



同濟大學  
TONGJI UNIVERSITY

**数学建模大赛  
论文**

参赛成员具体信息				
姓名	学号	学院	专业	联系方式

2016 年 5 月

# 游乐园游客疏导方案及酒店预订量预测模型

## 摘要

按照马斯洛的需求层析理论,当人们满足基本物质需求时就会向更高层次的精神层面进行迈进。近年来,旅游游玩的人数越来越多,导致游玩场所人满为患。

本文基于Youth游乐园的相关布局以及所得的数据,针对客流量较大所导致的人员拥挤问题,建立相关的模型。根据客流情况,及时分流人群,为顾客提供游园线路引导,保障游客得到最佳的游园体验。

问题一, Youth乐园的规划图可以用更为简洁明了的图论来表示,简化为10个项目点以及其中连接各个项目点的连接线,游客可沿着各个项目点连接线往返于各个游乐项目。基于实时调度数据进行实时动态仿真预测,进一步反馈指导调度决策,从游客分流管理角度构建景区管理耗散结构。利用哈密顿模型建立最短路径模型;假设各园区时间独立性,将每个园区视作一个M/M/1/N排队系统,从而得知某时刻各园区的游客分布情况;建立时空负荷率公式并且求出每个园区的时空负荷率,从而在考虑游客满意度尽量在符合最短路径的前提下进行有效分流。

问题二, 根据皇冠假日酒店已有的历史预订数据信息,综合考虑季节,工作日/周末,法定假日,寒假等对房间预定量造成影响的因素,对数据进行分类处理,分析其特征并进行线性拟合分析数据的长期趋势。用对数变换和归一化对数据进行预处理之后,建立基于日期的五次非线性回归模型来分析各种因素对房间预订量的影响机制;考虑到时间序列模型,运用计量经济学软件EViews对时间序列进行二次指数平滑法,Holt-Winters——无季节性模型,以及ARIMA模型建模预测,通过曲线拟合和参数估计建立数学模型,并根据R平方值检验模型拟合程度。最后,引入清明节的节假日指数对非线性回归方程进行修正,一定程度上提高了模型的预测精度,从而对2016年1月至3月每天预定房间数进行更为精确的预测。

**关键词:** 分流与疏导, 哈密顿路径, 排队论, 时空符合率, 预订量预测, 非线性回归, 时间序列模型, 节假日指数

## 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。

试就园区的整体规划，建立数学模型分析研究下面的问题：

1. 附件 1 为 Youth 乐园的规划图，共设 A-J 共 10 个项目点，游客可沿着图中标出的线路往返下个游乐项目。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。每个游乐项目安排请参见表 1。

2. 皇冠假日酒店是游乐园内的酒店，目前已开业，为有需要的游客提供住宿便利。请根据该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(附件 2), 预测 2016 年 1 月至 3 月每天预定房间数。

## 二、 问题分析

### 2.1 问题 1 分析

通过给定的有关 Youth 乐园的各项数据，可以通过数据处理得知一天中在园内各园区的分布情况。

子模型一、根据所给的 Youth 乐园园区规划图利用哈密顿模型建立最短路径模型，确保每位游客都能体验各游乐设施且能达到最优体验效果的前提。

子模型二、在假设各园区时间独立性的前提下，将每个园区视作一个 M/M/1/N 排队系统，从而得知某时刻各园区的游客分布情况。

子模型三、结合子模型一和子模型二的情况下，建立时空负荷率公式并且求出每个园区的时空负荷率，并考虑游客满意度尽量在符合最短路径的前提下进行有效分流。

### 2.2 问题 2 分析

通过对 2015 年酒店日预定量数据处理分析，我们发现影响预订量的主要因素有：长期变动趋势、星期、节假日、寒暑假等，为变量的假设和模型的建立提供依据。由于提供的数据波动性较强且随机性大，可以对数据进行对数变换和归一化预处理，以减弱数据的波动。用传统的线性拟合方法只能反映数据变化的长期趋势，初步考虑使用非线性回归的方法对图像进行拟合，得到拟合程度较高的非线性函数。进一步建立基于时间序列的二次指数平滑模型和 ARIMA 模型进行预测，最后通过设定的节假日虚拟变量对模型进行修正，得到 2016 年一到三月的酒店日预定量预测值。<sup>[4]</sup>

## 三、 模型的基本假设

### 3.1 问题 1 基本假设

1) 按照一般“朝九晚五”的工作习惯，假设 Youth 游乐园一天正常运营的时间为 8 个小时，即 480min。

2) 各园区的输入过程满足平稳性、无后效性和普通性，即可看作一个稳定的

M/M/1/N 排队系统。

3) 假设游客平均走路速度为 0.8m/s, 并使得游客在游乐项目之间通道花费的时间隶属于相邻园区的服务时间。不考虑游客在路上的逗留, 即游客游玩完一个游乐项目就直接步行到下一个游乐项目。

4) 不考虑游乐项目之间的热度不同及价格等因素而导致的游客有意识的选择游乐项目现象。

### 3.2 问题 2 基本假设

- 1) 假设对预订量有影响的节假日只有国家法定节假日（春节、清明节等等）。
- 2) 假设其他因素（如酒店房间价位等）的变动对预订量不造成影响。
- 3) 假设预测时间段内酒店所在地没有重大自然灾害或人为事故。
- 4) 假设日预订量数据真实可靠, 没有较严重的错误和遗漏。

## 四、符号说明

$G = \langle V, E \rangle$ , 有向图或无向图	$V$ , 节点集
$A = (a_{ij})_{n \times m}$ , 点边关联矩阵	$e = \langle u, v \rangle$ , $u$ 和 $v$ 是 $e$ 的端点
$p_n$ , 顾客到达的概率	$N$ , 系统容量
$L$ , 项目 $i$ 在 $t$ 时刻的时空负荷率	$E$ , 边集
$f(i)$ , 由 $V_i$ 点出发至终点 $V_m$ 的最短路程	$S(t_{ij})$ , 游客分流调度在等待时间为 $t_{ij}$ 的满意度
$t$ , 天数 (2015/1/1 设为 1)	$W_t$ , (是否寒假, 是为 0)
$H_i$ , 距清明节第 $i$ 天的节假日指数	$S_i$ , 星期指数 ( $i=1, 2, \dots, 7$ )
$N_t$ , 原始序列剥离星期影响做对数变换	$Y_t$ , 对 $N_t$ 归一化处理
$Y_t$ , 长期趋势项	$I_t$ , 随机项

## 五、游客疏导模型（问题一）

### 5.1 子模型 1-最短路径模型

#### 5.1.1 图论在游乐园游玩路线优化中的运用

图论 (Graph Theory) 是数学的一个分支, 它以图为研究对象。图论中的图是由若干个给定的点及连接两点的线所构成的图形, 这种图形通常用来描述某些事物之间的某种特定关系。其中, 图中的点代表事物, 连接两点的线表示相应的两个事物间具有某种特定的关系。

通常来说, 游客可以分为两种, 一种是游客在进园前就对游乐园有着较多的了解, 并且对于游乐项目及行走路线等相关内容都比较清楚, 这时游客入园后就能够有秩序地并且在时间与空间上较少地占用游乐园的负荷。另外一种情况就是游客在入园前对游乐园一无所知。这时, 由于游乐园内的线路众多, 游客在园区内选择游玩线路也就多种多样, 有很大程度的随机性或者无目的性。结果就会大大增加游客在游乐园区内的逗留时间, 同时对于游玩时间较长的项目游客游玩的完整性也无法保证, 往往出现的情况就是在前几个项目还能兴致勃勃, 满怀热情

地去游玩,但到后来发觉时间不够后其余的项目只能随便看看或者由于没找到地点,干脆不玩。最终,从游乐园管理者的角度来看,游乐园接纳游客的变动成本远低于固定成本,要获得最大的经济效益就需要在尽量不超过园内的基础设施承受能力及游客游兴的基础上园区能够最大限度的容纳最大的客流量。一般来说,园区对游客的最大容纳量与园区自身空间规模、设计及园内基础设施承受能力相关,然而从管理角度出发设计最佳游客流向安排可以极大提高园区对游客的容纳量。若将园区内的游玩线路看作图的边,园区内的项目看作图的结点,保证游客游玩目标的游玩线路就可以转化为图的哈密顿路,即通过求哈密顿路来解决游客随机选择游玩线路问题是实际可行的。<sup>[9]</sup>

#### 5.1.1.1 网络最短路在游乐园游玩线路优化中的运用

对游玩线路的定义,不同学者的看法之间略有差别,但共同点是:游玩者、交通线、游玩点、游玩服务是构成游玩线路不可缺少的要素。在本次研究中,可以认为游玩线路是指为了使游玩者能够以最短的时间获得最好的旅游体验,由游乐园利用交通线和相关服务设施串联若干项目点所形成的具有一定特色的路线。合理可行的游玩线路既可以使游客少走弯路,节约时间,也可以加速园区内游客流的周转率,减少个别项目人满为患,而其他项目无人问津、效益低下的现象,同时还可降低由于游客过多而导致的安全隐患。

本研究中,在园区内两项目间可能存在多条线路可以到达。我们可以认为游客在初次到达园区时对园区的了解情况有限,在游玩过程中不可避免的会走不少弯路、错路。另一方面,从众心里也会从另一方面导致园区内的拥堵程度加大,拉长了游客在园区内的游玩时间,并且多余的游玩时间对于园区和游客都是毫无益处的。网络最短路描述的是园区内任意两项目间距离最短的那条线路。应用网络最短路可以实现从一个项目到其他项目间的最短路线,这在线路优化过程中可以应用到两个节点之间的连接优化中。<sup>[9]</sup>优化后的线路连接对游客在园区内的游玩可起指引作用,对整体线路的优化具有重要意义。

#### 5.1.1.2 最短哈密顿路在园区游玩线路优化中的作用

通常,游客在刚刚达到园区时往往会根据自身对园区的了解及自身喜好任意选择游玩线路。在这种情况下,开始时,对可能要游玩的项目及线路的选择游客会在心中都有一个大致的规划,然而,随着游玩活动的进行,原先的规划会越来越模糊,有时候甚至可能对自己所处的地点都不清楚,也不知接下来应往哪个方向继续游玩,只能随意的选择一条线路或跟着人群,那里人多就往哪里走。这种随意及跟从是导致园区客流拥堵的重要因素。进行客流疏导的主要手段是将园区游玩线路进行优化并将其推荐给游客。网络最短路所做的是实现一个项目到另一个项目之间的连接优化,而不仅仅是两个项目之间的连接优化就可以使一条游玩线路得到优化,而要将游客需要游玩的所有节点进行连接优化。最短哈密顿路描述的是经过所有项目有且仅有一次的最短路线,是在项目组合及任意两项目的连接优化后将所有项目综合后进行的连接优化。利用最短哈密顿路进行游玩线路优化一方面可以从整体上对游客的游向进行管理;另一方面,若将项目数量固定,可以使游客在最短的时间内游玩完园区;若将时间固定,可以使游客在一定的时间内游玩尽量多的项目,可以间接的增大园区的周转率,从而间接地增大园区的容量,实现园区客流疏导的目的。<sup>[9]</sup>

#### 5.1.2 园区游玩图的图论描述

在用图论工具研究园区客流疏导的问题时,首先需要用图论语言描述园区游玩图。在图论中,图分为两大类:有向图和无向图。根据园区内游道的通行方向

及游客游玩的自主性，可以把园区游玩图转化为图论中的无向图和有向图：若游客在园区内所有道路上的游玩是自主的，即园区对游客在游玩线路上的方向是不限制的，则可以将园区游玩图看作一个无向图；若园区内存在只能单方向通行的游道则可以把园区游玩图看作一个有向图。关于无向图和有向图，可以用数学语言定义。

定义 1（无向图） 一个无向图  $G$  是一个有序二元组  $\langle V, E \rangle$ ，记作  $G = \langle V, E \rangle$ ，其中  $V$  是一个非空集合， $V$  中的元素称为节点； $E$  是无序积  $V \times V$  的多重子集（元素可重复出现的集合），称  $E$  为  $G$  的边集， $E$  中的元素称为无向边或简称为边。

定义 2（有向图） 一个有向图  $G$  是一个有序二元组  $\langle V, E \rangle$ ，记作  $G = \langle V, E \rangle$ ，其中  $V$  是一个非空的节点集， $E$  是笛卡尔积  $V \times V$  的多重子集，其元素称为有向边，也简称边或弧。

在一个图  $G = \langle V, E \rangle$  中，为了表示  $V$  和  $E$  分别是图  $G$  的节点集和边集，常将  $V$  记成  $V(G)$ ，也将  $E$  记成  $E(G)$ 。从定义看，只需将边进行区分即可区分无向图和有向图。边  $\langle V_i, V_j \rangle$  在无向图中表示以  $V_i$  和  $V_j$  为节点的边，在有向图

