3 日劳动节放假、6 月 1 号儿童节预定量也有显著增加、6 月 20 日至 22 日端午节放假、9 月 26 日至 9 月 28 日中秋节放假、9 月 29 日至 10 月 3 日国庆节, 这样与中秋国庆相连共放假 8 天。因为这些节假日往往持续影响前后期一段时间的预定量, 所以将节假日当天及附近受影响明显的日期均予以标注。

预订日期	预订日期数	房间订量	工作日	法定假日	暑假	季节	订单数量
2015-04-02	92	14	0	0	0	2	7
2015-04-03	93	667	0	0	0	2	132
2015-04-04	94	62	1	1	0	2	8
2015-09-17	260	88	0	0	0	4	20
2015-09-18	261	223	0	0	0	4	27
2015-09-19	262	65	1	0	0	4	13

表 7. 法定假日-预定量影响示例

其中，4月3号是清明节假期前一天，预定量达到高峰；9月18号是中秋节放假前夕，也会导致预定量大幅增长。

到店日期	到店日期数	房间订量	工作日	法定假日	暑假	季节	订单数量
2015-06-22	173	64	0	1	0	3	16
2015-06-23	174	179	0	0	0	3	53
2015-06-24	175	37	0	0	0	3	18
2015-09-20	263	114	1	0	0	4	18
2015-09-21	264	199	0	0	0	4	19
2015-09-22	265	109	0	0	0	4	35

表 8. 法定假日-到客量影响示例

其中，6月23号是端午节假期稍后，9月21号是中秋节前夕，都会使游乐园客流量增加，相应增加了酒店房间预定量。这些影响因素在模型中要予以体现。

$$H_i = \begin{cases} 0 & (\text{日期不受法定假日影响}) \\ 1 & (\text{日期受元旦节影响}) \\ 2 & (\text{日期受清明节影响}) \\ 3 & (\text{日期受劳动节影响}) \\ 4 & (\text{日期受儿童节影响}) \\ 5 & (\text{日期受端午节影响}) \\ 6 & (\text{日期受中秋节影响}) \\ 7 & (\text{日期受国庆节影响}) \end{cases}$$

#### ■ 添加不确定系数

根据一般条件下酒店的运营情况，在同一季节内，无影响因子作用时，酒店预定量每天的波动不会很大，表现为酒店第*i*天的预定量与第(*i*-1)天和第(*i*-2)天的预定量差值存在阈值。观察整个数据发现有一些异常数据，如下图标示出的几条数据，这些未知因素的影响原因不可得知，但是在预测时这些不确定因素的出现不容忽视。

预订日期	预订日期数	房间订量	工作日	法定假日	暑假	季节	订单数量
2015-05-20	140	76	0	0	0	2	23
2015-05-21	141	208	0	0	0	2	31
2015-05-22	142	122	0	0	0	2	33

表 9. 预定量波动示例

到店日期	到店日期数	房间订量	工作日	法定假日	暑假	季节	订单数量
2015-11-09	313	132	0	0	0	4	24
2015-11-10	314	164	0	0	0	4	46
2015-11-11	315	67	0	0	0	4	23

表 10. 到客量波动示例

我们引入不确定系数的概念，对于此类突变变量进行标示，定义不确定系数 $\delta$ ( $\delta = 0、1、2、3 \dots$ )表示房间数量的突变程度，房间数量增加量在一定的变化范围， $\delta$ 对应一个数值。

■ 按季节性因素分类

游乐场可以根据吸引游客的数量分为淡季、旺季、介于淡季和旺季之间的普通季节。由预定房间数量随预定日期及到店日期变化的折线图可以看出，游乐场内的酒店房间预定数量也有同样的变化规律。

