

2016 同济大学大学生数学建模竞赛

承 诺 书

我们仔细阅读了同济大学数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

学校：同济大学

队员信息：

基于引力模型的游乐园客流疏导方案

目录

| | | |
|-------|--------------------------|----|
| 1 | 问题的重述 | 4 |
| 1.1 | 游乐园现状 | 4 |
| 1.2 | 问题提出 | 4 |
| 2 | 问题分析 | 4 |
| 2.1 | 问题 1——游览提醒和疏导 | 4 |
| 2.2 | 问题 2——酒店预定房间预测 | 5 |
| 3 | 模型一的建立与求解 | 5 |
| 3.1 | 问题 1 假设 | 5 |
| 3.2 | 问题 1 的模型建立与求解 | 5 |
| 3.2.1 | 用户体验度指标 | 5 |
| 3.2.2 | 各项目点最优排队人数 | 7 |
| 3.3 | 引力模型引入 | 9 |
| 3.3.1 | 分流需求点与其余点间的引力值 | 9 |
| 3.3.2 | 项目点间相互作用强度 | 10 |
| 4 | 疏导模型检验 | 11 |
| 5 | 问题一进一步讨论 | 13 |
| 6 | 模型二的建立与求解 | 13 |
| 6.1 | 数据筛选 | 13 |
| 6.2 | 数据平稳性检验 | 14 |
| 6.3 | 模型识别 | 17 |
| 6.4 | 模型定阶 | 18 |
| 6.5 | 模型初步求解 | 19 |

| | | |
|-------|------------------------------|----|
| 6.6 | 模型适应性检验 | 20 |
| 6.7 | 单纯使用 ARMA 模型的缺点 | 20 |
| 6.8 | 基于 BP 神经网络方法修正时间序列 | 20 |
| 6.8.1 | 变量引入 | 20 |
| 6.8.2 | 神经网络的使用 | 20 |
| 6.8.3 | ARMA 与神经网络结合后预测 | 21 |
| 7 | | 22 |
| A | 附录 | 22 |
| A.1 | 问题二预测结果 | 22 |
| A.2 | 问题一代码实现 | 27 |

摘 要

Youth 游乐园即将盛大开园，我们在模型一中使用排队论模型和引力模型，对即将盛大开园的 Youth 游乐园的等候游客进行游览提醒和疏导，尽量提高每一个顾客的游园体验。在第一个模型中，确定出了一个完善的动态模型将所有的项目点动态的联系在一起。并且，在模型二中，我们基于时间序列理论中的 ARIMA 模型，以及构建的神经网络，通过分析、处理游乐园内的皇冠假日酒店 2015 年数据，帮助酒店预测 2016 年 1 月至 3 月每天预定房间数，并通过模型检验，考察模型方案的合理性，基本实现功能需求。

针对问题一，我们考虑了两个最优，游客体验最优和疏导最优。对于游客体验最优，我们使用了基于马尔科夫决策论中的排队论模型，应用了一般动态到达控制的思想，通过对公式的改写，从理论上使得每个游客的体验度达到了理想数值；对于疏导最优，我们基于时空分流引入了引力模型，根据牛顿万有引力定律， F 正比于 $(m_1 \times m_2) / r^2$ ，由此我们建立了负载疏导引力模型 $F \propto \frac{R_i^{\alpha_1} \times K_j^{\alpha_2}}{Z_{ij}^{\alpha_3}}$ ，最后通过代码模拟，用图示的方式呈现除了疏导模型的较优性。

针对问题二，对酒店预订房间的预测，本文引入自回归滑动平均模型（ARMA 模型）以描述原始数据在时间上的依存关系的延续性和受系统与自身扰动的影响。但是由于 ARMA 模型没有结合其他数据或时间序列，所以对某些特殊时间的描述较差，对此本文引入 BP 神经网络的工具把 ARMA 模型和季节，假期等因素有机结合起来，共同修正最后的预测结果。模型结合 ARMA 和神经网络的特点，即 ARMA 的时间序列描述能力与神经网络的任意非线性拟合和随机性扰动的描述能力相结合，以更好预测未来的时间序列的变化。

关键字: 排队论 引力模型 时空序列 游客疏导 数据预测

1 问题的重述

1.1 游乐园现状

国内游乐场所的旅游服务能力与游客需求无法达到完全匹配的程度，特别是山车的游乐园。固定单一的观光线路图，会导致在客流量大的情况下，队伍堵塞，带来安全隐患，降低游客的游园体验。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。

1.2 问题提出

1、Youth 游乐园即将开园，是作为本市建有最多过山车的游乐园，受到了青少年的热捧，预计将迎来每天 1 万的大客流。试就园区的整体规划，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。

2、根据游乐园内的已开业的皇冠假日酒店，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, 预测 2016 年 1 月至 3 月每天预定房间数。

2 问题分析

2.1 问题 1——游览提醒和疏导

该问题要求根据合理的游园体验度指标，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，提高整体的游园体验。在同一个游乐园中，衡量游园体验的标准直观上是由游玩项目的个数、每个项目游玩时间、排队等待时间和路上行走的时间组成。这里，考虑到假设中刚开业的游乐园，每个游客都希望体验完全部项目，并且每个项目的游玩时间又为衡定值，所以我们只考虑与游客体验负相关的等待因素和路程因素。[5]

我们分别通过这两种因素的均值、方差，构建出热情耗散 C 和等待不平衡度 U 这 2 个准则。通过对这 2 个准则分析，建立合理的函数方程，得到合理的游园体验度指标。

在提醒和疏导模型上，我们进行了如下探讨：

首先，在项目点还没有产生负荷情况时时，我们不需要对游客进行提醒和疏导，所有的游客按照个人意愿选择游览顺序。

其次,如果出现负荷情况,我们通过排队论理论,确定出每个项目点的最优排队人数, $\zeta_n(r)$, 判断如果某个项目点的总人数 $x > \zeta_n(r)$, 则游客被拒绝进入该项目点, 疏导至其他地方, 如果 $x \leq \zeta_n(r)$, 则游客被允许进入此项目点队列之中。

最后, 我们利用引力模型, 考虑到每个时段的实时空间负荷, 以及项目点之间的阻尼系数, 判断是否有必要对这 10 个点中的某些负荷大的点进行分流, 引入引力模型。通过对计算景点之间的引力值, 推算出最大引力值, 完成判断, 分配, 更新过程。