中则表示以  $V_i$  为起点， $V_j$  为终点的有向边。<sup>[9]</sup>

下面将园区游玩图进行图化（如图 5.1），把项目或园区内游道交叉点看作图的节点（其中用实心点表示项目，空心点表示园区内游道交叉点），将游玩道路看作图的边，则该园区游玩图可用无向图  $G = \langle V, E \rangle$  表示，其中  $V = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}\}$ ， $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}\}$ 。

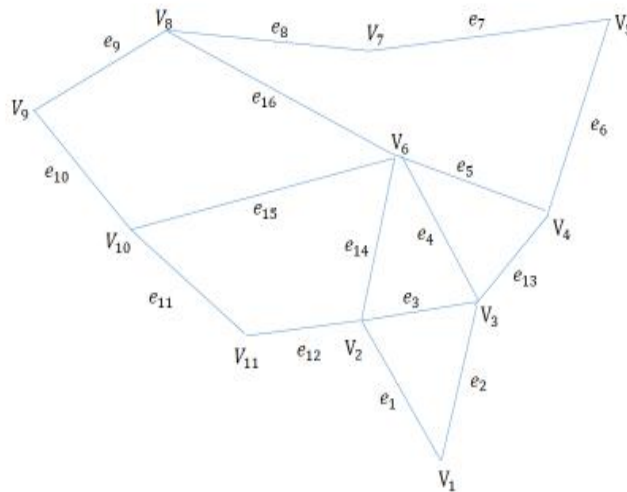


图 5.1 Youth 园区规划路线的图化

图中节点集  $V = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}\}$  只是表示园区内的项目点，利用图论方法可以对游玩线路进行优化：利用网络最短路模型可以优化游玩线路中任意两项目之间的连接路径及利用最短哈密顿路模型优化整条线路的连接路径。其中，利用图论方法对游玩线路进行节点连接优化的步骤为：第一步：对园区内的项目进行组合，将所选择的项目在图中用节点表示，根据园区线路图

标注所选节点之间存在的道路。第二步：寻找连接任意两项目的连接线路。第三步：利用最短哈密顿路求得经过所有选择的项目有且仅有一次的线路即为最终的优化线路。

### 5.1.3 网络最短路建模

#### 5.1.3.1 建立数学模型

在图  $G = \langle V, E \rangle$  中，当  $e = \langle u, v \rangle$  时，称  $u$  和  $v$  是  $e$  的端点（或结点），并称  $e$  与  $u$  和  $v$  是关联的。由于有向图可以由无向图规定边的方向得到，故本研究讨论的图都以无向图为例。

定义 3 设  $G = \langle V, E \rangle$  是图，称图的一个点、边交错序列  $(V_0 e_1 V_1 e_2 \dots V_{n-1} e_n V_n)$  为结点  $V_0$  到  $V_n$  的一条通路，其中  $e_i = \langle V_{i-1}, V_i \rangle (i = 1, 2, \dots, n)$ ， $V_0, V_n$  分别称为通路的起点和终点，当起点和终点相同时则称为回路。如果一条通路中结点  $V_0, V_1, V_2, \dots, V_{n-1}, V_n$  互不相同，则称为路径。建立网络最短路数学模型的理论思想为：给定  $m$  个点  $V_i (i = 1, 2, \dots, m)$  组成集合  $\{V_i\}$ ，将集合中任一点  $V_i$  到另一点  $V_j$  的距离用  $c_{ij}$  表示，如果  $V_i$  到  $V_j$  没有边连接则不定义  $c_{ij}$ ，规定

$c_{ij} = 0 (1 \leq i \leq n)$ 。指定一个终点  $V_m$ ，用动态规划的方法求从  $V_i$  到  $V_m$  的最短路线，具体思路如下：用点  $V_i$  表示状态，决策集合就是除  $V_i$  以外的点，选定一个点  $V_j$  以后，得到效益  $c_{ij}$  并转入新状态  $V_j$ ，当状态是  $V_m$  时，过程停止。显然这是一个不定期多段决策过程。

定义  $f(i)$  是由  $V_i$  点出发至终点  $V_m$  的最短路程，由最优化原理可得到网络最短路的数学模型：

$$\begin{cases} f(i) = \min\{c_{ij} + f(j)\}, & i = 1, 2, \dots, m-1, m \\ f(m) = 0 \end{cases}$$

### 5.1.4 最短哈密顿路建模

#### 5.1.4.1 建立数学模型

定义 4 设  $G = \langle V, E \rangle$  是无向图， $V = \{V_1, V_2, \dots, V_n\}$ ， $E = \{e_1, e_2, \dots, e_n\}$ ，令  $a_{ij} = \begin{cases} 1, & e_i \text{ 与 } V_i \text{ 相关联} \\ 0, & \text{否则} \end{cases}$ ，则称  $(a_{ij})_{n \times m}$  为  $G$  的点边关联矩阵，简记  $A$ 。

以园区线路中各边长度为系数，赋予园区线路图各边权向量  $c$ ，则在节点组合及任意两节点网络最短路优化后求解经过所有节点的最短哈密顿路可以转化为如下的目标函数及约束条件：

$$\begin{aligned} & \min c^T x; \\ & s.t. \quad Ax = 2I \quad (1) \\ & \quad x_i = 0, 1 \quad (2) \end{aligned}$$

其中， $c$  为  $m$  维权向量，上标  $T$  表示转置； $x$  为待求的  $m$  维边决策向量，它只有  $n$  个分量等于 1，即所求最短哈密顿路中对应边的分量取 1，否则取 0； $A$  是  $n \times m$  点边关联矩阵 ( $0 \sim 1$  矩阵，且每列只有两个元素为 1)； $I$  表示所有分量为 1 的向量。

对上述目标函数及约束条件在园区游玩图（如图 4.2）中作进一步分析：设

$G = \langle V, E, c \rangle$ , 其中  $V = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}\}$  为项目集,  $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}\}$  为连接项目的边集,  $c = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}\}$  为边集所对应的长度集。

$x = (x_1, x_2, \dots, x_{11})^T$ , 其中  $x_i = \begin{cases} 1, & \text{边 } e_i \text{ 是所求哈密顿路的一条边} \\ 0, & \text{否则} \end{cases}$ ,  $c$  为  $m$  维

长度向量,

$\min c^T x = c_1 x_1 + c_2 x_2 + \dots + c_{16} x_{16}$  表示目标, 即游玩线路总路程最短。

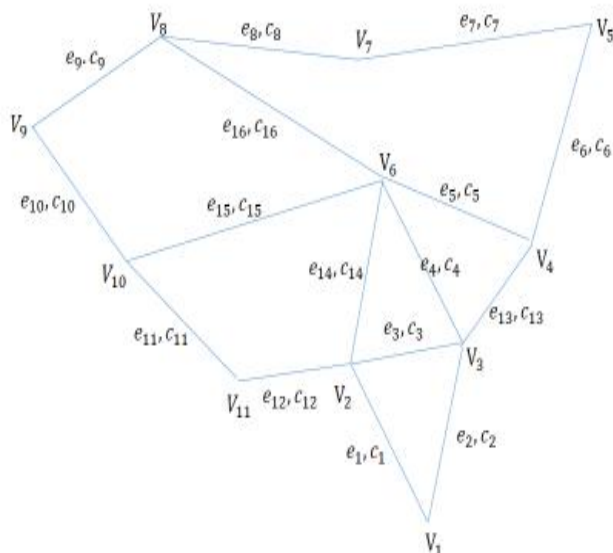


图 5.2 Youth 最短哈密顿路目标函数在园区线路图中的应用

由图可知:

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, I = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$



$$Ax = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \end{pmatrix}$$

$$= \begin{pmatrix} x_1 + x_2 \\ x_1 + x_3 + x_{12} + x_{14} \\ x_2 + x_3 + x_4 + x_{13} \\ x_5 + x_6 + x_{13} \\ x_6 + x_7 \\ x_4 + x_5 + x_{14} + x_{15} + x_{16} \\ x_7 + x_8 \\ x_8 + x_9 \\ x_9 + x_{10} \\ x_{10} + x_{11} + x_{15} \\ x_{11} + x_{12} \end{pmatrix}$$

由  $Ax = 2I$  得出以下 11 个条件

$$\begin{aligned} x_1 + x_2 &= 2 \dots \dots \dots (1) \\ x_1 + x_3 + x_{12} + x_{14} &= 2 \dots \dots \dots (2) \\ x_2 + x_3 + x_4 + x_{13} &= 2 \dots \dots \dots (3) \\ x_5 + x_6 + x_{13} &= 2 \dots \dots \dots (4) \\ x_6 + x_7 &= 2 \dots \dots \dots (5) \\ x_4 + x_5 + x_{14} + x_{15} + x_{16} &= 2 \dots \dots \dots (6) \\ x_7 + x_8 &= 2 \dots \dots \dots (7) \\ x_8 + x_9 &= 2 \dots \dots \dots (8) \\ x_9 + x_{10} &= 2 \dots \dots \dots (9) \\ x_{10} + x_{11} + x_{15} &= 2 \dots \dots \dots (10) \\ x_{11} + x_{12} &= 2 \dots \dots \dots (11) \end{aligned}$$

(1) 表示与点  $V_1$  相关联的边只有两条包含在所求游玩线路中，一条进入的边(路)和一条离开的边(路)，即确保每个项目只经过一次。(2)、(3)、(4)、... (11) 的意义与 (1) 相似。故 (1)、(2)、(3)、(4)、... (11) 表示游客经过每个项目有且仅有一次。对上述条件进行假设验证后发现虽然能够保证经过每个项目有且仅有一次，但也有可能存在子圈。要得到充要条件还需在条件(1)和(2)的基础上添加条件 (3):  $u_i - u_j + nx_{ij} \leq n - 1, 2 \leq i \neq j \leq n$ 。因此需要将上述模型

转化为:

$$\min c^T x; \quad (1)$$

$$s.t. \quad Ax = 2I \quad (2)$$

$$x_i = 0,1 \quad (2)$$

$$u_i - u_j + nx_{ij} \leq n - 1, \quad 2 \leq i \neq j \leq n \quad (3)$$

$$u_i \geq 0 \quad i = 2,3,\dots,n \quad (4)$$

#### 5.1.4.2 模型的程序实现

根据图 4.2 各条边的长度赋予距离矩阵, 若两项目之间不存在边, 则将其长度赋予 2000, 那么利用上述数学模型得到求解图 4.2 的最短哈密顿路的 Lingo 实现为 (只显示部分结果, 其余见附录)

```
Global optimal solution found.
Objective value:                4600.000
Objective bound:                4600.000
Infeasibilities:                0.8881784E-15
Extended solver steps:          0
Total solver iterations:        439
```

Variable	Value	Reduced Cost
N	11.000000	0.000000
U(1)	0.000000	0.000000
U(2)	9.000000	0.000000
U(3)	0.000000	0.000000
U(4)	2.000000	0.000000
U(5)	3.000000	0.000000
U(6)	1.000000	0.000000
U(7)	4.000000	0.000000
U(8)	5.000000	0.000000
U(9)	6.000000	0.000000
U(10)	7.000000	0.000000
U(11)	8.000000	0.000000

从程序结果分析, 图 4.2 的最短哈密顿的总路程为  $\min c^T x = 4600$ , 经过的线路为  $V_1 \rightarrow V_3 \rightarrow V_6 \rightarrow V_4 \rightarrow V_5 \rightarrow V_7 \rightarrow V_8 \rightarrow V_9 \rightarrow V_{10} \rightarrow V_{11} \rightarrow V_2 \rightarrow V_1$  或者相反

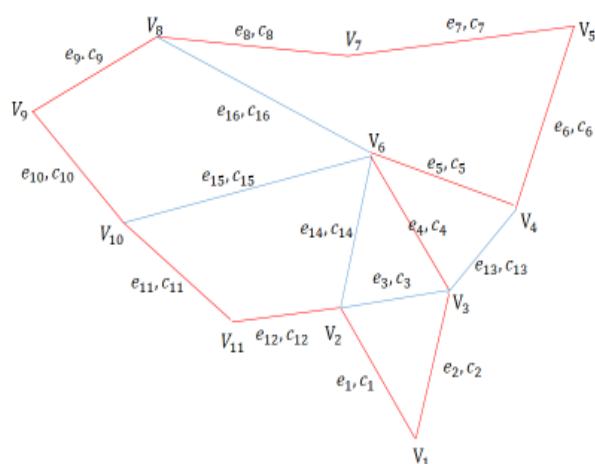


图 5.3 Youth 园区的最短哈密顿路

### 5.2 子模型 2-M/M/1/N 排队系统

排队论的主要任务是：通过建立数学模型来描述排队系统的概率规律性，研究诸如顾客平均的排队时间，排队顾客的平均数、服务员平均接待的顾客数等数量规律，为系统的最优设计与最优控制提供决策依据。本 Youth 园区环境下，应用排队论主要得出排队顾客的平均数即每个园区等待游客的平均数来为空间负荷率模型服务。

### 5.2.1 排队论的一些基本概念

一般的排队系统都有三个基本组成部分：输入过程、排队规则、服务机构。

[10]

### 1. 输入过程

输入过程即刻画顾客到达规律一种数学描述。我们主要讨论顾客的到来是 Poisson 流的情形,即在时间区间  $[t_1, t_2]$  内有  $n$  个顾客到达的概率  $p_n = (t_1, t_2)$  满足以下三个条件:

①平稳性 在 $[t_0, t_0 + t_1]$ 内到达的顾客数仅仅与时间间隔长度  $t$  有关, 而与时间起点 $t_0$ 无关。

②无后效性 在不相交的时间区间内顾客到达数是相互独立的。

③普通性 在充分小的时间间隔 $\Delta t$ 中, 有 2 个或者 2 个以上顾客到达的概率可以忽略, 即 $\sum_{n>2} p_n(t, t + \Delta t) = o(\Delta t)$ .