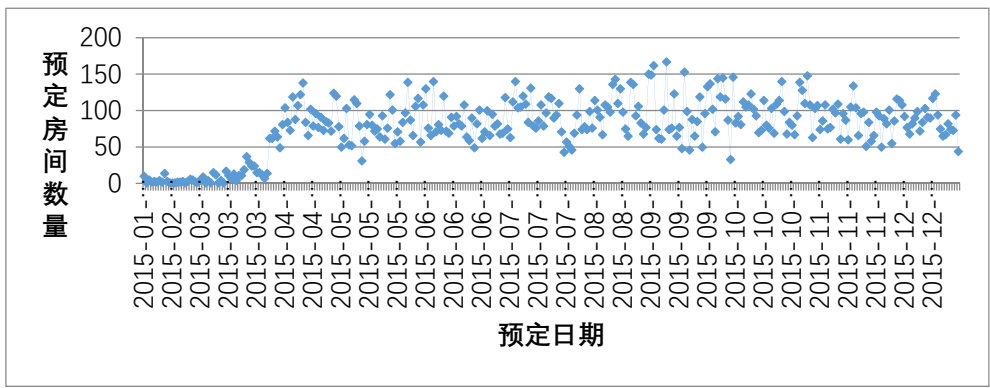


图 11. 预定房间数与预定日期折线图

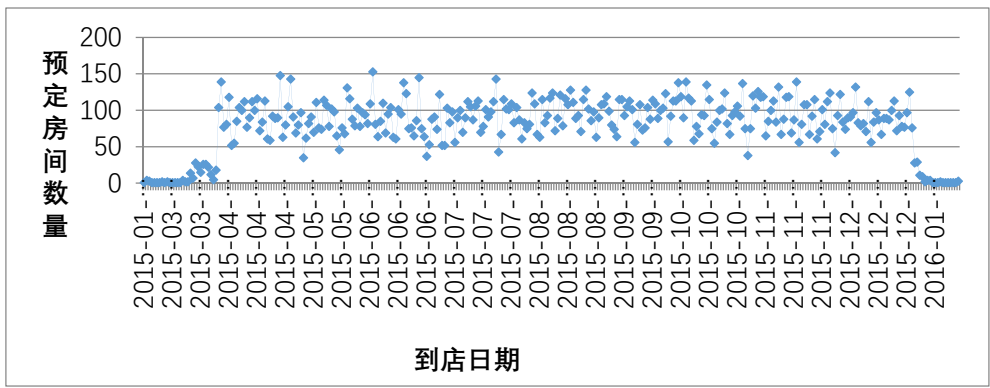


图 12. 预定房间数量与到店日期折线图

9.3 模型五：基于预定日期的支持向量机模型

9.3.1 模型按淡旺季分段

首先，根据预定日期把酒店房间预定量分为淡季、旺季、普通季分段建立模型。



图 12. 淡旺季分类时间轴图

图中，红色直线段表示旺季分布，黄色段表示淡旺季之间的过渡段，蓝色表示订单量较少的淡季。其中，每年的1月1号至3月10号为淡季，3月11号至4月2号为淡季和旺季之间的过渡期，4月3号至12月31号为旺季。

对于每一类模型，分别以预订房间数、季节、是否为周末、法定假日、暑假以及不

确定因子作为输入变量，建立支持向量机模型。分析各个因子对预订数量的影响，工作日和周末预订房间数量大致相同，因此在支持向量机的输入变量中减去工作日/周末的因子。

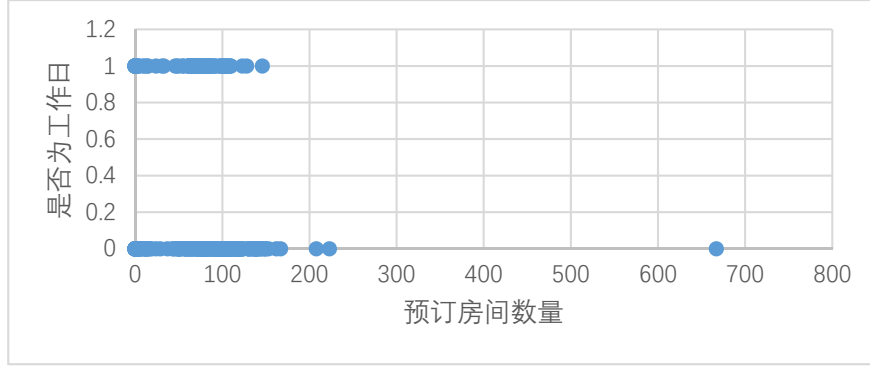


图 13. 工作日/周末-预定量影响图

### 9.3.2 模型形式的确定

到店日期为第  $(t-1)$  年第  $i$  天的需求房间数量为：

$$N_i^{(t-1)} = \sum_{j=1}^n X_{ij}^{(t-1)} \quad (i = 1, 2, \dots, 365)$$

到店日期为第  $t$  年第  $i$  天预定的需求房间数量为：

$$N_i^{(t)} = f(N_i^{(t-1)}) + \Delta(W_i^{(t)}, H_i^{(t)}, S_i^{(t)})$$

其中， $X_{ij}^{(t-1)}$  表示到店日期为第  $t$  年第  $i$  天的第  $j$  个需求的房间数量， $\Delta(W_i^{(t)}, H_i^{(t)}, S_i^{(t)})$  表示因为周末、法定假日、暑假影响而增加的酒店房间需求量。

### 9.3.3 机器学习确定各因素的影响

周末、法定假日、暑假的影响不能进行简单的线性描述，我们利用机器学习的方法解决该非线性问题。支持向量机的机理是将输入向量映射到一个高位的特征空间，并寻找一个满足分类要求的最优分类超平面，使得该超平面在保证分类精度的同时，能够使超平面两侧的空白区域最大化。