2.2 问题 2——酒店预定房间预测

该问题要求根据该酒店根据酒店 2015 年全年预定数据信息, 综合考虑影响房间预定的主要因素 (比如季节, 工作日/周末, 法定假日, 暑期等) 建立数学模型。

针对酒店的时间序列数据形式, 我们引入自回归滑动平均模型 (ARMA 模型) 以描述原始数据在时间上的依存关系的延续性和受系统与自身扰动的影响。

但是由于 ARMA 模型没有结合其他数据或时间序列, 所以对某些特殊时间的描述较差, 我们考虑引入 BP 神经网络的工具把 ARMA 模型和季节, 假期等因素有机结合起来, 共同修正预测, 完成调节偏差。

3 模型一的建立与求解

3.1 问题 1 假设

- 假设 1: 每个游客都体验完全部项目
- 假设 2: 在出现拥堵情况时, 所有游客均按照指定的提醒和疏导方案执行

3.2 问题 1 的模型建立与求解

考虑到要达到游客游园体验最优的效果, 首先要建立合理的用户体验指标, 之后要建立有效的模型。游乐场中总共只有 $A - J10$ 个项目点, 所以没有必要将这些点进行聚类分析, 可以在充分利用这 10 个点的情况下, 在引力模型的基础上进行改进, 动态等待维护这 10 个点的负荷。

3.2.1 用户体验度指标

参考自 [3], 我们获得了如下的用户等待时间与情绪的一个曲线。

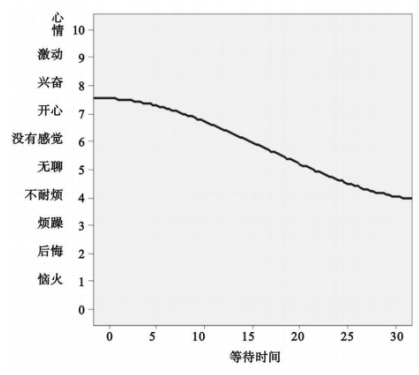


图 1: 等待曲线 [3]

通过分析，我们建立对等待时间带来的情绪的一个量化方程，变量及其产生公式如下表格。

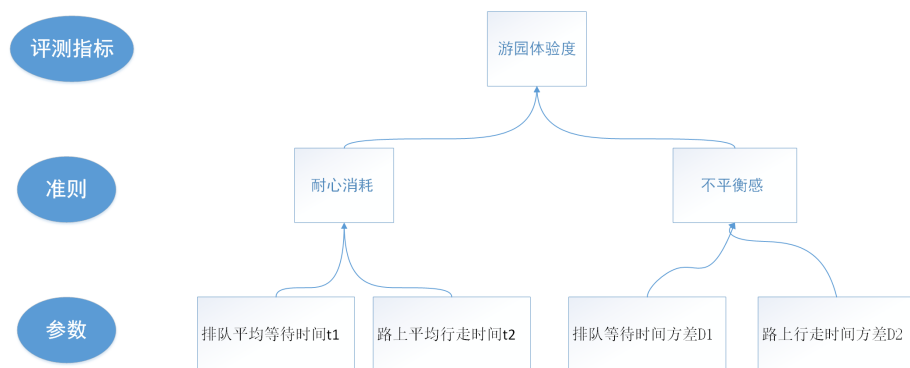


图 2: 量化指标

表 1: 变量解释及公式

| 变量符号 | 含义解释 | 公式 |
|------|----------|--|
| T1 | 排队平均等待时间 | $t_1 = \frac{\sum_{i=1}^N t_i}{N}$ |
| T2 | 路上平均行走时间 | $t_2 = \frac{\sum_{i=1}^N t_i}{M}$ |
| D1 | 排队等待时间方差 | $D_1 = \sum_{i=1}^N p_i \times (x_i - t_i)^2$ |
| D2 | 路上行走时间方差 | $D_2 = \sum_{i=2}^M p_i \times (x_i - t_i)^2$ |
| C | 热情耗散 | $Cost = (t_1 + t_2)^\alpha$ |
| U | 等待不平衡度 | $Unbalance = (D_1 + D_2)^\beta$ |
| r | 游园体验 | $r = \frac{3}{1 + e^{-\frac{1}{\cos t}} + e^{-\frac{1}{Unbalance}}}$ |

3.2.2 各项目点最优排队人数

• 排队论模型

依据排队系统的组成，排队系统的最优控制可分为到达控制和服务控制两大类，由于在我们模型假设中，每个项目点的服务控制的时间是恒定的，即游玩项目点的时间，所以只考虑到达控制。

在到达控制中，考虑到游客到达的动态性，即时间间隔的不确定性，最终决定使用一般动态到达，即一般的随机输入——定值输出的排队系统动态到达控制，模型介绍如下：

(1) **到达:** 到达间隔时间 t_i 相互独立，分布可与 n 有关。

(2) **系统流程:** 每一个顾客来时为系统带来的工作负荷为一个单位，即确切值为 1；在到达时刻，系统可以接受或拒绝此顾客并且假设拒绝不需要任何报酬或费用，被接受的顾客则将其带来的负荷加入系统的总量 x 中；第 n 个顾客到达进入系统可以增加体验度，此体验度为定值 r 。

(3) **输出:** 系统的潜在输出，即系统的状态是非负增量 1，即若设 0 时系统状态为 x ， $(0, t]$ 时顾客没有进入，则到 t 时系统的状态为 $x - 1$ 。

(4) **输出:** 系统的决策时刻是顾客是顾客的到达时间，定义状态 (x, r) 表示当一个顾客到达时所看到的系统的已有总负荷 x （不包括该到达顾客所携带的负荷）以及改到达顾客的报酬 r 。记 $V_n(x, r)$ 为从阶段 0 到阶段 n 的最大期望总报酬， $U_n(x)$ 表示在第 n 阶段

| 变量符号 | 含义解释 |
|--------------|---------------|
| x | 系统中的顾客数 |
| n | 系统中排队等待服务的顾客数 |
| r | 游客体验度 |
| $\zeta_n(r)$ | 项目点最优排队人数 |