由于平稳性, 可简记  $p_n(0, t) = p_n(t)$ , 根据上述三个条件可以推出, 存在  $\lambda > 0$  使得  $p_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$ ,  $t > 0, n = 0, 1, 2, \dots$ ,

即在长度为  $t$  的时间内到达的顾客数服从参数为  $\lambda t$  的 Poisson 分布。

## 2. 排队规则

排队规则即顾客排队和等待的规则。排队规则一般有即时制和等待制两种。即时制是一旦服务台被占用顾客即离去；等待制是服务台被占用时，顾客便排队等候服务。等待制的服务规则有先到先服务、随机服务、有优先权的先服务等，我们将主要讨论先到先服务的等待制系统。

### 3. 服务机构

服务机构可以有一个或多个服务员，服务员为每一个顾客的服务时间是随机的，通常假设它们服从一定的概率分布。研究排队问题的目的是，研究排队系统运行的效率，估计服务质量，确定系统参数的最优值，以决定系统结构是否合理，设计改进措施等。所以，必须确定用来判断系统运行优劣的基本数量指标，这些数量指标通常是：

① 队长 指排队系统中的顾客数，它的期望值记为 $L_{\text{系}}$ ；

排队长 指在排队系统中排队等待服务的顾客数，其期望值记为 $L_{\text{队}}$ 。 $L_{\text{系}}$ 与（或者 $L_{\text{队}}$ ）越大，说明系统服务效率越低。

② 逗留时间 指一个顾客在排队系统中的停留时间，其期望值记为 $W_{\text{系}}$

等待时间 指一个顾客在排队系统中的等待时间，其期望值记为 $W_{\text{队}}$

在机器故障修理问题中，无论是等待修理或正在修理都使工厂受到停工的损失，所以在该类系统中，逗留时间（停工时间）是顾客所关心的，但在门诊看病系统中，顾客关心的往往只是等待时间的长短。

③ 忙期 指服务机构连续工作的时间长度，它关系到服务员的工作强度。忙期长度与一个忙期中平均完成服务的顾客数都是衡量服务机构效率的指标。要计算以上指标，首先要求出系统在任意时刻  $t$  的状态 $P_n(t)$ ：

$$P_n(t) = P\{\text{t时刻系统内有 } n \text{ 个顾客}\}$$

关于 $P_n(t)$  的关系式一般是微分充分方程，其解称为是瞬态解。瞬态解一般不容易求出，从应用上来说，也不知道每一时刻的系统状态，故我们通常求稳态解 $P_n(t) = \lim_{t \rightarrow \infty} P_n(t)$ 。稳态的物理含义是，当系统运行了无限长的时间之后，初始状态的概率分布的影响将消失，而且系统的状态概率分布将不再随时间变化。

[10]

### 5.2.2 M/M/1/N 系统

系统容量为  $N$ ，即排队等待的顾客最多为  $N-1$ 。在顾客到达时，如果系统中已有  $N$  个顾客，那么这个顾客就被拒绝进入系统。

状态概率的稳态方程：

$$\begin{cases} \mu P_1 = \lambda P_0 \\ \mu P_{n+1} + \lambda P_{n-1} = (\lambda + \mu) P_n \\ \mu P_N = \lambda P_{N-1} \end{cases}$$

令 $\rho = \frac{\lambda}{\mu}$ ，得到
$$\begin{cases} P_0 = \frac{1-\rho}{1-\rho^{N+1}} \\ P_n = \frac{1-\rho}{1-\rho^{N+1}} \rho^n, \quad 1 \leq n \leq N \end{cases}$$

将该系统应用于 Youth 园区之中，当园区的人流量达到平衡之后，各个项目的 $\lambda$ 是相同的，即的 $\lambda = \frac{\text{日游客总量}}{\text{日工作时间}} = 20.83$ ，各个项目的 $\mu = \frac{\text{各处项目容纳量}}{\text{服务时间}} + \frac{\text{各处相邻通道半长}}{\text{游客行走速度}}$ 。并结合以上公式得出相关计算结果如下表格 5.1。

地点	P0	L 系	L 队	W 系	负荷率
----	----	-----	-----	-----	-----

A	0.0067	18.7857	147.7924	7.1429	9.782
B	0.3947	1.5339	0.9285	0.0736	0.0767
C	0.3033	2.2966	1.5999	0.1103	0.114
D	0.484	19.6509	18.6994	0.9434	1.53
E	0.278	2.5973	1.8752	0.1247	0.127
F	0.3485	1.8698	1.2183	0.0898	0.092
G	0.1967	4.0843	3.281	0.1961	0.217
H	0.278	2.5973	1.8752	0.1247	0.133
I	0.0383	25.0964	24.1347	1.2048	2.66
J	0.3334	1.999	1.3325	0.096	0.098

表格 5.1 Youth 各园区等待游客数量的 Poission 分布与时空负荷率

### 5.3 子模型 3-时空负荷率

#### 5.3.1 园区游客分流需求

目前很多园区在旅游高峰期存在热门项目人流量大、拥挤度强、负荷率高，而一些偏冷门的项目游客量少、项目负载低的现状，导致园区各个项目的游客分布不均衡，从整个园区的层面来看，可认为总负荷率不均衡。园区项目的游客分流需求是指游客搭乘交通工具或徒步移动到下一个项目，对交通工具和人行道路产生的需求，这种需求可以用项目的时空负荷率来表示，因为在项目环境容量一定的条件下，项目的负荷越大表示项目的游客数越多，需要分流出去的游客也应越多。利用时空负荷率来表示项目当前的分流需求，可以更科学地解释项目的分流需求。因为项目的分流需求除了与当前游客数有关，还与项目的自然环境容量和游客在项目的逗留时间相关。依据园区游客流分布的历史数据、最近访问过的景点和游客到达时间、离开时间、等待时间等信息，预测园区将来的游客分布状况，获知园区项目在未来一段时间的分流需求，进而对园区的分流资源进行调度，达到园区游客分布实时均衡<sup>[6]</sup>。

在旅游项目自然生态环境容量的基础上引入游客的逗留时间，定义项目  $i$  在  $t$  时刻的时空负荷率  $L$  如下所示：

$$L = \frac{c_j}{N_j/t_j} = \frac{c_j t_j}{N_j} (j \in A; L \geq 0; j = 1, 2, \dots, n)$$

其中  $A$  为项目集； $N_j$  为项目  $j$  的自然容量； $t_j$  为旅客在项目  $J$  的平均逗留时间； $c_j$  为项目  $j$  在  $t$  时刻的游客人数。计算结果如图 5.5：

#### 5.3.2 游客满意度函数

定义  $S(t_{ij})$  为游客分流调度在等待时间为  $t_{ij}$  的满意度。表达式如下：

$$S(t_{ij}) = \begin{cases} 1 & 0 \leq t_{ij} \leq T_{min} \\ -\frac{2 + T_{min}}{2T_{max}} + \frac{2 + T_{min}}{2} & T_{min} < t_{ij} < T_{max} \\ 0 & t_{ij} \geq T_{max} \end{cases}$$

式中  $T_{min}$  为满意度为 1 的最长等待时间， $T_{max}$  为满意度为 0 的最短等待时间。心理学家将时间属性在人们心理上的综合反映称为时间知觉或时间体验，顾客在消费过程中也同样伴随着时间知觉或时间体验 (Vikas, wagner, 2001)。游客在

旅游园区游览时,同样也存在时间满意度,在本文的研究中主要体现为游客的等待时间。游客的时间满意度一般与游客等待分流的时间呈负相关性,游客的等待时间越长,其满意度就越低。所以园区在充分利用分流资源的前提下,应尽可能减少游客等待分流的时间,从而改善游客的旅游体验。

### 5.3.3 结果分析

根据图 5.5 给出的各项目的时空负荷率,要使各处负荷率达到均衡,并且在尽量符合最短路径以保证游客体验最优的前提下,对各点的提醒与疏导分析如下:

A:A 处的时空负荷率显著高于其他各处,且项目持续时间长(33min),游客体验的满意度受时间的影响波动较大,应尽量将等待的游客往 B 处疏导,从而减轻 A 园区的负担。当 A 处人数过多时,也可以向 E 处疏导,再向 B 处靠拢。

B:B 点的时空负荷率是全园区最低的,并且项目持续时间短,可提醒游客耐心等待。

C:C 处的负荷率也同样相对较低,但由于相邻的 D 区负荷率相对较高,可建议游客耐心等待,并在去往 D 区的路上提醒放慢脚步,欣赏沿途的风景。

D:该处的负荷率相对邻接的园区很高,并且地处偏远,不便向其他园区疏导,可提醒等待游客耐心排队。

E:E 点的持续时间虽然有点长,但容量大,负荷率底,可提醒游客留下耐心等待,提醒游览完去 A 和 B 没有去过的一个地方。

F:F 点的负荷率也很低,同 C 相邻负荷率较高的 D,可提醒等待的游客游览完 F 后在去 D 的路上放缓脚步,用心感受一路花草。

G:G 点的负荷率稍高,但就持续时间对满意度造成的影响来看,同样可以提醒游客耐心等待,同时也可以向 E 处疏导并让其返回。

H:H 的地理位置与 D 相似,但不同的是负荷率相对更低,邻近 I 处的负荷率高,提醒下一站去 I 的等待游客放慢脚步。

I:I 处的负荷率相对较高,主要受容纳量的限制,对游客满意度的影响也大一些,可提醒耐心等待,并疏导部分游客前往 E 游览并再次返回。

J:负荷率低,但邻近负荷率高,提醒游客在去往 A 或者 I 的路上走走停停,欣赏沿途的美好景色。

## 六、酒店预订量预测模型(问题二)

### 6.1 数据特征分析

数据来源于 Youth 游乐园内的皇冠假日酒店 2015 年全年的日预定数据。通过 Excel 对原始数据进行分类汇总,图 1 显示了汇总后的每日预订量随时间变化曲线及线性拟合趋势(图中实线表示汇总数据随时间变化,虚线表示线性拟合):酒店日预定量总体来看波动较频繁但范围窄,极大部分数据均落在[0, 200]区间内;其中,小部分数据 4/3/2015(667,清明节前最近的周五)、5/21/2015(208)和 9/18/2015(223)处超出这一范围形成峰值;而且,该数据明显受寒暑假及节假日的影响,如[0, 50]区间内大部分数据接近于零,可能与此时正处于寒假和春节(2/18/2015-2/24/2015)的影响有关;此外,一元线性拟合函数图像的斜率(0.1947)较小说明数据整体较平稳,由于  $R^2=0.1247$ ,表明拟合效果极差,因此不能用来预测下一年的日预订量。