对于学习样本集（训练样本集） $S = \{(x_i, y_i), x_i \in R^n\}$  是非线性的，通过定义适当的核函数  $K(x_i, y_i)$  将输入样本空间非线性变换到另一个高位特征空间，在这个特征空间中构造线性回归函数，使得

$$f(x) = w^T \Phi(x_i) + b$$

变为解决以下的优化问题：

$$\min \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$$

约束条件变为：

$$\begin{cases} w^T \Phi(x_i) + b - y_i \leq \xi_i^* + \varepsilon, i = 1, 2, \dots, l \\ y_i - w^T \Phi(x_i) - b \leq \xi_i^* + \varepsilon, i = 1, 2, \dots, l \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

式子中  $C>0$  为惩罚因子,  $C$  越大表示对误差大的数据惩罚越大。可以利用拉格朗日乘子法来求解该约束最优化问题, 为此构造如下拉格朗日函数:

$$L(w, b, \alpha, \alpha^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) - \sum_{i=1}^l \alpha_i [\varepsilon + \xi_i - y_i + \xi_i^* + w^T \Phi(x_i) + b] - \sum_{i=1}^l \alpha_i^* [\varepsilon + \xi_i - y_i + \xi_i^* + w^T \Phi(x_i) + b] - \sum_{i=1}^l (\beta_i \xi_i + \beta_i^* \xi_i^*)$$

根据 KKT 条件, 将  $L$  分别对  $w$ 、 $b$ 、 $\xi_i$ 、 $\xi_i^*$  求偏导数并令其为零, 得到:

$$\begin{cases} w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi(X_i) \\ \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ C - \alpha_i - \beta_i = 0 \\ C - \alpha_i^* - \beta_i^* = 0 \\ i = 1, 2, \dots, l \end{cases}$$

将上述方程组代入公式, 得到对偶最优化问题, 进一步可求得非线性回归函数。这样就确定了酒店房间预定量随各影响因素的变化关系。

#### 9.4 模型六：基于到店日期预测的支持向量机组合模型

##### 9.4.1 模型按淡旺季分段

首先, 根据到店日期把酒店房间每天的房间到客量分为淡季、旺季、普通季分段建立模型。



图 14. 淡旺季分类时间轴图

图中, 红色直线段表示旺季分布, 黄色段表示淡旺季之间的过渡段, 蓝色表示订单量较少的淡季。其中, 每年的 1 月 2 号至 1 月 5 号、3 月 23 日至 4 月 2 号为淡季和旺季之间的过渡期, 1 月 6 号至 3 月 22 号为淡季, 4 月 3 号至 12 月 31 号以及 1 月 1 号为旺季。

##### 9.4.2 支持向量机确立各参数影响

支持向量机原理与模型五中相同, 我们选取季节、是否为工作日、法定假日、是否在暑假为输入变量, 房间日到客数量为输出变量, 训练集和检测集的比例为 3:1, 选取遗传算法进行参数优化。

##### 9.4.3 预定提前天数概率分布的确定

游客预定房间的提前天数根据个人的心理和习惯等因素相关。要根据酒店每天的需求量预测第  $i$  天的预定量, 计算到店日期与预定日期的差值  $D=T_a-T_0$  并研究其分布。预订提前天



数的描述统计如下：平均值为 9.65 天，方差为 187.17，可见数据偏离程度太大，平均值不能体现出数据的整体分布态势。我们分析预订天数是否呈现一定的分布规律，或者有明显的聚类现象。用 MATLAB 编写代码做出预订提前天数的频率分布直方图：

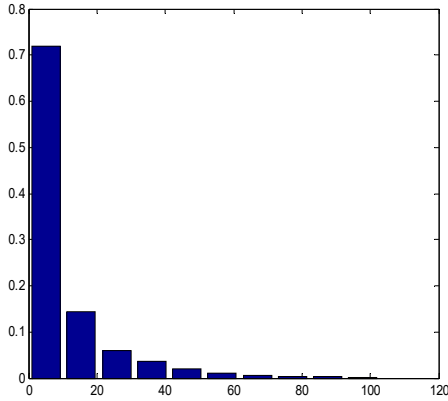


图 15. 预订提前天数频率分布直方图

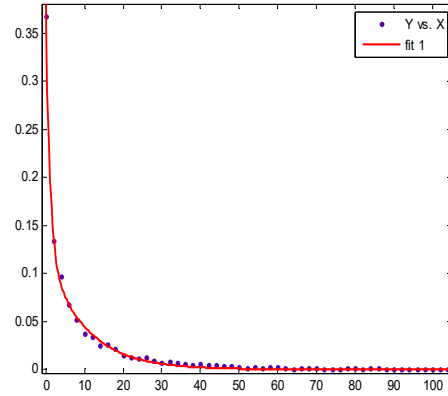


图 16. 预订提前天数概率函数拟合曲线图

提前时间在 1~10 之间的频率达到 0.7 以上，样本数量较大，我们以频率代替概率。用 MATLAB 曲线拟合的方法得出提前预定天数的概率分布函数。

以误差较小的插值函数的形式拟合：

$$f(x) = a \cdot \exp(b \cdot x) + c \cdot \exp(d \cdot x)$$

Coefficients (with 95% confidence bounds 95%致信区间内的拟合常数)：

$$\begin{cases} a = 0.2453 (0.2332, 0.2574) \\ b = -0.9394 (-1.053, -0.826) \\ c = 0.1222 (0.1111, 0.1333) \\ d = -0.1011 (-0.1096, -0.09268) \end{cases}$$