表 2: 最优排队人数参数表

决策后系统的总负荷为 x 时的最小期望折扣总报酬。

(4) **排队限制:** 当某一个项目点排的队伍长度超过 $\zeta(r)$ 的时候, 则将此时再进入的游客进行疏导, 引导到其他项目景点。当某一个项目点排的队伍长度小于 $\zeta(r)$ 时, 允许此时进来的游客进行排队等候或者进入项目。根据排队论模型理论, 我们可以得到如下结果。

$$\text{最优排队人数} \begin{cases} V_n(x, r) = U_n(x) + [r - u_n(x)]^+, u_0(x) = 0 \\ U_n(x) = E \{e^{-\alpha t_n} V_{n-1}(x-1, r)\}, \alpha \in [0, 1) \\ u_n = U_n(x) - E \{U_n(x+1)\} \end{cases} \quad (1)$$

通过上述公式计算(1), 再根据折扣准则 DTMDP, 我们可知:

$$\zeta_n(r) = \sup \{x | \mu_n(x) \leq r\} \quad (2)$$

3.3 引力模型引入

我们的模型通过计算每个项目节点的负载率，节点之间的通行能力以及每两个节点之间的阻尼系数，将其作为 3 个基本解释变量类型，类比牛顿引力模型中幂函数衰减规则。模型中，我们把项目点之间的相互作用类比为引力强度，并恰当的反映出了游乐项目间的相互作用，受到距离即阻力系数影响而衰减。计算求得项目点之间的引力测度值之后，选择其中的最大节点，对人流进行新的匹配调度，确保负载率高的项目点优先获得最匹配的疏散点。

这样通过引力模型，我们的各个节点不再是孤立的，所有的节点之间存在类似引力般的影响。在考虑到节点通行能力的前提下，通过引力模型，使得在园区整个系统中，承载能力大的项目去分流游客更加拥挤，负载率更高的项目点。尽量保证各个项目点在最优排队人数范围内，减少游客等待时间，提高游客满意度 r ，实现对游客的疏导、提醒。在控制范的排队人数适当减少。

3.3.1 分流需求点与其余点间的引力值

• 负载率

通过参考表 1. 每个游乐项目的时间安排，我们可以获得每个项目每场的容纳量 c_i ，游客体验此项目节点所需要的时间 T_i ，图中 A-J 10 个项目点的 c_i 和 T_i 均为定值。每个项目点的负载率 R_i 受到项目容纳量 C_i 、项目点所容纳的人数 N_i 以及队伍中的平均等待时间 t_i 而动态影响。[4]

我们假设 t_i 为队伍中等待的平均时间， Pd_i 为预计排队人数， Re_i 为该项目点释放人数，计算公式如下：

$$\text{负载率计算公式} \begin{cases} R_i = \frac{S_i \times t_i}{C_i} \\ t_i = \frac{C_i \times T_i \times \left(1+2+3+\dots + \left\lfloor \xi(r)/c_i \right\rfloor\right)}{N_i} \\ S_i = (Pd_i - \zeta_n(r)) + (Re_i) \end{cases} \quad (3)$$

• 结点间通行能力

结合上面的讨论，我们定义了一个指标 K_{ij} 来反映从项目点 i 到项目点 j 的通行能力，

| 变量符号 | 含义解释 | 计算公式 |
|-----------|---------------|---|
| \bar{R} | 平均负载量 | $\bar{R} = \frac{1}{10} \sum_{i=1}^{10} R_i$ |
| ρ_j | 道路可承载的最大人流密度 | $\rho_j = \left\lfloor \frac{L_{ij} \times w_i}{D^2} \right\rfloor$ |
| w_i | 道路宽度 | 假设为定值 5m |
| v_s | 行人移动速度 | 1.5m/s |
| D | 道路上行人间的安全距离 | 2.5m |
| L_{ij} | 道路长度 | 参考自表格数据 |
| R_i | 项目点 i 的负载率 | $R_i = \frac{S_i \times t_i}{C_i}$ |
| S_i | 释放人数和拒绝排队人数之和 | $S_i = (Pd_i - \zeta_n(r)) + (Re_i)$ |
| X_{ij} | ij 道路的分流能力 | $X_{ij} = \rho_j \times w_i \times v_s$ |

表 3: 通行能力评测

K_{ij} 的定义如下: α, β 为常数

$$K_j = \alpha (\bar{R} - R_j) + \beta X_{ij} \quad (4)$$

• 点之间的阻力系数

点之间的阻力系数受到游客当前的游园体验度 r 和通过这 2 个结点 i, j 所需要的时间 t_{ij} 影响, 其中游园体验度 $r \propto (1, 3)$, r 值越大, 说明该游客的体验度越好。[2]

得到下面的公式, 其中 V_s 表示行人行走速度, $\alpha_1 \alpha_2$ 为定值。

$$\text{阻力系数} \begin{cases} t_{ij} = \frac{L_{ij}}{V_s} \\ Z_{ij} = \alpha_1 t_{ij} + \alpha_2 \frac{1}{r} \end{cases} \quad (5)$$

3.3.2 项目点间相互作用强度

牛顿模型主要用于动态测量项目点间相互作用强度, 幂函数阻抗模式可以引申到问题中的阻力系数。由重力模型 $F \propto \frac{m_1^2 \times m_2^2}{r^3}$, 我们对此进行修正。得到如下方程:

$$F \propto \frac{R_i^{\alpha_1} \times K_j^{\alpha_2}}{Z_{ij}^{\alpha_3}} \quad (6)$$

其中 α_1 α_2 α_3 为定值。

4 疏导模型检验

我们通过编程实现，对我们建立的一个完整模拟，模拟过程如下：

基于我们建立的引力模型来疏导大客流以及排队论来解决的游客体验度，我们进行了多次模型检验模拟，以下为多次模拟的其中之一：

- 第一步，我们对游客进行随机分布，为了观察方便我们以正态分布来体现游客到达后的每一个项目点的预计排队长度，然后计算出了在未进行疏导前，可能出现超负荷的项目点，包括了超出额定排队人数部分以及释放项目点人数，经模拟后计算出了负荷率，并以 RGB 的颜色调试出如下模拟图：