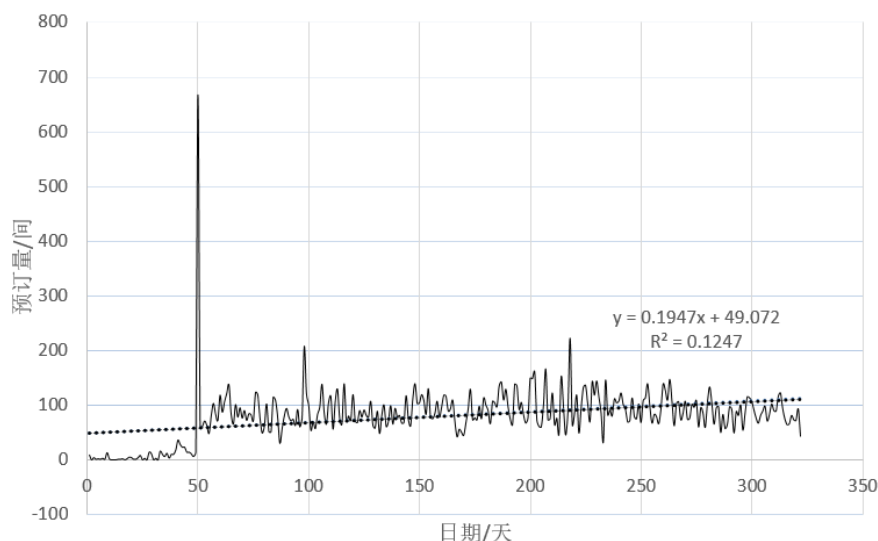


图 6.1 2015 全年日预订量曲线及一元线性拟合

将缺失数据以零补充，得 365 天完整的全年日预订量数据，在 Eviews 中进行简单的统计学分析。结果如图 2 所示，365 个观测值的均值为 71.02740，中位数为 76，标准差为 54.75705，频率分布直方图表明绝大部分观测值处于 [0, 175] 之间。<sup>[7]</sup>

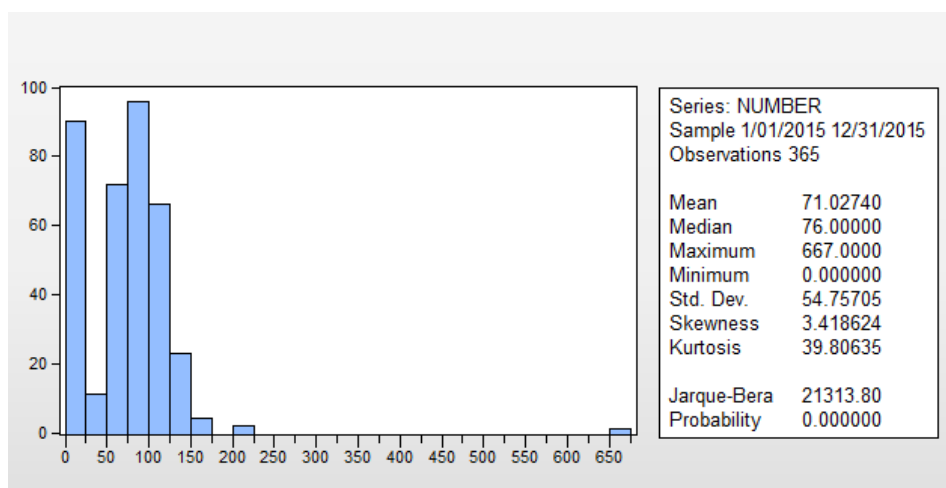


图 6.2 Eviews 简单统计学分析

由于数据波动频繁且随时间有一定的上升趋势，所以可以利用时间序列模型对数据进行分析处理。通过观察上图，结合合理的推断，我们总结出影响预订量主要因子及其描述性分类是：

天数  $t$ ——常规稳定趋势变化因素；

星期数——周期性影响因素；

节假日（寒假、春节、清明节）——突发性影响因素。

在这里，我们把 2015/01/01 天数记为 1。并且引入虚拟变量， $W_t$  代表是否寒假（平均寒假区间 [2015/1/21-2015/2/24]，是为 0，不是为 1）， $H_i$  代表清明节前第  $i$  天的节假日指数。星期的周期性变化和季节性因素是非常类似的，因此本文借用季节指数这一概念，引入星期指数来衡量每周特定星期与预订量之间的稳定关系。<sup>[1]</sup>

为了单独研究星期指数的影响情况，我们抽取了 10 月 12 日至 12 月 27 日这

11 周的数据来进行分析。因为从图 1 中可以直观上发现：首先，在这段时间里没有节假日对于预订量的影响；其次，这段时间正处于 2015 年中后期，预订量相较于初期是比较稳定的。因此，这一段时间的波动主要是由星期因素引起的，单独研究这一段预订量能更加准确地提取出星期指数。

定义星期指数：

$$S_i = \frac{SS_i}{\overline{SS}}$$

来表示星期  $i$  的预订量与平均水平的偏离情况。其中： $SS_i = \sum_t x_t^i$  表示所有星期  $i$  的预订量的总和， $x_t^i$  表示第  $t$  天是星期  $i$ ， $i=1, 2, \dots, 7$ ， $\overline{SS} = \frac{1}{7} \sum_{i=1}^7 \sum_t x_t^i$  表示这十一周 (77 天) 的预订量的总和的平均值。由定义容易知道，星期指数越大 ( $S_i > 1$ )，说明每星期的这一天预订量与平均水平相比普遍偏高；反之星期指数越小，这一天的预订量与平均水平相比普遍偏低。并且。

$$\overline{S_i} = \frac{1}{7} \sum_{i=1}^7 S_i = \frac{1}{7} \sum_{i=1}^7 \frac{SS_i}{\overline{SS}} = \frac{\sum_{i=1}^7 SS_i}{7 \times \overline{SS}} = 1$$

即星期指数的平均值为 1。通过计算，具体数据见下表：

	星期一	星期二	星期三	星期四	星期五	星期六	星期日
星期指数	1.15349 4	0.99642 7	0.86036 9	1.00743 2	1.13547 8	0.94140 3	0.90538 8

表格 6.1 星期指数

为了直观观察星期指数的影响价值，将原始数据除以相应的星期指数。得到剥离星期因素后的预订量。即：

去除星期影响的预订量 =  $\frac{\text{原始游客数}}{S_i}$  ( $i=1$  表示星期一，由此类推)

## 6.2 数据预处理

为提高预测精度并使数据平稳，数据输入模型之前需要对各模型数据进行预处理，首先对原始序列取对数 (1) 得到对数序列  $N_t$ ，再通过常用公式 (2) 将各模型数据归一化处理到区间  $[0.15, 0.85]$  内。

$$N_t = \ln N \quad (1)$$

$$Y_t = \frac{N_t - N_{\min}}{N_{\max} - N_{\min}} 0.7 + 0.15 \quad (2)$$

式中： $N_{\max}$  和  $N_{\min}$  分别代表各模型数据集的最大值和最小值。



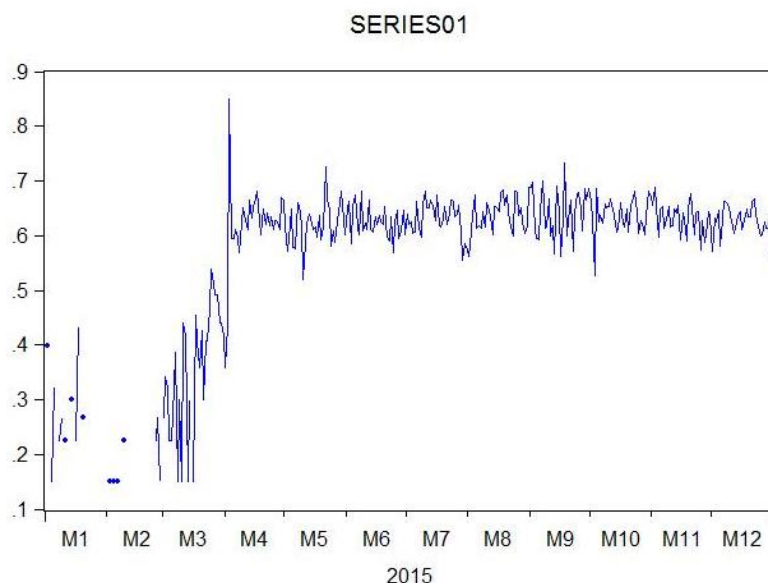


图 6.1 数据归一化处理

### 6.3 模型建立

根据前文分析，预订量主要与日期、当天是星期几以及是否为节假日有关。接下来，根据以上因子建立回归模型来分析各种因素对于预订量的影响机制。

为了方便检验模型的可靠性，在此只使用 12 个月数据中的前 11 个月的数据，剩下一个月用来做模型的事前预测，从而检验模型的有效性。

#### 6.3.1 模型结构的确定

首先，结合星期指数的计算原理，在此可以认为星期指数是以乘积的形式影响序列变化的。其次，随着时间的推移，长期递增是预订量变化的基本趋势。而节假日则会对这个趋势带来短期的突发影响。因此可以将节假日并入趋势  $T_t$  作为虚拟变量，与日期  $t$  一起进行回归分析，因而尝试使用复合模型拟合该序列：

$$Y_t = S_t g(T_t + I_t)$$

其中  $S_t$  是星期指数； $T_t$  是变化趋势； $I_t$  是模型主要的分析对象，是随机波动项，当中含有此模型中没有进行讨论的其他影响因素。

根据估算出来的星期指数  $\hat{S}_t$ ，得到消除星期影响之后的序列  $\frac{Y_t}{\hat{S}_t} = T_t + I_t$ ，这个序列基本上消除了星期因素对游客数的影响，而只剩下长期趋势和随机波动的影响。

对这个序列用直线、指数曲线和五次曲线进行拟合试验，发现五次曲线的拟合效果最好，达到 0.5149，因此对于日期  $t$  选择五次曲线的形式拟合增长趋势。再考虑到天气因素和假期因素的影响，因而尝试使用如下模型：

$$T_t = W_t (a + bt + ct^2 + dt^3 + et^4 + ft^5)$$

其中  $W_t$  是虚拟变量： $W_t$  代表是否寒假（是，取 0；否，取 1）<sup>[2]</sup>

#### 6.3.2 模型求解

在 EViews 中对于长期趋势  $T_t$  做非线性回归，得到拟合函数： $T_t = W_t (0.29073526914 - 0.000266152014861t - 0.000105351988719t^2 - 1.83566415218e-07t^3 + 5.11233652005e-08t^4 - 3.64839993411e-10t^5)$

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.290735	0.095150	3.055541	0.0039
T	-0.000266	0.020527	-0.012966	0.9897
T^2	-0.000105	0.001342	-0.078494	0.9378
T^3	-1.84E-07	3.60E-05	-0.005102	0.9960
T^4	5.11E-08	4.22E-07	0.121257	0.9041
T^5	-3.65E-10	1.79E-09	-0.203287	0.8399
R-squared	0.514902	Mean dependent var	0.315961	
Adjusted R-squared	0.455744	S.D. dependent var	0.118217	
S.E. of regression	0.087213	Akaike info criterion	-1.922175	
Sum squared resid	0.311853	Schwarz criterion	-1.685986	
Log likelihood	51.17112	Hannan-Quinn criter.	-1.833296	
F-statistic	8.703808	Durbin-Watson stat	2.404944	
Prob(F-statistic)	0.000011			

表格 6.2 非线性回归拟合参数

由图 4 可知，非线性回归拟合函数 R 平方和调整 R 平方分别为 0.514902 和 0.455744，说明拟合效果并不是很理想，这可能与序列的强随机性以及节假日影响有关，因此有待进一步修正。

日期	预测值	日期	预测值	日期	预测值
1/02/2016	0.289781	2/27/2016	0.224151	3/17/2016	0.361939
1/04/2016	0.287986	3/01/2016	0.234708	3/18/2016	0.371404
1/05/2016	0.286779	3/02/2016	0.240523	3/19/2016	0.38088
1/08/2016	0.281967	3/03/2016	0.246686	3/20/2016	0.390328
1/09/2016	0.279986	3/04/2016	0.253189	3/21/2016	0.39971
1/11/2016	0.275505	3/05/2016	0.260023	3/22/2016	0.408983
1/14/2016	0.267624	3/06/2016	0.267175	3/23/2016	0.418103
1/16/2016	0.261723	3/07/2016	0.274633	3/24/2016	0.427023
1/17/2016	0.258614	3/08/2016	0.282382	3/25/2016	0.435696
1/20/2016	0.248815	3/09/2016	0.290405	3/26/2016	0.444069
2/02/2016	0.206977	3/10/2016	0.298685	3/27/2016	0.452088
2/04/2016	0.202048	3/11/2016	0.307202	3/28/2016	0.459698
2/06/2016	0.197877	3/12/2016	0.315934	3/29/2016	0.46684
2/09/2016	0.193294	3/13/2016	0.324858	3/30/2016	0.473452
2/25/2016	0.215053	3/14/2016	0.333949	3/31/2016	0.479471
2/26/2016	0.219418	3/16/2016	0.35252		