SSE (方差)：0.0003117

R-square (决定系数)：0.998

Adjusted R-square (校正后的决定系数)：0.9979

RMSE (标准差)：0.002522

确定最终拟合函数为：

$$P(d) = 0.2453 \cdot \exp(-0.9394 \cdot x) + 0.1222 \cdot \exp(-0.1011 \cdot x)$$

#### 9.4.4 日预定量的计算

用  $dm$  表示预订提前天数的上限，根据每天的需求量和预订提前相应天数的概率相乘，并进行累加，得到预定量：

$$Q_i^{(t)} = [\sum_{d=0}^{dm} N_{i+d}^{(t)} P(d)] \quad \text{当 } i+dm \leq \text{第 } t \text{ 年的总}$$

天数  $n$  时；

$$Q_i^{(t)} = [\sum_{d=0}^{n-i} N_{i+d}^{(t)} P(d) + \sum_{d=1}^{i+dm-n} N_d^{(t+1)} P(d)] \quad \text{当 } i+dm > \text{第 } t \text{ 年的总天数 } n \text{ 时.}$$

篇幅限制，模型六我们只提供一个思路，这里不再进行求解和预测。

### 9.5 模型的求解和预测

#### 9.5.1 模型五的求解

通过 MATLAB 编写代码建立酒店房间需求量的支持向量机模型，模型需要求解的参数有：

- g gama: 核函数中的 gamma 函数设置 (针对多项式/rbf/sigmoid 核函数)
- c cost: 设置 C-SVC, epsilon-SVR 和 nu-SVC 的参数 (损失函数)

-p epsilon: 设置 epsilon-SVR 中损失函数 epsilon 的值

在具体操作时, 我们选取季节、是否为工作日、法定假日、是否在暑假为输入变量, 房间预订数量为输出变量, 训练集和检测集的比例为 3:1, 选取遗传算法进行参数优化。