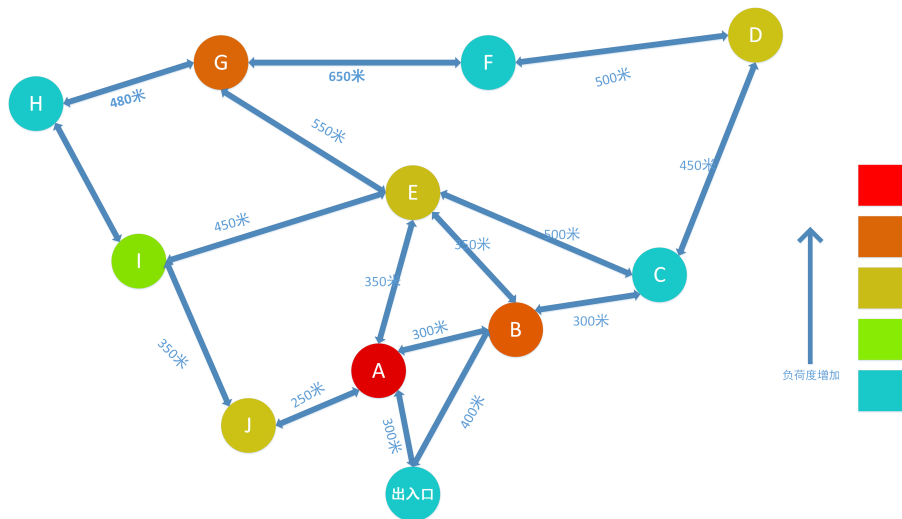


图 3: 项目点负荷情况图

- 第二步，我们将计算出的压力值和根据实时计算的道路分流能力带入引力方程，求出了每个疏散游客的疏导方向，如下图所示，疏导的依据是引力值大小来进行加权平均分配。

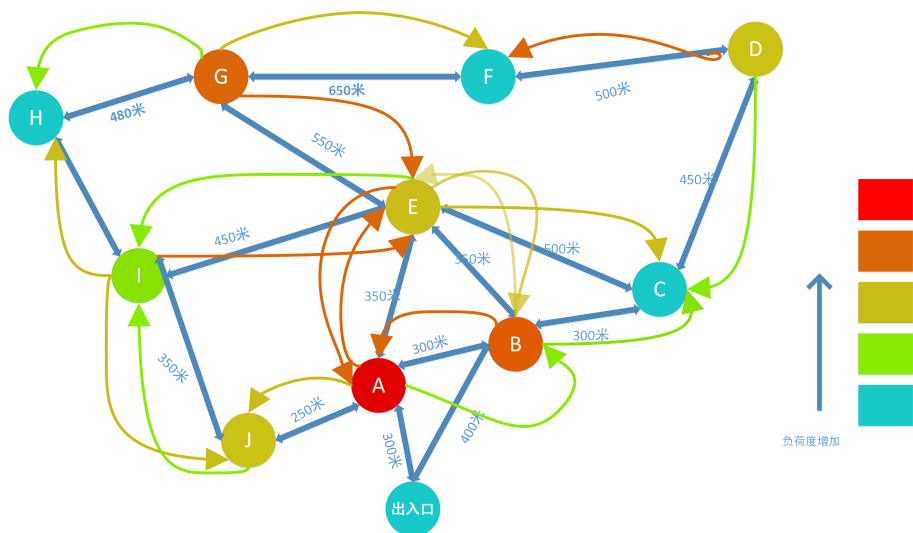


图 4: 项目点负荷情况图

- 模拟结果，经过分配人数，得到数据后进行调色，RGB 都趋于稳定在 (223, 81, 209)，即最后的紫色，如下图所示：

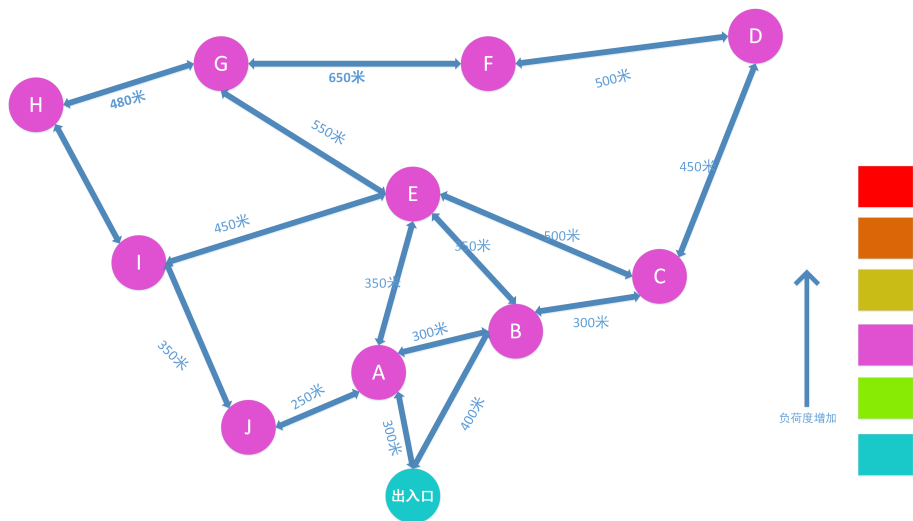


图 5: 项目点负荷情况图

- 结论：多次模拟实验结果基本都趋近于上图的结果，所以认为此模型较优。

5 问题一进一步讨论

- 由于没有真实的数据，没办法进一步模拟现实情景，只能依靠假设来模拟人流状态，比如人流速度，人流密度等，所以可能在实际应用中还需要进一步完善。
- 游客体验度的评估没有非常严谨的从心理学的角度评估，只根据自身经验，给出一些指标来模拟体验舒适度。
- 没有考虑低峰期对高峰期各方面的影响，只站在高峰期的角度模拟模型。

6 模型二的建立与求解

6.1 数据筛选

首先我们先根据附件数据表格，分析获取到 2015 年每一天住在酒店的人数 n_1 ，再加入分析季节、假日、周末因素，综合考虑，并合理选择数据，最后通过建立时空数据模型和贝叶斯预测，预测出 2016 年每一天住在酒店的人数 n_2 的结果。

- 定性分析：对于预定人次 n_2 这项数据，经过绘制它的折线图，我们发现有一个奇异点。订单数量在 4 月 1 号清明节前，是正常时间段的预定量的 7 倍左右。通过逻辑推断，我们根据此可以大致推测出酒店的入住房间量在近期也会出现增长，对比观察入住房间的折线图，直观上证明了我们的判断。