表格 6.3 非线性回归拟合预测观测值

### 6.3.3 时间序列模型

#### 6.3.3.1 二次指数平滑法

时间序列 $y_t$ 的单次平滑序列 $\hat{S}_t$ 的计算公式如下：

$$\hat{S}_t = \alpha y_t + (1 - \alpha) y_{t-1}$$

其中， $y_t$ 是时间序列； $y_{t-1}$ 是上期平滑值序列（smoothed series）； $\alpha$ 是平滑系数（smoothing parameter），也叫衰减因子。平滑系数的取值范围为 $0 \leq \alpha \leq 1$ 。重复迭代以上公式，可得：

$$\hat{S}_t = \alpha \sum_{i=0}^{t-1} (1 - \alpha)^i y_{t-1}$$

二次指数平滑法适用于序列值在一个常数均值上下随机波动、有线性趋势无

季节要素的情况。时间序列 $y_t$ 的二次指数平滑以递归形式定义为

$$S_t = \alpha y_t + (1 - \alpha)S_{t-1}$$

$$D_t = \alpha S_t + (1 - \alpha)S_{t-1}$$

式中： $0 \leq \alpha \leq 1$ ， $S_t$ 是单指数平滑后的序列， $D_t$ 是对 $S_t$ 序列又进行一次单指数平滑，即对原序列的二次指数平滑得到的序列。二次指数平滑的预测如下：

$$\hat{y}_{T+k} = \left(2 + \frac{\alpha k}{1-\alpha}\right) S_T - \left(1 + \frac{\alpha k}{1-\alpha}\right) D_T = 2S_T - D_T + \frac{\alpha}{1-\alpha} (S_T - D_T)k$$

(3.3.1)

式中： $k > 0$ 。式（3.3.1）表明二次指数平滑的预测具有截距为 $2S_T - D_T$ ，斜率为 $\alpha k(S_T - D_T)/(1 - \alpha)$ 的线性趋势。<sup>[3]</sup>

Date: 05/02/16 Time: 20:39		
Sample: 3/16/2015 11/30/2015		
Included observations: 260		
Method: Double Exponential		
Original Series: SERIES01		
Forecast Series: SERIESSM		
<hr/>		
Parameters:	Alpha	0.1280
Sum of Squared Residuals		0.700564
Root Mean Squared Error		0.051908
<hr/>		
End of Period Levels:	Mean	0.620350
	Trend	-0.000666
<hr/>		

图 6.2 二次指数平滑预测参数

拟合函数： $\hat{y}_{T+k} = W_t(1.8719S_T + 0.8721S_{T-1})$

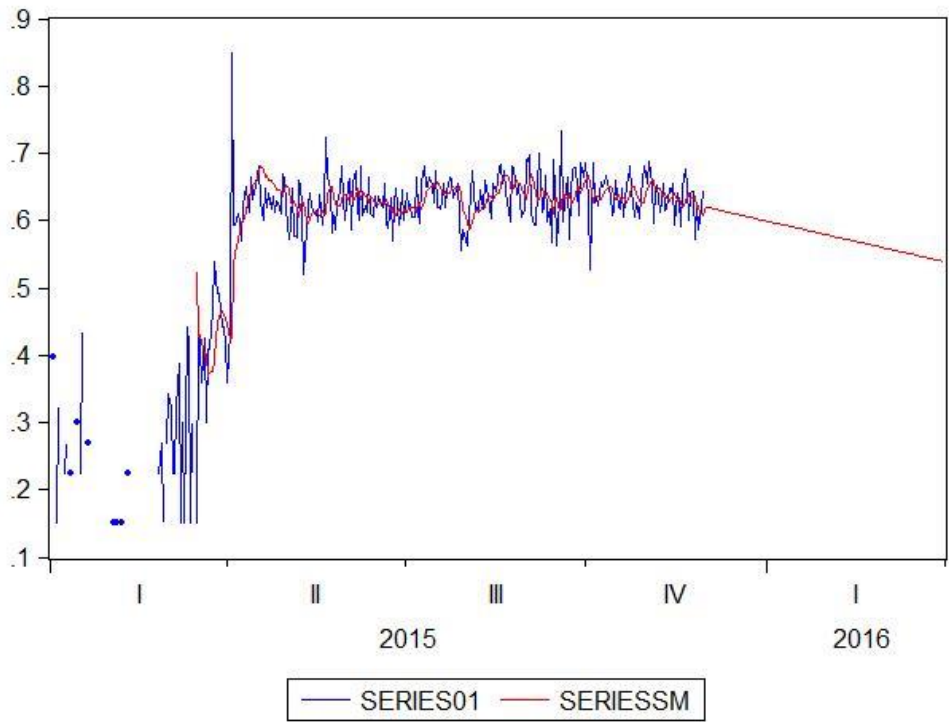


图 6.3 二次指数平滑预测结果

预测值	实际值	百分差	预测值	实际值	百分差
0.619684	0.571114	0.085045	0.609029	0.62696	0.0286
0.619018	0.633184	0.022372	0.608363	0.648911	0.062486
0.618352	0.624366	0.009632	0.607697	0.635576	0.043865
0.617686	0.6468	0.045011	0.607031	0.634387	0.043122
0.617021	0.581374	0.061315	0.606365	0.662629	0.084911
0.616355	0.629493	0.020872	0.605699	0.668013	0.093282
0.615689	0.661705	0.069543	0.605033	0.639068	0.053257
0.615023	0.659833	0.067912	0.604367	0.614761	0.016907
0.614357	0.654013	0.060636	0.603701	0.599357	0.007249
0.613691	0.636753	0.036219	0.603035	0.602619	0.000691
0.613025	0.617594	0.007398	0.602369	0.624366	0.035231
0.612359	0.604214	0.013481	0.601703	0.613316	0.018934
0.611693	0.624366	0.020298	0.601037	0.611851	0.017674
0.611027	0.634387	0.036823	0.600371	0.639068	0.060552
0.610361	0.644647	0.053186	0.599705	0.557353	0.075988
0.609695	0.610366	0.0011			

表格 6.4 二次指数平滑法预测 12 月数据的检验

SERIES01 代表原序列，SERIESSM 代表预测曲线，由拟合百分差可知 12 月数据拟合程度较高。

经反归一化及指数变换得 2016/01/01-2016/03/31 预测结果：

日期	预测值	日期	预测值	日期	预测值
2016/1/1	1.78167	2016/1/31	1.778624	2016/3/1	1.775584
2016/1/2	1.781568	2016/2/1	1.778522	2016/3/2	1.775483
2016/1/3	1.781466	2016/2/2	1.778421	2016/3/3	1.775381
2016/1/4	1.781364	2016/2/3	1.77832	2016/3/4	1.77528
2016/1/5	1.781263	2016/2/4	1.778218	2016/3/5	1.775179
2016/1/6	1.781161	2016/2/5	1.778117	2016/3/6	1.775078
2016/1/7	1.78106	2016/2/6	1.778015	2016/3/7	1.774976
2016/1/8	1.780958	2016/2/7	1.777914	2016/3/8	1.774875
2016/1/9	1.780857	2016/2/8	1.777813	2016/3/9	1.774774
2016/1/10	1.780755	2016/2/9	1.777711	2016/3/10	1.774673
2016/1/11	1.780654	2016/2/10	1.77761	2016/3/11	1.774572
2016/1/12	1.780552	2016/2/11	1.777509	2016/3/12	1.77447
2016/1/13	1.78045	2016/2/12	1.777407	2016/3/13	1.774369
2016/1/14	1.780349	2016/2/13	1.777306	2016/3/14	1.774268
2016/1/15	1.780247	2016/2/14	1.777205	2016/3/15	1.774167
2016/1/16	1.780146	2016/2/15	1.777103	2016/3/16	1.774066
2016/1/17	1.780044	2016/2/16	1.777002	2016/3/17	1.773965
2016/1/18	1.779943	2016/2/17	1.776901	2016/3/18	1.773863
2016/1/19	1.779841	2016/2/18	1.776799	2016/3/19	1.773762
2016/1/20	1.77974	2016/2/19	1.776698	2016/3/20	1.773661
2016/1/21	1.779638	2016/2/20	1.776597	2016/3/21	1.77356

2016/1/22	1.779537	2016/2/21	1.776495	2016/3/22	1.773459
2016/1/23	1.779435	2016/2/22	1.776394	2016/3/23	1.773358
2016/1/24	1.779334	2016/2/23	1.776293	2016/3/24	1.773257
2016/1/25	1.779233	2016/2/24	1.776191	2016/3/25	1.773155
2016/1/26	1.779131	2016/2/25	1.77609	2016/3/26	1.773054
2016/1/27	1.77903	2016/2/26	1.775989	2016/3/27	1.772953
2016/1/28	1.778928	2016/2/27	1.775888	2016/3/28	1.772852
2016/1/29	1.778827	2016/2/28	1.775786	2016/3/29	1.772751
2016/1/30	1.778725	2016/2/29	1.775685	2016/3/30	1.77265
				2016/3/31	1.772549

表格 6.5 2016/1/1-2016/3/31 酒店预订量预测值

### 3.3.2 Holt-Winters——无季节性模型

这种方法适用于具有线性时间趋势但无季节变化的序列。与二次指数平滑法一样，这种方法以线性趋势进行预测。但不同的是，二次指数平滑法只用一个参数，而这种方法用两个参数。 $y_t$ 平滑后的序列 $\hat{y}_t$ 由下式给出

$$\hat{y}_{t+k} = a_t + b_t k$$

式中： $a_t$ 表示截距， $b_t$ 表示斜率，这两个参数由如下递归式定义

$$a_t = \alpha y_t + (1 - \alpha)(a_{t-1} + b_{t-1})$$

$$b_t = \beta(a_t - a_{t-1}) + (1 - \beta)b_{t-1}$$

式中： $k > 0$ ， $\alpha$ ， $\beta$  在 0~1 之间，为阻尼因子。预测值由下式计算

$$y_{t+k} = a_T + b_T k$$

Date: 05/02/16 Time: 20:30		
Sample: 3/16/2015 11/30/2015		
Included observations: 260		
Method: Holt-Winters No Seasonal		
Original Series: SERIES01		
Forecast Series: SERIESSM		
<hr/>		
Parameters:	Alpha	0.3600
	Beta	0.0300
Sum of Squared Residuals		0.656868
Root Mean Squared Error		0.050263
<hr/>		
End of Period Levels:	Mean	0.624698
	Trend	-0.000278
<hr/>		

图 4.6 Holt-Winters——无季节性预测参数

拟合函数：

$$y_{t+k} = 0.36y_t + 0.64(a_{t-1} + b_{t-1}) + 0.03k(a_t - a_{t-1}) + 0.97b_{t-1}$$

其中， $a_t$ 表示截距， $b_t$ 表示斜率。<sup>[3]</sup>

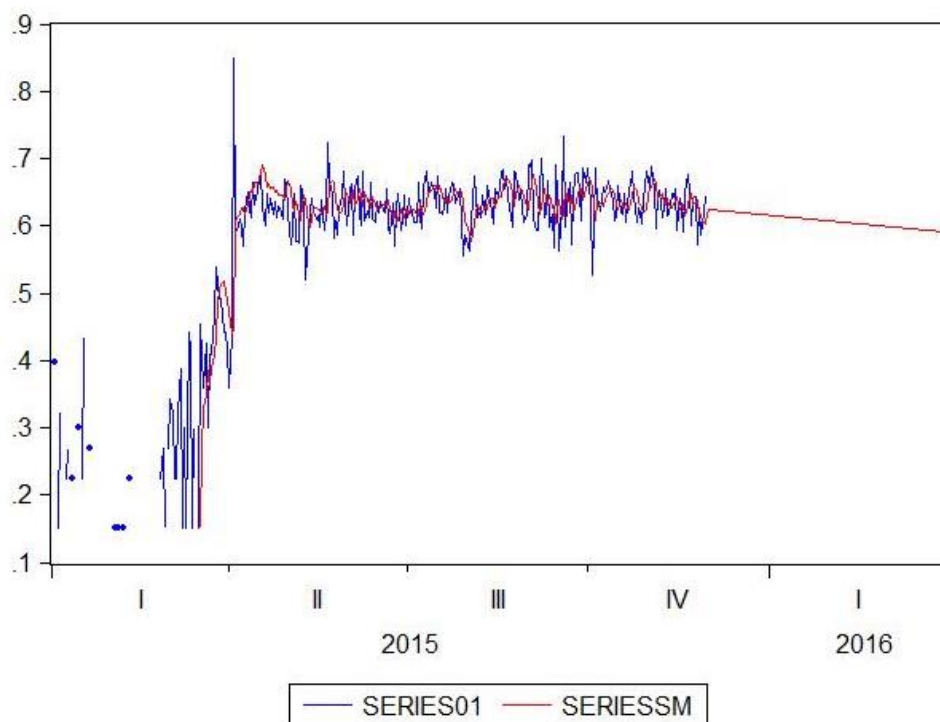


图 6.5 Holt-Winters—无季节性模型结果

预测值	实际值	百分差	预测值	实际值	百分差
0.62442	0.571114	0.085369	0.619971	0.62696	0.011148
0.624142	0.633184	0.014487	0.619693	0.648911	0.045026
0.623864	0.624366	0.000805	0.619415	0.635576	0.025429
0.623586	0.6468	0.037226	0.619136	0.634387	0.02404
0.623308	0.581374	0.067277	0.618858	0.662629	0.066057
0.62303	0.629493	0.010374	0.61858	0.668013	0.074
0.622752	0.661705	0.062551	0.618302	0.639068	0.032494
0.622474	0.659833	0.060018	0.618024	0.614761	0.005308
0.622195	0.654013	0.051138	0.617746	0.599357	0.030682
0.621917	0.636753	0.023855	0.617468	0.602619	0.024641
0.621639	0.617594	0.006508	0.61719	0.624366	0.011494
0.621361	0.604214	0.027597	0.616912	0.613316	0.005863
0.621083	0.624366	0.005286	0.616634	0.611851	0.007816
0.620805	0.634387	0.021878	0.616355	0.639068	0.03554
0.620527	0.644647	0.03887	0.616077	0.557353	0.105362
0.620249	0.610366				