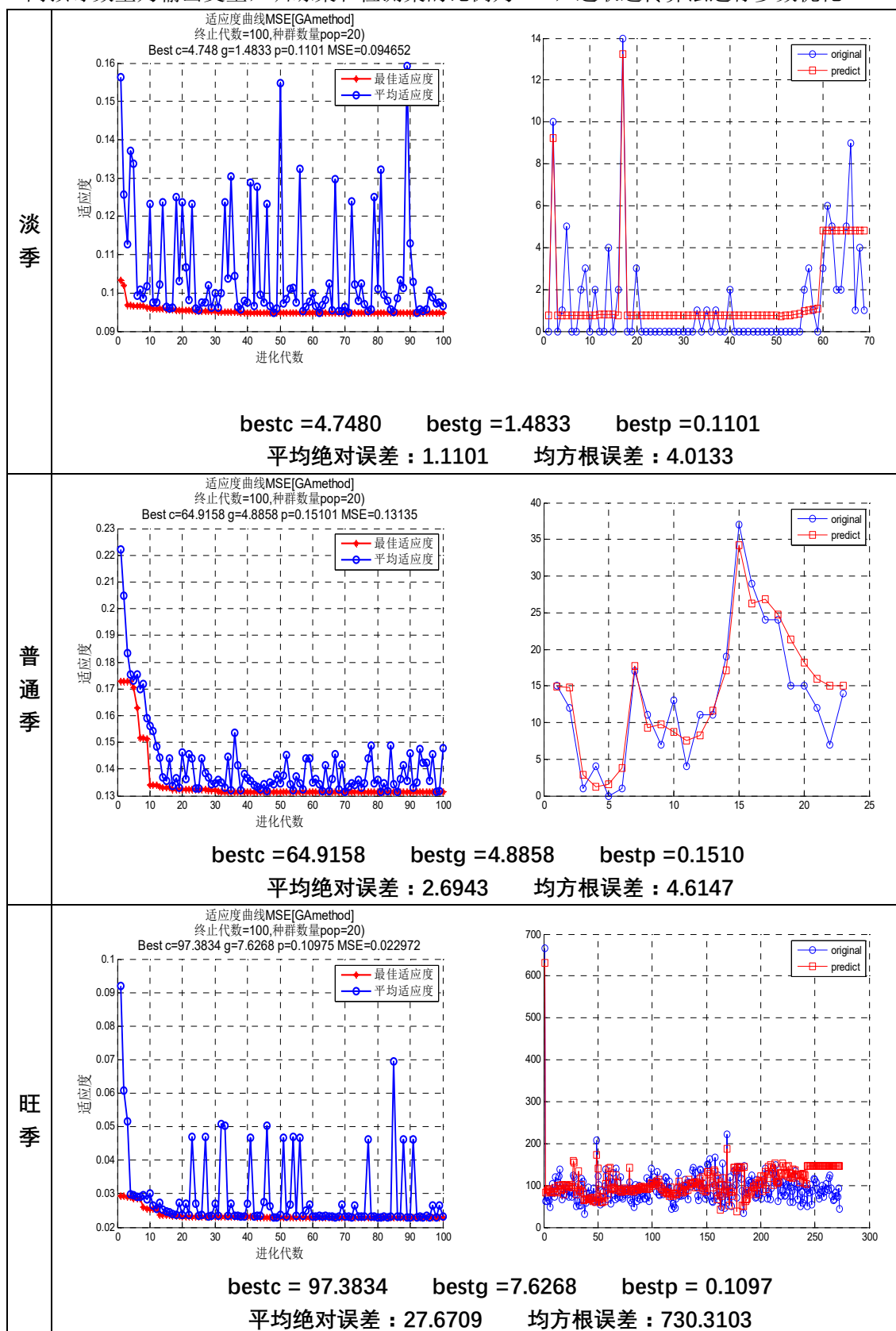


表 11. 模型五参数及误差三段图

### 9.5.2 基于模型五的房间预定量预测

利用求解得到的最优模型对 2016 年 1 月至 3 月份的预定量进行预测，结果如下：

2016/1/1	2016/1/2	2016/1/3	2016/1/4	2016/1/5	2016/1/6	2016/1/7	2016/1/8	2016/1/9	2016/1/10	2016/1/11	2016/1/12
0	2	0	2	5	0	0	2	1	0	1	0
2016/1/13	2016/1/14	2016/1/15	2016/1/16	2016/1/17	2016/1/18	2016/1/19	2016/1/20	2016/1/21	2016/1/22	2016/1/23	2016/1/24
0	4	0	2	5	0	0	3	0	0	1	0
2016/1/25	2016/1/26	2016/1/27	2016/1/28	2016/1/29	2016/1/30	2016/1/31	2016/2/1	2016/2/2	2016/2/3	2016/2/4	2016/2/5
1	0	0	0	0	1	0	1	1	0	1	0
2016/2/6	2016/2/7	2016/2/8	2016/2/9	2016/2/10	2016/2/11	2016/2/12	2016/2/13	2016/2/14	2016/2/15	2016/2/16	2016/2/17
1	0	1	2	0	0	0	1	0	1	0	0
2016/2/18	2016/2/19	2016/2/20	2016/2/21	2016/2/22	2016/2/23	2016/2/24	2016/2/25	2016/2/26	2016/2/27	2016/2/28	2016/2/29
1	1	1	0	1	0	0	1	1	1	0	1
2016/3/1	2016/3/2	2016/3/3	2016/3/4	2016/3/5	2016/3/6	2016/3/7	2016/3/8	2016/3/9	2016/3/10	2016/3/11	2016/3/12
2	4	1	1	3	4	1	2	0	13	10	5
2016/3/13	2016/3/14	2016/3/15	2016/3/16	2016/3/17	2016/3/18	2016/3/19	2016/3/20	2016/3/21	2016/3/22	2016/3/23	2016/3/24
5	5	1	17	11	13	13	5	13	11	19	37
2016/3/25	2016/3/26	2016/3/27	2016/3/28	2016/3/29	2016/3/30	2016/3/31					
13	13	13	13	15	12	7					

## 十一、模型评价

### 优点

本文考虑了多种情况，分别建模，从而形成了一套较为完整的引导方案。分为有游客标识和没有游客标识的情况，并考虑了局部拥挤过度的疏散方案。

模型一在无游客标识的情况下，进行了合理的简化假设，使得随时间动态变化的系统变为线性规划问题，将游乐园所有人在调度完成时的损耗时间之和最小作为目标，便于求解。

模型二确定了景区评定等级指标和信息发布策略。确定了景区划分的指标，具有借鉴意义，对信息发布策略进行了探讨，考虑了游客行为的反馈作用。

模型三根据当前游乐园的状态进行建模，为游客推荐一条最短时间的游玩路线，使游客游园体验提高。

模型四对于某处人流过大的情况考虑了应急预案的生成，整合了人员转移和消耗时间少的目标为消耗的总时间最少，形成线性规划问题。

模型五结合机器学习的支持向量机算法分类建模更精准地把握了变动规律。

模型六考虑每日到客量与各影响因子的关系，结合预定提前天数的概率函数对每日预定量进行计算，是一种崭新的思路。

### 缺点

模型一虽然对何时进行引导的时间上针对的是单个项目点或某个区域，但是由于各项目点之间人员流动后相互影响，因此只能基于总体最优进行建模，可能会牺牲局部人群的利益。没有对单处不平衡小范围内的人员流动进行建模。