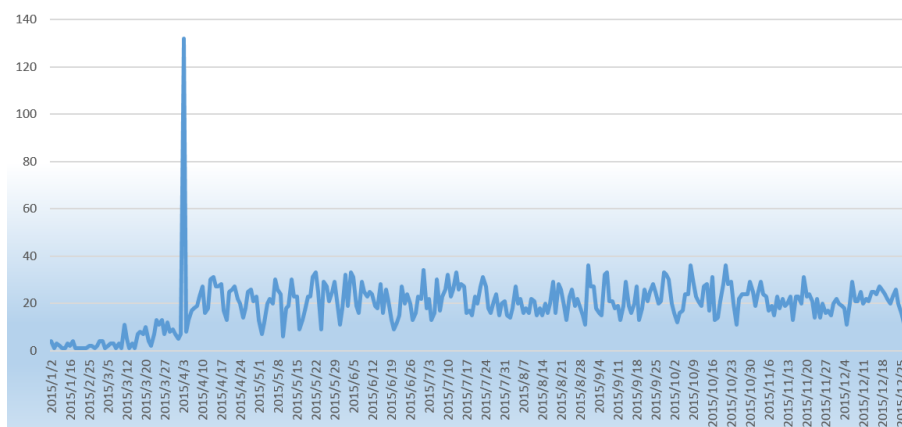


图 6: 2015 年预定时间人次

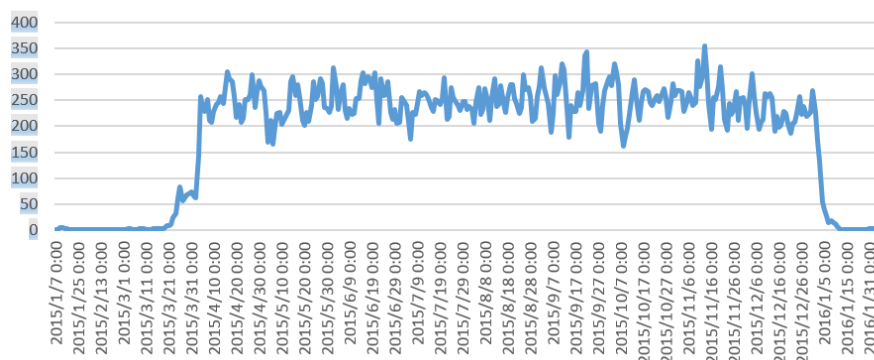


图 7: 2015 当天已住房间数

- 定量分析：通过建立相关性计算公式方程，得到入住房间与预定房间之间的相关系数，对直观预测结果进行定量说明。

$$r = \frac{\sum (k_i - \bar{k}) (j_i - \bar{j})}{\sqrt{\sum (k_i - \bar{k})^2 \sum (j_i - \bar{j})^2}}$$

按照上述公式，导入数据得到的每一天住在酒店的人数 n_1 与预定人次 n_2 之间的相关系数 r 最终为 0.590360573。之后，在数据中去掉预定人次中的奇异点后，得到新的相关系数 r 为 0.741211123。

综上所述可得，预定人次与入住人数有较强的相关性，所以，在数据分析时，考虑到简化计算，我们不将预定人次纳入模型计算。

6.2 数据平稳性检验

之后，我们对此采集到的酒店入住数据整理后进行了平稳性检验，入住时间、入住人数等组成的序列 X ，平稳性检验 ADF 检验，结果如下：

| ADF Test Statistic | 5% Critical Value |
|--------------------|-------------------|
| -1.3065 | -1.9415 |

表 4: ADF 检验结果

由上可知，时间序列具有明显的趋势性。为了将趋势性去除，从而留下相对纯粹的时间序列，方便分开处理，我们将采用 FFT 中常用的去除线性趋势性的方法。

- 最小二乘拟合

首先我们根据时间序列，进行最小二乘拟合数据，得到一个线性方程：

$$y = 0.8351x + 74.7752 \quad (8)$$

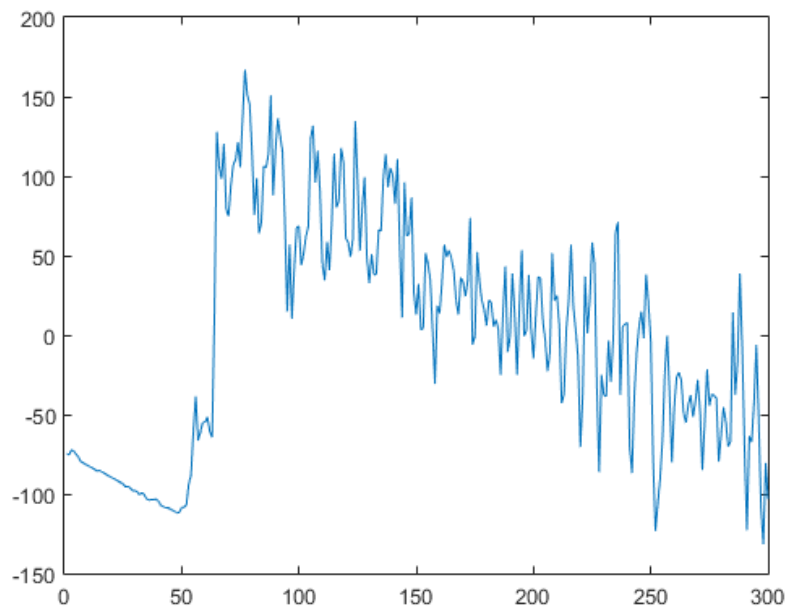


图 8: 平稳性检验

使用上式去除线性趋势后，我们得到一个新的时间序列 X_t 。

由图所示，明显发现时间序列并不稳定，因此我们对 X_t 做一阶差分后，得到 Z_t 后，我们可以得到新的图像，并对 X_t 进行 ADF 检验。

| ADF Test Statistic | 5% Critical Value |
|--------------------|-------------------|
| -4.2870 | -1.9415 |

表 5: ADF 一阶差分后检验结果

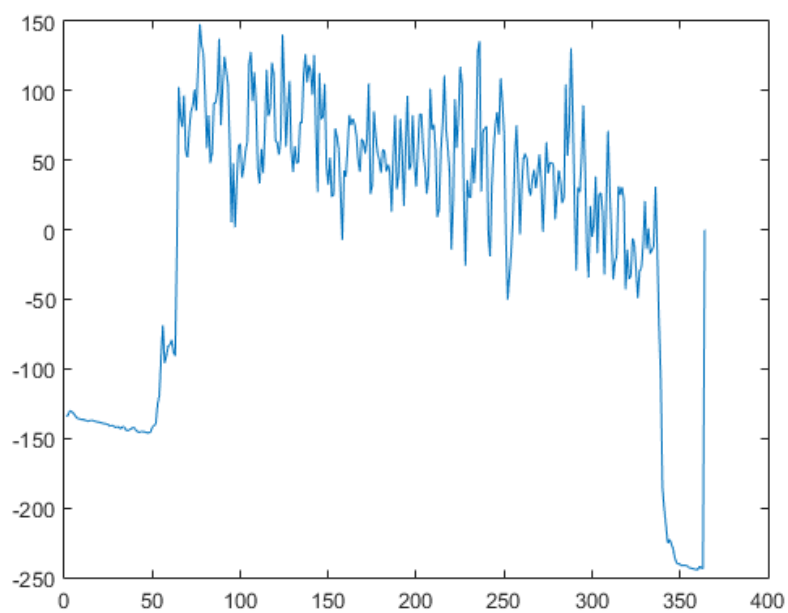


图 9: 一阶差分后

| ADF Test Statistic | Mackinnon 临界值 |
|--------------------|---------------|
| -31.0097 | -1.9415 |