表格 6.6 Holt-Winters—无季节性模型预测 12 月数据的检验

SERIES01 表示实际观测值曲线，SERIESSM 表示拟合函数曲线。

经反归一化及指数变换得 2016/01/01-2016/03/31 预测结果，可知 Holt-Winters—无季节性模型的拟合效果不如二次指数平滑模型。

### 3. 3. 3 基于时间序列的 ARIMA 模型

NUMBER11 序列单位根（ADF）检验：

Null Hypothesis: NUMBER11 has a unit root  
Exogenous: Constant  
Lag Length: 6 (Automatic - based on SIC, maxlag=16)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-2.604402	0.0931
Test critical values: 1% level	-3.450161	
5% level	-2.870164	
10% level	-2.571434	

\*MacKinnon (1996) one-sided p-values.

图 6.6 NUMBER11 序列单位根 (ADF) 检验结果

NUMBER11 的 ADF 检验值:  $-2.604402 > 1\%$  和  $10\%$  对应的 T 统计量, 因此可以认为 NUMBER11 序列是不平稳的。为了消除序列 NUMBER11 的非平稳性, 需要对 NUMBER11 序列进行差分处理。在 EViews 命令窗口中输入命令: `series DNUMBER11=NUMBER11-NUMBER(-1)`, 即可生成一阶差分序列 DNUMBER11。<sup>[5]</sup> 对 DNUMBER11 进行单位根 (ADF) 检验:

Null Hypothesis: DNUMBER11 has a unit root  
Exogenous: Constant  
Lag Length: 6 (Automatic - based on SIC, maxlag=16)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-12.65607	0.0000
Test critical values: 1% level	-3.450223	
5% level	-2.870192	
10% level	-2.571449	

\*MacKinnon (1996) one-sided p-values.

图 6.7 DNUMBER11 序列单位根 (ADF) 检验结果

DNUMBER11 的 ADF 检验值:  $-12.65607 < 1\%$  对应的 T 统计量, 有  $0\%$  的把握认为 DNUMBER11 序列有一个单位根, 因此可以认为 DNUMBER11 序列是平稳的。对 DNUMBER11 进行相关性检验:

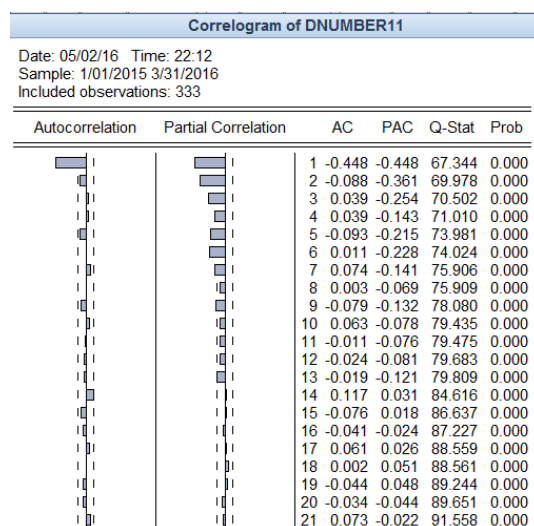


图 6.8 序列 DNUMBER11 的自相关图和 Q 统计量

由图可知, DNUMBER11 序列的偏自相关 (Partial Correlation) 函数呈指数衰减, 但衰减速度非常缓慢; 而该序列的自相关 (Autocorrelation) 函数衰减地非常快, 在第二期就衰减为 0, 因此同样可以判断该序列为平稳序列, 可以进行 ARIMA 建模。根据对序列 DNUMBER11 特征的分析, 打算建立一个 ARIMA(P,D,Q) 模型, 在估计模型之前首先要确定模型的形式即 P、D、Q 的阶数。为了消除序



列的趋势和非平稳性，对序列 **NUMBER11** 进行过一次差分，因此  $D=1$ ；观察图 中 **DNUMBER11** 序列的相关图，该序列的自相关函数在第一期后基本截尾，因此  $Q=1$ ；偏自相关函数拖尾，到第 8 期才为 0，因此可以判断  $P=0$ 。

综上所述，本案例考虑建立一个 **ARIMA (0,1,1)** 模型。

模型估计：

对本案例中的 **ARIMA (0,1,1)** 模型进行估计，操作过程如下：单击 **EViews** 主菜单中的 **Quick | Estimate Equation** 对话框中的 **Estimation specification** 输入框中输入：**DNUMBER11 c ma(1)**，得到如图 所示的模型估计结果。

Dependent Variable: DNUMBER11				
Method: Least Squares				
Date: 05/02/16 Time: 22:17				
Sample (adjusted): 1/02/2015 11/30/2015				
Included observations: 333 after adjustments				
Convergence achieved after 8 iterations				
MA Backcast: 1/01/2015				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.270788	0.223440	1.211907	0.2264
MA(1)	-0.909368	0.023023	-39.49891	0.0000
R-squared	0.441624	Mean dependent var	0.279279	
Adjusted R-squared	0.439937	S.D. dependent var	58.16628	
S.E. of regression	43.53011	Akaike info criterion	10.39077	
Sum squared resid	627202.2	Schwarz criterion	10.41364	
Log likelihood	-1728.063	Hannan-Quinn criter.	10.39989	
F-statistic	261.7903	Durbin-Watson stat	1.960214	
Prob(F-statistic)	0.000000			
Inverted MA Roots	.91			

图 6.9 ARIMA(0, 1, 1)模型估计结果

MA 过程有一个复根，这个根的倒数为  $0.91 < 1$ ，因此各滞后多项式倒数根的模都在单位圆内，从而表明 ARIMA 模型是平稳的也是可逆的。

模型预测：

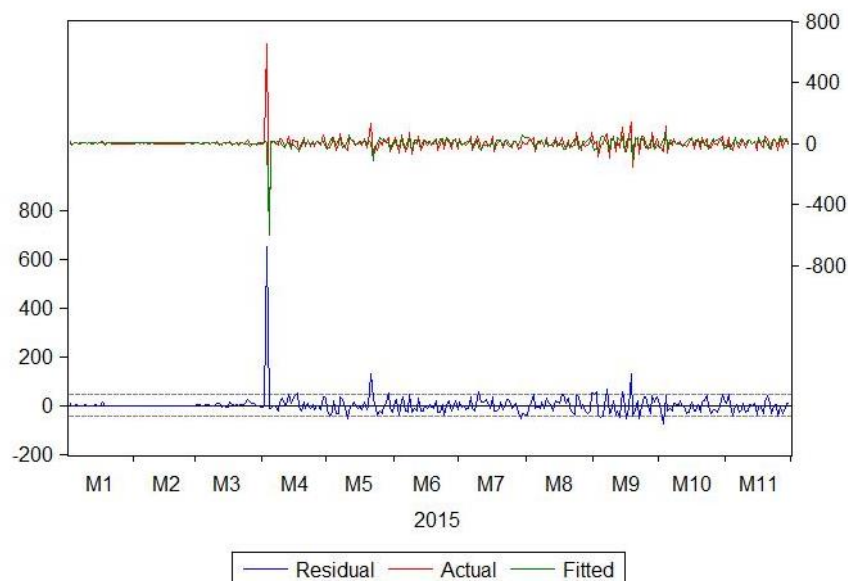


图 6.10 ARIMA(0, 1, 1)模型拟合曲线



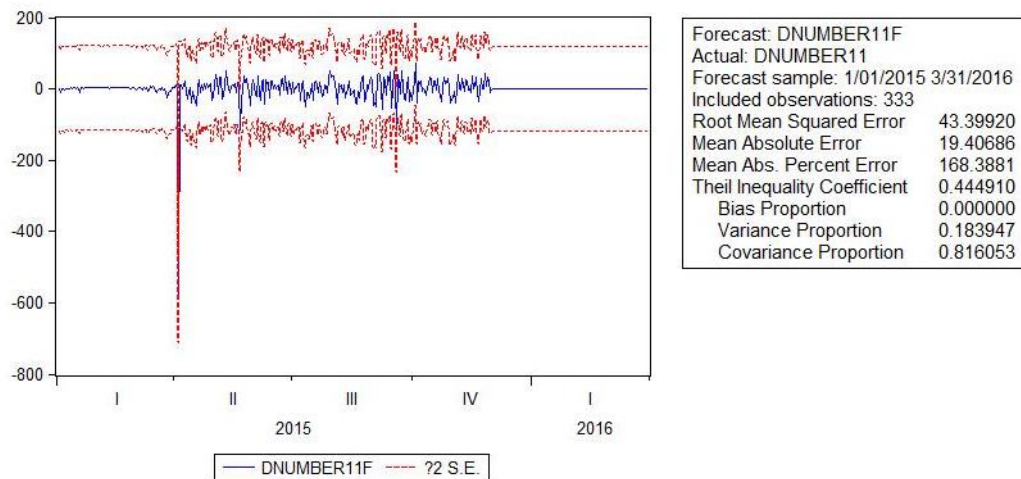


图 6.11 ARIMA(0, 1, 1)模型的预测结果

2016/1/1	0.270788	2016/2/1	0.270788	2016/3/1	0.270788
2016/1/2	0.270788	2016/2/2	0.270788	2016/3/2	0.270788
2016/1/3	0.270788	2016/2/3	0.270788	2016/3/3	0.270788
2016/1/4	0.270788	2016/2/4	0.270788	2016/3/4	0.270788
2016/1/5	0.270788	2016/2/5	0.270788	2016/3/5	0.270788
2016/1/6	0.270788	2016/2/6	0.270788	2016/3/6	0.270788
2016/1/7	0.270788	2016/2/7	0.270788	2016/3/7	0.270788
2016/1/8	0.270788	2016/2/8	0.270788	2016/3/8	0.270788
2016/1/9	0.270788	2016/2/9	0.270788	2016/3/9	0.270788
2016/1/10	0.270788	2016/2/10	0.270788	2016/3/10	0.270788
2016/1/11	0.270788	2016/2/11	0.270788	2016/3/11	0.270788
2016/1/12	0.270788	2016/2/12	0.270788	2016/3/12	0.270788
2016/1/13	0.270788	2016/2/13	0.270788	2016/3/13	0.270788
2016/1/14	0.270788	2016/2/14	0.270788	2016/3/14	0.270788
2016/1/15	0.270788	2016/2/15	0.270788	2016/3/15	0.270788
2016/1/16	0.270788	2016/2/16	0.270788	2016/3/16	0.270788
2016/1/17	0.270788	2016/2/17	0.270788	2016/3/17	0.270788
2016/1/18	0.270788	2016/2/18	0.270788	2016/3/18	0.270788
2016/1/19	0.270788	2016/2/19	0.270788	2016/3/19	0.270788
2016/1/20	0.270788	2016/2/20	0.270788	2016/3/20	0.270788
2016/1/21	0.270788	2016/2/21	0.270788	2016/3/21	0.270788
2016/1/22	0.270788	2016/2/22	0.270788	2016/3/22	0.270788
2016/1/23	0.270788	2016/2/23	0.270788	2016/3/23	0.270788
2016/1/24	0.270788	2016/2/24	0.270788	2016/3/24	0.270788
2016/1/25	0.270788	2016/2/25	0.270788	2016/3/25	0.270788
2016/1/26	0.270788	2016/2/26	0.270788	2016/3/26	0.270788
2016/1/27	0.270788	2016/2/27	0.270788	2016/3/27	0.270788
2016/1/28	0.270788	2016/2/28	0.270788	2016/3/28	0.270788
2016/1/29	0.270788	2016/2/29	0.270788	2016/3/29	0.270788
2016/1/30	0.270788			2016/3/30	0.270788