模型二由于缺少对影响游客选择行为的数据，无法进行 logit 拟合，只能给出理论上的方法，无法给出具体规律。

模型三只是将当前查询游客的时间最短作为目标函数，并没有从宏观上进行考虑，且根据的是当前的游乐园人流状态，没有考虑人流状态随时间的变化。

模型四假设游玩出来的游客到达项目点与离开项目点的人数大致相等，但实际上是有

影响的。且没有考虑道路的容量限制。

模型五在输入变量的选取和定性化表示方面还有待完善。

模型六所需为每日到客量，仅凭借 2015 年全年预定量不能得知 2015 年全年到客量，所以需要更加完备的数据。

## 推广

在没有游客标识的情况下可以在某个项目点人群等待时间过多时，限制人员的流动在附近的某一范围内，从而减少整个游乐园人员的调动。在拥有数据的情况下可以分析各项目点人群的到达规律人数的变化情况，掌握系统随时间变化的规律，更加合理地建模。

在发布游乐园人流信息后，通过一定时间对人们选择行为的观测，进行数据拟合，可以优化信息发布策略。

针对每个游客推荐时，由于对所有节点构建虚拟弧，虽然在路径距离已经存在惩罚，但依然可能导致很多重复路径，可以在该模型基础上考虑增加对于重复路径的惩罚，尽量避免重复路径的产生。同时，也可以考虑游乐园状态随时间的变化，提高模型的精度。

在拥挤疏散时，考虑人的感受，增加拥挤度指标，设置道路的容量限制，从而实现更加科学的疏散。

在预测房间预定量时，结合支持向量机算法对影响因素深入挖掘，利用遗传算法优化模型参数，可以具体预测到每一天的预定量，且达到较高的准确度。该模型不仅适用于酒店房间，还可对其他受多因素影响的数据进行预测。

## 参考文献

- [1]. Logit模型在个体选择行为中的研究演进\_陈锟.
- [2]. 张春勤, 道路交通动态信息发布策略研究, 2011, 吉林大学. 第 102页.
- [3]. 交通信息对交通行为的影响和信息发布策略研究的一个新模型\_吴文祥.
- [4]. 交通信息发布策略的研究\_尹涛
- [5]. 傅家良, 运筹学方法与模型, 上海市国权路 579 号: 复旦大学出版社, 2014.