表 6: Mackinnon 临界值对应的显著水平为 5%

表格中的 Mackinnon 临界值对应的显著水平为 5%，所以由此得出，序列 Z_t 为平稳序列。

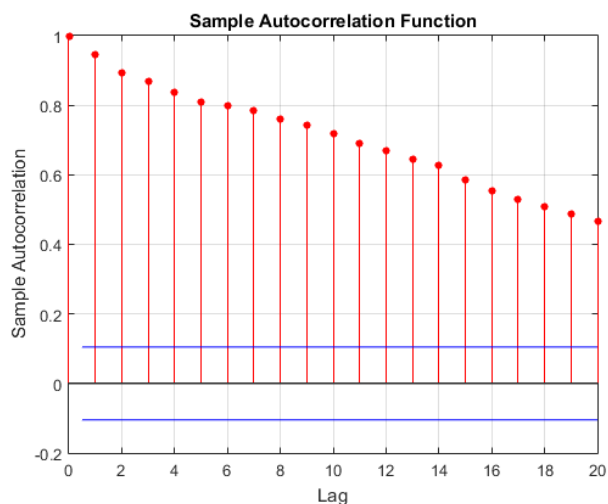


图 10: 模型识别 -1

6.3 模型识别

为了确定时序序列的模式，我们对进行稳定化后的数据序列 Z_t 将会绘制从 1 至 16 阶的自相关 ACP 图和偏自相关 $PACP$ 图。由自相关图和偏自相关图可知，自相关图和偏自相关图均有明显的拖尾性，因而可以初步判断模型为 ARMA 模型。

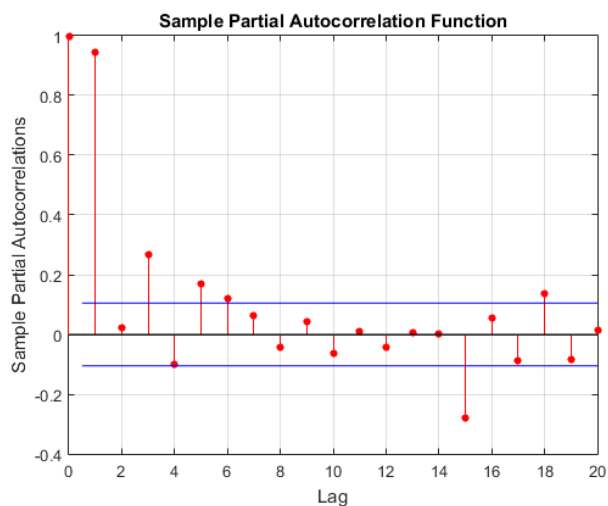


图 11: 模型识别 -2

6.4 模型定阶

- AIC 准则（赤池信息准则）

为了进一步确定 $ARIMA$ 模型，我们采用 AIC 准则（赤池信息准则）为 $ARIMA$ 模型定阶。这次仅适用 AIC 准则确定 p 和 q ，即可以简化为 $ARMA(m, n)$ 定阶。[1][6]

对于某个 $ARMA(m, n)$ 法则， AIC 准则函数可表示为：

$$AIC(n, m) = -2 \ln[L] + 2r \quad (9)$$

其中，(9)中的 L 为 $ARMA m n$ 模型中的极大似然值， r 为模型的独立参数个数，即 $r = m + n$ ，此处 L 可用模型残差方差的极大似然函数代替。

$$AIC(n, m) = \ln[\hat{\sigma}_a^2] + \frac{2r}{N} \quad (10)$$

使用拟合后的参数，进行 $ARMA$ 的模拟，生成时间序列 x'_t 。

差分还原后，加上前面出去的线性趋势函数，可得到最后的时间序列。其中，红色线

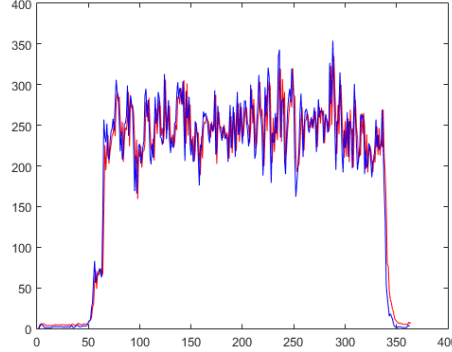


图 12: 模拟曲线

条为模型预测的曲线，蓝色为真实情况下的曲线，由图示可知，我们可以近似得出，时间序列拟合的结果基本符合需求。

其中(9)， N 为样本大小，在 AIC 实际求取中，为了求取方便，我们这里将使用最小二乘估计后所得残差方差近似取得。

AIC 准则中，前半部分 $\ln[\hat{\sigma}_a^2]$ ，用于反映模型拟合的好坏，而后者 $\frac{2r}{N}$ ，部分表明模型参数的多少。分析可得，随着模型阶数上升， $\ln[\hat{\sigma}_a^2]$ 将随之快速下降。但当达到某一阶数