2016/1/31	0.270788			2016/3/31	0.270788
-----------	----------	--	--	-----------	----------

表格 6.7 2016/1/1-2016/3/31 酒店预订量 ARIMA 模型预测值

结合图 13 和图 14 可知，经一阶差分后的序列拟合程度比较高，说明 ARIMA (0, 1, 1) 模型的拟合是成功的。但是由于问题仅提供了一年的数据，无法以较高精确度反映序列的周期性，因此预测结果的置信区间较大，从而造成 R 平方和调整 R 平方水平较低的拟合结果。

模型改进：

引入节假日虚拟变量对非线性回归模型和时间序列模型进行修正，定义： $H_i$  代表清明节前第  $i$  天的节假日指数。<sup>[8]</sup>

$H_i$	$i$	$H_i$	$i$
0.215768	16	0.39834	8
0.06639	15	0.248963	7
0.182573	14	0.248963	6
0.182573	13	0.19917	5
0.315353	12	0.116183	4
0.614108	11	0.232365	3
0.481328	10	11.07054	2
0.39834	9	1.029046	1

表格 6.8 清明节前第  $i$  天的节假日指数

引入节假日指数的非线性回归预测模型：

$$T_t = H_i W_t (0.29073526914 - 0.000266152014861t - 0.000105351988719t^2 - 1.83566415218e-07t^3 + 5.11233652005e-08t^4 - 3.64839993411e-10t^5)$$

模型修正后的预测结果：

2016/1/1	15	2016/2/1	0	2016/3/1	4
2016/1/2	2	2016/2/2	0	2016/3/2	5
2016/1/3	2	2016/2/3	0	2016/3/3	5
2016/1/4	8	2016/2/4	0	2016/3/4	6
2016/1/5	2	2016/2/5	0	2016/3/5	6
2016/1/6	2	2016/2/6	0	2016/3/6	6
2016/1/7	3	2016/2/7	0	2016/3/7	6
2016/1/8	4	2016/2/8	0	2016/3/8	7
2016/1/9	2	2016/2/9	0	2016/3/9	7
2016/1/10	0	2016/2/10	0	2016/3/10	7
2016/1/11	0	2016/2/11	0	2016/3/11	7
2016/1/12	0	2016/2/12	0	2016/3/12	7
2016/1/13	0	2016/2/13	0	2016/3/13	7
2016/1/14	0	2016/2/14	2	2016/3/14	8
2016/1/15	0	2016/2/15	2	2016/3/15	8
2016/1/16	0	2016/2/16	2	2016/3/16	8
2016/1/17	0	2016/2/17	2	2016/3/17	9
2016/1/18	0	2016/2/18	2	2016/3/18	9
2016/1/19	0	2016/2/19	2	2016/3/19	19
2016/1/20	0	2016/2/20	2	2016/3/20	6

2016/1/21	0	2016/2/21	3	2016/3/21	16
2016/1/22	0	2016/2/22	3	2016/3/22	16
2016/1/23	0	2016/2/23	3	2016/3/23	28
2016/1/24	0	2016/2/24	3	2016/3/24	54
2016/1/25	0	2016/2/25	3	2016/3/25	42
2016/1/26	0	2016/2/26	3	2016/3/26	35
2016/1/27	0	2016/2/27	3	2016/3/27	35
2016/1/28	0	2016/2/28	4	2016/3/28	22
2016/1/29	0	2016/2/29	4	2016/3/29	22
2016/1/30	0			2016/3/30	17
2016/1/31	0			2016/3/31	10

表格 6.9 引入节假日指数的非线性回归模型预测结果

## 七、模型评价

### 7.1 问题一模型评价

本研究在对 Youth 游乐园相关情况进行分析的基础上，阐述了景区客流疏导的理论基础和游览线路优化的内容、原则及景区客流疏导与游览线路优化的相关性，并提出通过对景区游览线路进行优化促进对高峰期客流进行疏导，再利用图论中的网络最短路模型和最短哈密顿路模型对景区游览线路进行优化，使游客在游览目标的指引下能够尽快到达目标点，缓解游客因从众、走错路、重路、弯路引起的景区拥堵。

但此模型本研究中的对游览提醒和疏导是在诸多假设的基础上进行研究的，对园区游览线路做了大量简化，在更为复杂的线路网络中，需要进行进一步的修正与改进。并且路线优化只是辅助景区进行客流疏导的一个手段，要完全实现客流高峰期景区客流疏导，需要考虑更多的因素。

### 7.2 问题二模型评价

酒店预订量需求预测随着旅游热越来越重要。本文根据酒店 2015 年的预订量分布特征建立了非线性回归模型和时间序列模型，并通过相关系数确定各模型输入项，设定星期指数消除工作日和周末的影响，并对节假日采用虚拟变量予以表征。同整年度模型相比加入虚拟变量的非线性回归模型的各方面预测结果评价指标都更优，原因主要是节假日虚拟变量大大降低了数据的跳跃波动性。总之，本文按数据分布特征所建立的非线性回归模型不仅为酒店预订量预测方法提供了一种新的视角，同时酒店决策人员也可以根据预订量预测对物资、人员等提前进行安排，对提高酒店服务质量、合理配置房间资源以及提高营业收益等方面具有重要意义。

虽然非线性回归模型和时间序列模型能较好地对日预订量进行预测，但仍有需要改进提高之处，如两种模型在用虚拟变量处理之前得到的拟合曲线有较大的线性趋势，不能很好地反映预订量随时间变化的周期性，此外预测误差仍然很大而且对酒店淡季的预测效果不好。因此下一步计划是收集分析更多与预订量相关的数据，如游乐场日客流量和酒店日住宿量等数据，同时数据处理方法可能也需要进一步优化。

## 参考文献

- [1] 成诚, 杜豫川, 刘新, 考虑节假日效应的交通枢纽客流量预测模型[J], 交通运输系统工程与信息, Vol. 15, No. 5, 2015。
- [2] 杜强, 贾丽艳, SPSS 统计分析从入门到精通[M], 北京: 人民邮电出版社, 2009。
- [3] 高铁梅, 计量经济分析方法与建模: EViews 应用及实例[M], 北京: 清华大学出版社, 2009。
- [4] 姜启源, 谢金星, 叶俊, 数学模型(第四版)[M], 北京: 高等教育出版社, 2011。
- [5] 梁昌勇, 马银超, 陈荣, 梁焱, 基于 SVR-ARMA 组合模型的日旅游需求预测[J], 管理工程学报, Vol. 29, No. 1: 122-127, 2015。
- [6] 李山, 王铮, 钟章奇, 旅游空间相互作用的引力模型及其应用[J], 地理学报, 2012
- [7] 李嫣怡等, EViews 统计分析与应用(修订版)[M], 北京: 电子工业出版社, 2013。
- [8] 马银超, 梁昌勇, 路彩虹, 基于分类模型的日客流量预测[J], 国土资源科技管理, Vol. 32, No. 5: 109-114, 2015。
- [9] 饶亚玲, 基于客流疏导的景区游览线路优化研究, 福建, 华侨大学, 2015。
- [10] 吴孟达, 成礼智, 吴翊, 毛紫阳, 王丹, 数学建模教程, 北京: 高等教育出版社, 2011。
- [11] 张佑印, 顾静, 马耀峰, 旅游流研究的进展、评价与展望[J], 旅游学刊, 2013

## 附录

### 问题一 哈密顿 lingo 程序及结果

```

model:
sets:
spot/1..11/:u;
link(spot,spot):
dist, x;
endsets
n=@size(spot); data:
dist=
0    300  400  2000 2000 2000 2000 2000 2000 2000 2000
300  0    300  2000 2000 350  2000 2000 2000 2000 250
400  300  0    300  2000 350  2000 2000 2000 2000 2000
2000 2000 300  0    450  500  2000 2000 2000 2000 2000
2000 2000 2000 450  0    2000 500  2000 2000 2000 2000
2000 350  350  500  2000 0    2000 550  2000 450  2000
2000 2000 2000 2000 500  2000 0    650  2000 2000 2000
2000 2000 2000 2000 2000 550  650  0    400  2000 2000
2000 2000 2000 2000 2000 2000 2000 400  0    450  2000
2000 2000 2000 2000 2000 450  2000 2000 450  0    350
2000 250  2000 2000 2000 2000 2000 2000 2000 350  0
;
enddata
min=@sum(link:dist*x);
@FOR(spot(K):
@sum(spot(I)|I #ne# K:x(I,K))=1;
@sum(spot(J)|J #ne# K:x(K,J))=1;
);
@for(spot(I)|I #gt# 1:
@for(spot(J)|J #gt# 1 #and# I #ne# J:
u(I)-u(J)+n*x(I,J)<=n-1);
@for(spot(I)|I #gt# 1:u(I)<=n-2);
);
@for(link:@bin(x));
End
Global optimal solution found.
Objective value:                4600.000
Objective bound:                4600.000
Infeasibilities:                0.8881784E-15
Extended solver steps:          0
Total solver iterations:        439

```

Variable	Value	Reduced Cost
N	11.00000	0.000000
U( 1)	0.000000	0.000000
U( 2)a	9.000000	0.000000
U( 3)b	0.000000	0.000000
U( 4)c	2.000000	0.000000
U( 5)d	3.000000	0.000000
U( 6)e	1.000000	0.000000
U( 7)f	4.000000	0.000000
U( 8)g	5.000000	0.000000
U( 9)h	6.000000	0.000000
U( 10)i	7.000000	0.000000
U( 11)j	8.000000	0.000000
DIST( 1, 1)	0.000000	0.000000
DIST( 1, 2)	300.0000	0.000000
DIST( 1, 3)	400.0000	0.000000
DIST( 1, 4)	2000.000	0.000000
DIST( 1, 5)	2000.000	0.000000
DIST( 1, 6)	2000.000	0.000000
DIST( 1, 7)	2000.000	0.000000
DIST( 1, 8)	2000.000	0.000000
DIST( 1, 9)	2000.000	0.000000
DIST( 1, 10)	2000.000	0.000000
DIST( 1, 11)	2000.000	0.000000
DIST( 2, 1)	300.0000	0.000000
DIST( 2, 2)	0.000000	0.000000
DIST( 2, 3)	300.0000	0.000000
DIST( 2, 4)	2000.000	0.000000
DIST( 2, 5)	2000.000	0.000000
DIST( 2, 6)	350.0000	0.000000
DIST( 2, 7)	2000.000	0.000000
DIST( 2, 8)	2000.000	0.000000
DIST( 2, 9)	2000.000	0.000000
DIST( 2, 10)	2000.000	0.000000
DIST( 2, 11)	250.0000	0.000000
DIST( 3, 1)	400.0000	0.000000
DIST( 3, 2)	300.0000	0.000000
DIST( 3, 3)	0.000000	0.000000
DIST( 3, 4)	300.0000	0.000000
DIST( 3, 5)	2000.000	0.000000
DIST( 3, 6)	350.0000	0.000000
DIST( 3, 7)	2000.000	0.000000
DIST( 3, 8)	2000.000	0.000000
DIST( 3, 9)	2000.000	0.000000

DIST( 3, 10)	2000.000	0.000000
DIST( 3, 11)	2000.000	0.000000
DIST( 4, 1)	2000.000	0.000000
DIST( 4, 2)	2000.000	0.000000
DIST( 4, 3)	300.0000	0.000000
DIST( 4, 4)	0.000000	0.000000
DIST( 4, 5)	450.0000	0.000000
DIST( 4, 6)	500.0000	0.000000
DIST( 4, 7)	2000.000	0.000000
DIST( 4, 8)	2000.000	0.000000
DIST( 4, 9)	2000.000	0.000000
DIST( 4, 10)	2000.000	0.000000
DIST( 4, 11)	2000.000	0.000000
DIST( 5, 1)	2000.000	0.000000
DIST( 5, 2)	2000.000	0.000000
DIST( 5, 3)	2000.000	0.000000
DIST( 5, 4)	450.0000	0.000000
DIST( 5, 5)	0.000000	0.000000
DIST( 5, 6)	2000.000	0.000000
DIST( 5, 7)	500.0000	0.000000
DIST( 5, 8)	2000.000	0.000000
DIST( 5, 9)	2000.000	0.000000
DIST( 5, 10)	2000.000	0.000000
DIST( 5, 11)	2000.000	0.000000
DIST( 6, 1)	2000.000	0.000000
DIST( 6, 2)	350.0000	0.000000
DIST( 6, 3)	350.0000	0.000000
DIST( 6, 4)	500.0000	0.000000
DIST( 6, 5)	2000.000	0.000000
DIST( 6, 6)	0.000000	0.000000
DIST( 6, 7)	2000.000	0.000000
DIST( 6, 8)	550.0000	0.000000
DIST( 6, 9)	2000.000	0.000000
DIST( 6, 10)	450.0000	0.000000
DIST( 6, 11)	2000.000	0.000000
DIST( 7, 1)	2000.000	0.000000
DIST( 7, 2)	2000.000	0.000000
DIST( 7, 3)	2000.000	0.000000
DIST( 7, 4)	2000.000	0.000000
DIST( 7, 5)	500.0000	0.000000
DIST( 7, 6)	2000.000	0.000000
DIST( 7, 7)	0.000000	0.000000
DIST( 7, 8)	650.0000	0.000000
DIST( 7, 9)	2000.000	0.000000