## 附录

模型一算例一

1. 计算调配后人数

```
x=sym('x',[11,11]);
M=sym('M',[1,11]);
for i=1:11
    for j=1:11
        if td<t(j)+d(i,j)&td<t(i)
            M(i)=M(i)+Y(i,j)*x(j,i)-Y(i,j)*x(i,j)+Y(i,j)*(Q(j,i)-Q(i,j));
        elseif td<t(j)+d(i,j)&td>t(i)
            M(i)=M(i)+Y(i,j)*x(j,i)-Y(i,j)*x(i,j)-C(i)*(floor((td-
t(i))/(T(i)+tt(i)))+1)+Y(i,j)*(Q(j,i)-Q(i,j));
        elseif td>t(j)+d(i,j) & td<t(i)
            n=floor((td-(d(i,j)+t(j)))/(T(j)+tt(j)+d(i,j)))+1;
            M(i)=M(i)+Y(i,j)*x(j,i)-Y(i,j)*x(i,j)+Y(i,j)*(Q(j,i)-Q(i,j));
            for k=1:n
                M(i)=M(i)+Y(i,j)*C(j)*P(i,j);
            end
        else
            n=floor((td-(d(i,j)+t(j)))/(T(j)+tt(j)+d(i,j)))+1;
            M(i)=M(i)+Y(i,j)*x(j,i)-Y(i,j)*x(i,j)+Y(i,j)*(Q(j,i)-Q(i,j))-
C(i)*(floor((td-t(i))/(T(i)+tt(i)))+1);
            for k=1:n
                M(i)=M(i)+Y(i,j)*C(j)*P(i,j);
            end
        end
    end
end
M=M+N
3. 计算调配完成时的总消耗时间
TT=zeros(1,11);
TT=TT+w.*M;
% for i=1:11
%     for j=1:11
%         TT(i)=TT(i)+0.5*Q(i,j);
%     end
% end
TT
total=sum(TT);
for i=1:11
    for j=1:11
        total=total+0.5*Q(i,j)*d(i,j)+x(i,j)*d(i,j);
    end
end
```

```

end
total
模型三算例 1
将距离矩阵变为行向量
ddd=[d(1,:),d(2,:),d(3,:),d(4,:),d(5,:),d(6,:),d(7,:),d(8,:),d(9,:),d(10,:),d(11,:)];
生成等待人数随机数
Nn=round([50*rand(1,1)+50 480*rand(1,1)+600 100*rand(1,1)+80 100*rand(1,1)+170
100*rand(1,1)+80 100*rand(1,1)+350 100*rand(1,1)+170 100*rand(1,1)+80 100*rand(1,1)+80
100*rand(1,1)+35 100*rand(1,1)+170])
Nn(6)=600
相应系数矩阵和向量
A
A=[0,1,1,zeros(1,8),-1,-1,zeros(1,108);0,-1,zeros(1,9),1,0,1,0,0,1,0,0,0,0,1,0,-
1,zeros(1,32),-1,zeros(1,54),-1,zeros(1,9);0,0,-1,zeros(1,10),-
1,zeros(1,8),1,1,0,1,zeros(1,9),-1,zeros(1,21),-1,zeros(1,63);zeros(1,25),-
1,zeros(1,9),1,0,1,1,zeros(1,8),-1,zeros(1,10),-1,zeros(1,62);zeros(1,37),-
1,zeros(1,9),1,0,0,1,zeros(1,19),-1,zeros(1,50);zeros(1,16),-1,zeros(1,10),-1,zeros(1,10),-
1,zeros(1,17),1,1,1,0,0,0,1,0,1,zeros(1,17),-1,zeros(1,21),-1,zeros(1,16);zeros(1,50),-
1,zeros(1,19),1,0,0,1,zeros(1,9),-1,zeros(1,37);zeros(1,62),-1,zeros(1,10),-
1,zeros(1,8),1,1,0,1,zeros(1,9),-1,zeros(1,25);zeros(1,85),-1,zeros(1,9),1,0,1,zeros(1,9),-
1,zeros(1,13);zeros(1,64),-1,zeros(1,32),-1,zeros(1,21),-1,0;zeros(1,21),-1,zeros(1,87),-
1,0,1,zeros(1,7),1,0]
b
b=0.9*ks-Nn-200*p
Aeq1
Aeq1=[zeros(1,16),1,zeros(1,10),1,zeros(1,10),1,zeros(1,43),1,zeros(1,21),1,zeros(1,16);
zeros(1,62),1,zeros(1,58);zeros(1,58),1,zeros(1,62);zeros(1,57),1,zeros(1,63);zeros(1,56),1,
zeros(1,64);zeros(1,64),1,zeros(1,56)]
beq1
beq1=[150;0;0;0;0;0];
lb
lb=zeros(1,121);
linprog(ddd,A,b,Aeq1,beq1,lb,[])
round(ans)
模型三算例 2
将距离矩阵变为行向量
ddd=[d(1,:),d(2,:),d(3,:),d(4,:),d(5,:),d(6,:),d(7,:),d(8,:),d(9,:),d(10,:),d(11,:)];
生成等待人数随机数
nn=round([50*rand(1,1)+50 480*rand(1,1)+600 100*rand(1,1)+80 100*rand(1,1)+170
100*rand(1,1)+80 100*rand(1,1)+350 100*rand(1,1)+170 100*rand(1,1)+80 100*rand(1,1)+80
100*rand(1,1)+35 100*rand(1,1)+170])
nn(10)=300
相应系数矩阵和向量

```

```

A=[0,1,1,zeros(1,8),-1,-1,zeros(1,108);0,-1,zeros(1,9),1,0,1,0,0,1,0,0,0,0,1,0,-
1,zeros(1,32),-1,zeros(1,54),-1,zeros(1,9);0,0,-1,zeros(1,10),-
1,zeros(1,8),1,1,0,1,zeros(1,9),-1,zeros(1,21),-1,zeros(1,63);zeros(1,25),-
1,zeros(1,9),1,0,1,1,zeros(1,8),-1,zeros(1,10),-1,zeros(1,62);zeros(1,37),-
1,zeros(1,9),1,0,0,1,zeros(1,19),-1,zeros(1,50);zeros(1,16),-1,zeros(1,10),-1,zeros(1,10),-
1,zeros(1,17),1,1,1,0,0,0,1,0,1,zeros(1,17),-1,zeros(1,21),-1,zeros(1,16);zeros(1,50),-
1,zeros(1,19),1,0,0,1,zeros(1,9),-1,zeros(1,37);zeros(1,62),-1,zeros(1,10),-
1,zeros(1,8),1,1,0,1,zeros(1,9),-1,zeros(1,25);zeros(1,85),-1,zeros(1,9),1,0,1,zeros(1,9),-
1,zeros(1,13);zeros(1,64),-1,zeros(1,32),-1,zeros(1,21),-1,0;zeros(1,21),-1,zeros(1,87),-
1,0,1,zeros(1,7),1,0]

```

b

```
b=0.9*ks-nn
```

Aeq

```

Aeq=[zeros(1,64),1,zeros(1,32),1,zeros(1,21),1,0;zeros(1,107),1,zeros(1,13);zeros(1,104)
,1,zeros(1,16);zeros(1,109),1,zeros(1,11)]

```

beq

```
beq=[165;0;0;0]
```

lb

```
lb=zeros(1,121);
```

```
linprog(ddd,A,b,Aeq,beq,lb,[])
```

round(ans)