时，随着模型的增高，残差方差下降得较慢，这是 AIC 的变化将主要受 $\frac{2x}{N}$ 的影响，而随着上升。

- 分析结果

由上分析可得，ARMA 取最优值时为 AIC 值最小时，所以，在在某个 (M, N) 范围内，最优的模型为 n_0, m_0 ，结论如下：

$$AIC(n_0, m_0) = \min_{1 \leq n, m \leq M, N} AIC(n, m) \quad (11)$$

上图为对于 x, y 轴为 (p, q) ， z 轴为 AIC 值。

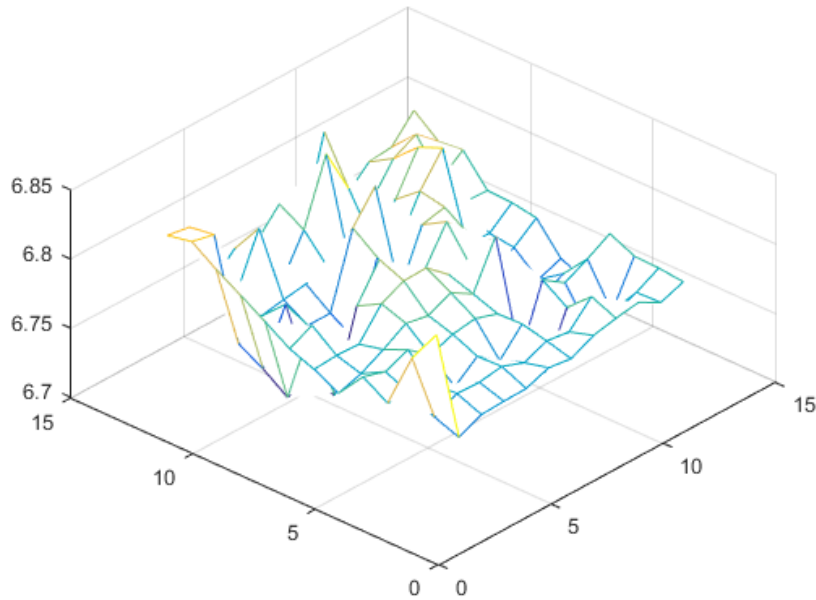


图 13: 模型定阶 -3d

经过检验可得， AIC 在 $(1, 2)$ 处取最小值。由此我们可以确定函数为 $ARMA(1, 2)$ 模型。

6.5 模型初步求解

我们根据最小二乘法进行参数模拟后，可以得到(12)。

$$Z_t - 0.9914Z_{t-1} = a_t - 0.0345a_{t-1} - 0.3531a_{t-2} \quad (12)$$

6.6 模型适应性检验

适应性检验，指的是一个时间序列模型解释动态性的程度，即判断 ARMA 模型用于描述该事件序列是否恰当。我们这里的适应性检验为检验序列是否是白噪声。

使用自相关法可以看出，自相关函数基本符合要求，即 $\hat{\rho} \leq 1.96/\sqrt{N}$

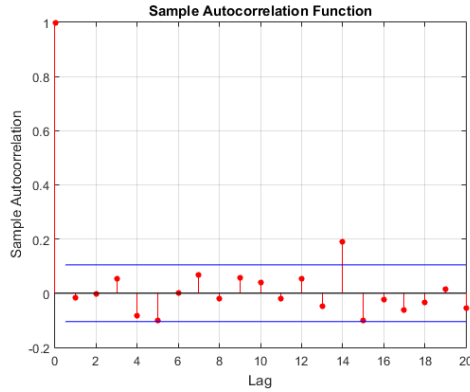


图 14: ACP

6.7 单纯使用 ARMA 模型的缺点

在上述模型之中，我们紧紧利用了 ARMA 模型分析时间序列中系统扰动和系统的延后影响，得到的规律，但是现实生活中，我们知道，某天是否是旅游人数往往与法定假日，学生放假，季节等因素有着不可分割的关系，即我们可以得知某天旅游人数与这天的固有属性有着不可分割的关系，但是这些关系都是 ARMA 中无法表示的，我们将无法表示时间序列与这些特别的天数之间的关系。

6.8 基于 BP 神经网络方法修正时间序列

由于季节，假日等时间因素与时间序列之间的关系，极有可能为一个非线性的函数，我们引入 BP 神经网络对 ARMA 模型所得到的结果进行修正。我们引入神经网络来描述神经网络来描述时间序列与季节，法定假日等的关系。

6.8.1 变量引入

6.8.2 神经网络的使用

- 训练

对于第 k 天的数据的训练，BP 神经网络的输入为引入的 6 个布尔变量，以及第 k 天和 $k-1$ 天的 ARMA 模型预测值，输出为第 k 天 ARMA 预测值与真实值的差，即残差。

经过反复训练，我们选择 8 个输入，18 个中间节点，1 个输出的神经网络拓扑结果。训练迭代次数为 150 次。

为了说明我们的神经网络是有效的，我们仅适用 363 天中前的 300 天进行神经网络训练，后 63 天进行预测，检验神经网络的准确度。

• 训练结果

训练结果为后 63 天的残差方差由 54740 降至 27260，模型有了明显的改善。

6.8.3 ARMA 与神经网络结合后预测

使用 ARMA 预测后，加上通过神经网络算出的残差修正值，可以得到最后我们对 2016 年 1-3 月的预订房间预测。（预测见附表）

参考文献

- [1] 姜明. 网络流量预测中的时间序列模型比较研究. 电子学报, 37(11):2353–2358, 2009.
- [2] 张影莎. 基于排队论的方特欢乐世界主题公园容量研究. 旅游学刊, 27(1):66–72, 2012.
- [3] 吴茜媛. 一种 e-learning 用户心理体验质量的定量分析方法. 2012.

| 变量名称 | 真值 | 对应含义 |
|----------------|------|-------------|
| <i>weekday</i> | true | 为这天为星期六或星期天 |
| <i>holiday</i> | true | 为这天为假期 |
| <i>spring</i> | true | 为这天是否在春天 |
| <i>summer</i> | true | 为这天是否在夏天 |
| <i>autumn</i> | true | 为这天是否在秋天 |
| <i>winter</i> | true | 为这天是否在冬天 |

表 7: 6 个布尔型的变量

- [4] 肖雄辉. 旅游高峰期景区游客引力分流调度模型研究——以九寨沟风景区为例. 旅游科学, 27(6), 2013.
- [5] 吴义宏. 主题公园的拥挤感知影响因素和使用水平研究——以方特欢乐世界为例. PhD thesis, 安徽师范大学, 2014.
- [6] 廖民生. 基于模糊时间序列模型预测三亚旅游总收入. 经济研究导刊, (4):237-242, 2015.