DIST( 7, 10)	2000.000	0.000000
DIST( 7, 11)	2000.000	0.000000
DIST( 8, 1)	2000.000	0.000000
DIST( 8, 2)	2000.000	0.000000
DIST( 8, 3)	2000.000	0.000000
DIST( 8, 4)	2000.000	0.000000
DIST( 8, 5)	2000.000	0.000000
DIST( 8, 6)	550.0000	0.000000
DIST( 8, 7)	650.0000	0.000000
DIST( 8, 8)	0.000000	0.000000
DIST( 8, 9)	400.0000	0.000000
DIST( 8, 10)	2000.000	0.000000
DIST( 8, 11)	2000.000	0.000000
DIST( 9, 1)	2000.000	0.000000
DIST( 9, 2)	2000.000	0.000000
DIST( 9, 3)	2000.000	0.000000
DIST( 9, 4)	2000.000	0.000000
DIST( 9, 5)	2000.000	0.000000
DIST( 9, 6)	2000.000	0.000000
DIST( 9, 7)	2000.000	0.000000
DIST( 9, 8)	400.0000	0.000000
DIST( 9, 9)	0.000000	0.000000
DIST( 9, 10)	450.0000	0.000000
DIST( 9, 11)	2000.000	0.000000
DIST( 10, 1)	2000.000	0.000000
DIST( 10, 2)	2000.000	0.000000
DIST( 10, 3)	2000.000	0.000000
DIST( 10, 4)	2000.000	0.000000
DIST( 10, 5)	2000.000	0.000000
DIST( 10, 6)	450.0000	0.000000
DIST( 10, 7)	2000.000	0.000000
DIST( 10, 8)	2000.000	0.000000
DIST( 10, 9)	450.0000	0.000000
DIST( 10, 10)	0.000000	0.000000
DIST( 10, 11)	350.0000	0.000000
DIST( 11, 1)	2000.000	0.000000
DIST( 11, 2)	250.0000	0.000000
DIST( 11, 3)	2000.000	0.000000
DIST( 11, 4)	2000.000	0.000000
DIST( 11, 5)	2000.000	0.000000
DIST( 11, 6)	2000.000	0.000000
DIST( 11, 7)	2000.000	0.000000
DIST( 11, 8)	2000.000	0.000000
DIST( 11, 9)	2000.000	0.000000



DIST( 11, 10)	350.0000	0.000000
DIST( 11, 11)	0.000000	0.000000
X( 1, 1)	0.000000	0.000000
X( 1, 2)	0.000000	300.0000
X( 1, 3)	1.000000	400.0000
X( 1, 4)	0.000000	2000.000
X( 1, 5)	0.000000	2000.000
X( 1, 6)	0.000000	2000.000
X( 1, 7)	0.000000	2000.000
X( 1, 8)	0.000000	2000.000
X( 1, 9)	0.000000	2000.000
X( 1, 10)	0.000000	2000.000
X( 1, 11)	0.000000	2000.000
X( 2, 1)	1.000000	300.0000
X( 2, 2)	0.000000	0.000000
X( 2, 3)	0.000000	300.0000
X( 2, 4)	0.000000	2000.000
X( 2, 5)	0.000000	2000.000
X( 2, 6)	0.000000	350.0000
X( 2, 7)	0.000000	2000.000
X( 2, 8)	0.000000	2000.000
X( 2, 9)	0.000000	2000.000
X( 2, 10)	0.000000	2000.000
X( 2, 11)	0.000000	250.0000
X( 3, 1)	0.000000	400.0000
X( 3, 2)	0.000000	300.0000
X( 3, 3)	0.000000	0.000000
X( 3, 4)	0.000000	300.0000
X( 3, 5)	0.000000	2000.000
X( 3, 6)	1.000000	350.0000
X( 3, 7)	0.000000	2000.000
X( 3, 8)	0.000000	2000.000
X( 3, 9)	0.000000	2000.000
X( 3, 10)	0.000000	2000.000
X( 3, 11)	0.000000	2000.000
X( 4, 1)	0.000000	2000.000
X( 4, 2)	0.000000	2000.000
X( 4, 3)	0.000000	300.0000
X( 4, 4)	0.000000	0.000000
X( 4, 5)	1.000000	450.0000
X( 4, 6)	0.000000	500.0000
X( 4, 7)	0.000000	2000.000
X( 4, 8)	0.000000	2000.000
X( 4, 9)	0.000000	2000.000

X( 4, 10)	0.000000	2000.000
X( 4, 11)	0.000000	2000.000
X( 5, 1)	0.000000	2000.000
X( 5, 2)	0.000000	2000.000
X( 5, 3)	0.000000	2000.000
X( 5, 4)	0.000000	450.0000
X( 5, 5)	0.000000	0.000000
X( 5, 6)	0.000000	2000.000
X( 5, 7)	1.000000	500.0000
X( 5, 8)	0.000000	2000.000
X( 5, 9)	0.000000	2000.000
X( 5, 10)	0.000000	2000.000
X( 5, 11)	0.000000	2000.000
X( 6, 1)	0.000000	2000.000
X( 6, 2)	0.000000	350.0000
X( 6, 3)	0.000000	350.0000
X( 6, 4)	1.000000	500.0000
X( 6, 5)	0.000000	2000.000
X( 6, 6)	0.000000	0.000000
X( 6, 7)	0.000000	2000.000
X( 6, 8)	0.000000	550.0000
X( 6, 9)	0.000000	2000.000
X( 6, 10)	0.000000	450.0000
X( 6, 11)	0.000000	2000.000
X( 7, 1)	0.000000	2000.000
X( 7, 2)	0.000000	2000.000
X( 7, 3)	0.000000	2000.000
X( 7, 4)	0.000000	2000.000
X( 7, 5)	0.000000	500.0000
X( 7, 6)	0.000000	2000.000
X( 7, 7)	0.000000	0.000000
X( 7, 8)	1.000000	650.0000
X( 7, 9)	0.000000	2000.000
X( 7, 10)	0.000000	2000.000
X( 7, 11)	0.000000	2000.000
X( 8, 1)	0.000000	2000.000
X( 8, 2)	0.000000	2000.000
X( 8, 3)	0.000000	2000.000
X( 8, 4)	0.000000	2000.000
X( 8, 5)	0.000000	2000.000
X( 8, 6)	0.000000	550.0000
X( 8, 7)	0.000000	650.0000
X( 8, 8)	0.000000	0.000000
X( 8, 9)	1.000000	400.0000

X( 8, 10)	0.000000	2000.000
X( 8, 11)	0.000000	2000.000
X( 9, 1)	0.000000	2000.000
X( 9, 2)	0.000000	2000.000
X( 9, 3)	0.000000	2000.000
X( 9, 4)	0.000000	2000.000
X( 9, 5)	0.000000	2000.000
X( 9, 6)	0.000000	2000.000
X( 9, 7)	0.000000	2000.000
X( 9, 8)	0.000000	400.0000
X( 9, 9)	0.000000	0.000000
X( 9, 10)	1.000000	450.0000
X( 9, 11)	0.000000	2000.000
X( 10, 1)	0.000000	2000.000
X( 10, 2)	0.000000	2000.000
X( 10, 3)	0.000000	2000.000
X( 10, 4)	0.000000	2000.000
X( 10, 5)	0.000000	2000.000
X( 10, 6)	0.000000	450.0000
X( 10, 7)	0.000000	2000.000
X( 10, 8)	0.000000	2000.000
X( 10, 9)	0.000000	450.0000
X( 10, 10)	0.000000	0.000000
X( 10, 11)	1.000000	350.0000
X( 11, 1)	0.000000	2000.000
X( 11, 2)	1.000000	250.0000
X( 11, 3)	0.000000	2000.000
X( 11, 4)	0.000000	2000.000
X( 11, 5)	0.000000	2000.000
X( 11, 6)	0.000000	2000.000
X( 11, 7)	0.000000	2000.000
X( 11, 8)	0.000000	2000.000
X( 11, 9)	0.000000	2000.000
X( 11, 10)	0.000000	350.0000
X( 11, 11)	0.000000	0.000000

Row	Slack or Surplus	Dual Price
1	0.000000	0.000000
2	4600.000	-1.000000
3	0.000000	0.000000
4	0.000000	0.000000
5	0.000000	0.000000
6	0.000000	0.000000
7	0.000000	0.000000

8	0.000000	0.000000
9	0.000000	0.000000
10	0.000000	0.000000
11	0.000000	0.000000
12	0.000000	0.000000
13	0.000000	0.000000
14	0.000000	0.000000
15	0.000000	0.000000
16	0.000000	0.000000
17	0.000000	0.000000
18	0.000000	0.000000
19	0.000000	0.000000
20	0.000000	0.000000
21	0.000000	0.000000
22	0.000000	0.000000
23	0.000000	0.000000
24	0.000000	0.000000
25	1.000000	0.000000
26	3.000000	0.000000
27	4.000000	0.000000
28	2.000000	0.000000
29	5.000000	0.000000
30	6.000000	0.000000
31	7.000000	0.000000
32	8.000000	0.000000
33	9.000000	0.000000
34	0.000000	0.000000
35	19.00000	0.000000
36	12.00000	0.000000
37	13.00000	0.000000
38	0.000000	0.000000
39	14.00000	0.000000
40	15.00000	0.000000
41	16.00000	0.000000
42	17.00000	0.000000
43	18.00000	0.000000
44	9.000000	0.000000
45	17.00000	0.000000
46	8.000000	0.000000
47	0.000000	0.000000
48	9.000000	0.000000
49	12.00000	0.000000
50	13.00000	0.000000
51	14.00000	0.000000

52	15.00000	0.000000
53	16.00000	0.000000
54	7.000000	0.000000
55	16.00000	0.000000
56	7.000000	0.000000
57	9.000000	0.000000
58	8.000000	0.000000
59	0.000000	0.000000
60	12.00000	0.000000
61	13.00000	0.000000
62	14.00000	0.000000
63	15.00000	0.000000
64	6.000000	0.000000
65	18.00000	0.000000
66	9.000000	0.000000
67	0.000000	0.000000
68	12.00000	0.000000
69	13.00000	0.000000
70	14.00000	0.000000
71	15.00000	0.000000
72	16.00000	0.000000
73	17.00000	0.000000
74	8.000000	0.000000
75	15.00000	0.000000
76	6.000000	0.000000
77	8.000000	0.000000
78	9.000000	0.000000
79	7.000000	0.000000
80	0.000000	0.000000
81	12.00000	0.000000
82	13.00000	0.000000
83	14.00000	0.000000
84	5.000000	0.000000
85	14.00000	0.000000
86	5.000000	0.000000
87	7.000000	0.000000
88	8.000000	0.000000
89	6.000000	0.000000
90	9.000000	0.000000
91	0.000000	0.000000
92	12.00000	0.000000
93	13.00000	0.000000
94	4.000000	0.000000
95	13.00000	0.000000

96	4.000000	0.000000
97	6.000000	0.000000
98	7.000000	0.000000
99	5.000000	0.000000
100	8.000000	0.000000
101	9.000000	0.000000
102	0.000000	0.000000
103	12.000000	0.000000
104	3.000000	0.000000
105	12.000000	0.000000
106	3.000000	0.000000
107	5.000000	0.000000
108	6.000000	0.000000
109	4.000000	0.000000
110	7.000000	0.000000
111	8.000000	0.000000
112	9.000000	0.000000
113	0.000000	0.000000
114	2.000000	0.000000
115	0.000000	0.000000
116	2.000000	0.000000
117	4.000000	0.000000
118	5.000000	0.000000
119	3.000000	0.000000
120	6.000000	0.000000
121	7.000000	0.000000
122	8.000000	0.000000
123	9.000000	0.000000
124	1.000000	0.000000