问题二 MATLAB 代码

```
DATA=xlsread('2016.xlsx','sheet1');
```

```
train_x=DATA(1:105,2:6);
```

```
train_y=DATA(1:105,1);
```

```
test_x=DATA(81:105,2:6);
```

```
test_y=DATA(81:105,1);
```

```
Method_option.plotOriginal = 0;
```

```
Method_option.xscale = 1;
```

```
Method_option.yscale = 1;
```

```
Method_option.plotScale = 0;
```

```
Method_option.pca = 0;
```

```
Method_option.type = 5;
```

```
[predict_Y,mse,r] = SVR(train_y,train_x,test_y,test_x,Method_option);
```

```
[bestCVmse,bestc,bestg,bestp] = ...
```

```
gaSVMcgpForRegress(TrainL,Train,ga_option)
```

```
cmd = ['-c ',num2str(bestc),' -g ',num2str(bestg),' -p ',num2str(bestp),' -s 3'];
```

```
model = svmtrain(train_y,train_x,cmd);
```

```
B=xlsread('2016.xlsx','sheet2');
```

```
pre_x=B(71:196,2:6);
```

```
pre_y=B(71:196,1);
```

```
[pre_qua] = svmpredict(pre_y,pre_x,model);
```

模型三代码

```

Q=[0 50 22 0 0 0 0 0 0 0 0 ;
    77 0 80 0 0 66 0 0 0 0 42;
    92 47 0 72 0 35 0 0 0 0 0 ;
    0 0 60 0 29 39 0 0 0 0 0 ;
    0 0 0 0 39 0 59 0 0 0 0 ;
    0 56 58 69 0 0 0 78 0 45 0;
    0 0 0 0 44 0 0 77 0 0 0;
    0 0 0 0 0 91 33 0 58 0 0;
    0 0 0 0 0 0 0 60 0 23 0;
    0 0 0 0 0 22 0 0 25 0 28;
    0 36 0 0 0 0 0 0 0 26 0 ;];
D=[ 100000      300   400   700   1150   650   1650   1200   1300   850   550;
    300   100000   300   600   1050   350   1550   900   1050   600   250;
    400   300   100000   300   750   350   1250   900   1250   800   550;
    700   600   300   100000   450   500   950   1050   1400   950   850;
    1150   1050   750   450   100000   950   500   1150   1550   1400   1300;
    650   350   350   500   950   100000   1150   550   900   450   600;
    1650   1550   1250   950   500   1150   100000   650   1050   1500   1800;
    1200   900   900   1050   1150   550   650   100000   400   850   1150;
    1300   1050   1250   1400   1550   900   1050   400   100000   450   800;
    850   600   800   950   1400   450   1500   850   450   100000   350;
    550   250   550   850   1300   600   1800   1150   800   350   10000;];
c=[0 400 30 50 30 100 50 30 30 20 50];
t=[10000 1668 66 26 109 171 166 88 85 47 61];
T=[10000 1980 75 150 150 300 150 120 90 90 120];
N=[51 530 80 120 116 178 225 172 105 124 179];
tt=[0 60 16 21 16 30 21 16 16 13 21];
x=Q./D;
A2j=Q(2,:)./sum(Q(2,1:11));
sx(1)=1;
tf(1)=0;
nl=[0 1 1 1 1 1 1 1 1 1 1];
e1=[1 1 1 1 1 1 1 1 1 1 1];
e0=[0 0 0 0 0 0 0 0 0 0 0];
for i=1:11
    if (T(2)-t(2))<D(2,i)
        x(2,i)=x(2,i)-(c(2)*A2j(i)/D(2,i));
    end
end
for n=1:10
    na=e0;
    tfn=e0;
    ta=e0;
    ta=e0;

```



```

    if n==1
        ta=D(sx(n),:);
    elseif (n==2)
        ta=D(sx(n),:)+tf(n);
    else
        ta=D(sx(n),:)+T(sx(n))+tf(n);
    end
    for i=1:11
na(i)=(((sum(x(:,i))-sum(x(i,:))).*ta(i))+N(i));
    if i==2
        for j=1:11
            if x(2,j)==0
                else
                    na(i)=na(i)-(floor((ta(i)-t(2))/(T(2)+tt(2)))+1)*c(2)*A2j(i);
                end
            end
        end
    end
    if na(i)<0
        na(i)=0;
    end
    end
na=na+A2j.*c(2).*(floor((ta-t(2))/(T(2)+tt(2)))+1);
tfn=(floor(na./c)+e1+floor((ta-t)./(T+tt))).*(T+tt)+t
    min=1000000;
    mini=0;
    for i=1:11
        if nl(i)==1
            if tfn(i)<min
                min=tfn(i);
                mini=i;
            end
        end
    end
    end
    sx(n+1)=mini;
    tf(n+1)=min;
    nl(mini)=0;
end
sx

```