7

A 附录

A.1 问题二预测结果

| 2016 年 1 月至 3 月 | 每天预定房间数 |
|-----------------|---------|
| 2016/1/1 | 211 |
| 2016/1/2 | 180 |
| 2016/1/3 | 137 |
| 2016/1/4 | 67 |
| 2016/1/5 | 55 |
| 2016/1/6 | 41 |
| 2016/1/7 | 32 |
| 2016/1/8 | 31 |
| 2016/1/9 | 28 |
| 2016/1/10 | 20 |
| 2016/1/11 | 13 |
| 2016/1/12 | 9 |
| 2016/1/13 | 8 |
| 2016/1/14 | 7 |
| 2016/1/15 | 5 |
| 2016/1/16 | 6 |
| 2016/1/17 | 5 |
| 2016/1/18 | 6 |
| 2016/1/19 | 4 |
| 2016/1/20 | 5 |

表 8: data table

| 2016 年 1 月至 3 月 | 每天预定房间数 |
|-----------------|---------|
| 2016/1/21 | 4 |
| 2016/1/22 | 5 |
| 2016/1/23 | 4 |
| 2016/1/24 | 4 |
| 2016/1/25 | 7 |
| 2016/1/26 | 5 |
| 2016/1/27 | 7 |
| 2016/1/28 | 6 |
| 2016/1/29 | 7 |
| 2016/1/30 | 5 |
| 2016/1/31 | 3 |
| 2016/2/1 | 2 |
| 2016/2/2 | 4 |
| 2016/2/3 | 3 |
| 2016/2/4 | 2 |
| 2016/2/5 | 5 |
| 2016/2/6 | 4 |
| 2016/2/7 | 5 |
| 2016/2/8 | 6 |
| 2016/2/9 | 4 |
| 2016/2/10 | 3 |
| 2016/2/11 | 4 |
| 2016/2/12 | 4 |
| 2016/2/13 | 5 |
| 2016/2/14 | 5 |
| 2016/2/15 | 5 |

表 9: data table

| 2016 年 1 月至 3 月 | 每天预定房间数 |
|-----------------|---------|
| 2016/2/16 | 6 |
| 2016/2/17 | 6 |
| 2016/2/18 | 5 |
| 2016/2/19 | 6 |
| 2016/2/20 | 7 |
| 2016/2/21 | 7 |
| 2016/2/22 | 7 |
| 2016/2/23 | 8 |
| 2016/2/24 | 8 |
| 2016/2/25 | 8 |
| 2016/2/26 | 9 |
| 2016/2/27 | 9 |
| 2016/2/28 | 9 |
| 2016/2/29 | 10 |
| 2016/3/1 | 10 |
| 2016/3/2 | 10 |
| 2016/3/3 | 10 |
| 2016/3/4 | 11 |
| 2016/3/5 | 12 |
| 2016/3/6 | 13 |
| 2016/3/7 | 34 |
| 2016/3/8 | 45 |

表 10: data table

| 2016 年 1 月至 3 月 | 每天预定房间数 |
|-----------------|---------|
| 2016/3/9 | 66 |
| 2016/3/10 | 78 |
| 2016/3/11 | 94 |
| 2016/3/12 | 116 |
| 2016/3/13 | 156 |
| 2016/3/14 | 216 |
| 2016/3/15 | 237 |
| 2016/3/16 | 257 |
| 2016/3/17 | 257 |
| 2016/3/18 | 262 |
| 2016/3/19 | 336 |
| 2016/3/20 | 263 |
| 2016/3/21 | 220 |
| 2016/3/22 | 240 |
| 2016/3/24 | 249 |
| 2016/3/25 | 236 |
| 2016/3/26 | 336 |
| 2016/3/27 | 304 |
| 2016/3/28 | 259 |
| 2016/3/29 | 292 |
| 2016/3/30 | 253 |
| 2016/3/31 | 206 |

表 11: data table

A.2 问题一代码实现

动态规划，为顾客提供游园线路引导

Input python source:

```
# coding=utf-8
import random

class node:
    c = 0
    tT = 0
    R = 0
    t = 0
    PD = 0
    S = 0
    E = 0
    need = 0

C = [400,30,50,30,100,50,30,30,20,50]
T = [1980,75,150,150,300,150,120,90,90,120]
pd = [400,90,150,90,300,150,90,90,80,150]
e=[(0,1,1000),(0,9,833),(1,2,1000),(0,4,1166),(1,4,1166)
    ,(2,4,1666),(8,9,1166),(4,8,1500),(2,3,1500),(3,5,1666)
    ,(5,6,2166),(6,7,1333),(7,8,1550),(4,6,1833)]

iNode = []

for i in range(0,10):
    point = node()
    point.c = C[i]
```

```

point.tT = T[i]
point.E = pd[i]
iNode.append(point)

count = 0

while(count<5):
    sumR = 0
    for i in range(0,10):
        print str(i+1)
        iNode[i].PD = random.normalvariate(0,10)+iNode[i].E
        pdTest = iNode[i].PD
        print pdTest
        iNode[i].S = (random.normalvariate(0,10))+C[i]
        STest = iNode[i].S
        print STest
        if iNode[i].S >0 and iNode[i].PD > iNode[i].E:
            iNode[i].need = 1
            temp = iNode[i].E/C[i]
            sum = 0
            for k in range(1,temp):
                sum += k
            iNode[i].t = (C[i]*T[i]*sum)/(iNode[i].PD+C[i])
            TTest = iNode[i].t
            print TTest
            iNode[i].R = (iNode[i].S*iNode[i].t)/C[i]
            RTest = iNode[i].R
            print RTest
            sumR += iNode[i].R

```

```

        print '\n'

a = 0.5
b = 0.5
avgR = sumR/10
K = []

for i in range(0,10):
    if(iNode[i].need == 1):
        Y = []
        for (u,v,d) in e:
            if u == i or v == i:
                if u == i:
                    x=v
                    p1 = avgR - iNode[u].R
                    O = a*p1+b*d
                else:
                    x=u
                    p1 = avgR - iNode[v].R
                    O = a*p1+b*d
            Y.append((i,x,O))
        K.append(Y)

F = []

sumYinLi = 0
    for (u,v,d) in K[m]:

```

```

X = u
Y = v
lala = type(K[m])
F.append((u,v,YinLi))
sumYinLi += YinLi

for (u,v,d) in F:
    print(str(u)+": "+str(iNode[u].S))
    print(str(v)+": "+str(iNode[v].S))
    print '\n'
    iNode[v].S += (d/sumYinLi)*iNode[u].S
    print(str(u)+": "+str(iNode[u].S-iNode[v].S))
    print(str(v)+": "+str(iNode[v].S))
    print '\n'

for i in range(0,10):
    iNode[i].need = 0
print '\n'
count = count+1

```