

评委一评分，签名及备注	队号：  1206	评委三评分，签名及备注
评委二评分，签名及备注	选题：  A	评委四评分，签名及备注

题目：火车票购票网站优化

### 摘要

本文主要研究 12306 网站出售火车票系统中目前队列处理数据的能力以及错误率，同时根据所建立数学模型对现行的网站分时策略提出优化方案，并结合现时情况关于缓解网站访问压力提出合理建议。

第一问：首先将每 1000 个数据为一组分别使用 spss 软件以获取号码次数为因变量得到频数分布表，接着构造出 10 个样本数据并计算每个样本平均每秒处理的数据量。其次使用无偏估计量估计样本整体每秒处理的数据量为 59.6550，然后在合理假设下根据附表数据计算错误率是 1.52%，最后为保证模型的可靠性使用区间估计对整体每秒处理的数据量进行正态检验。

第二问：首先根据 Aleax 网站提供的 12306 网站访问量走势图采用实际测量法估计出 1 月 17 号该网站的总访问量，接着使用 *Fibonacci* 级数原理  $F_n = F_{n-1} + F_{n-2} (n > 2), F_1 = F_2 = 1$  并在 VC6.0++ 环境下编程，模拟出各个时间段的访问量；然后根据分表数据研究得到访问者到达间隔时间分布和服务时间分布分别服从参数为  $\lambda$  和  $\mu$  的负指数分布；再次在各个时间段分别建立  $M/M/n$  模型，在 *matlab*7.0 中编程求的每一个队列模型的概率、队长、排队长、逗留时间、等待时间五个数量指标，并进一步从中抽象出符合整体的队列模型，最后计算整体队列模型的数量指标，在合理假设下列出方程求出需要 6 个队列能够满足需求。

第三问：首先对第二问模型给出简单评价并指出了在实际操作中存在的诸如网站成本，顾客消耗时间成本等问题技术建立愿望模型，接着确立了消费者购票等待时间成本的研究方向对第二问中的模型进行优化；然后根据第二问已得到的各时间段队列的数量指标重新抽象概括出总体的队列模型，并求出需要 18 个队列能满足需求。

最后，本文给出了关于解决 12306 网站拥堵问题的具体意见，如合理分配网络放票量与车站售票量、实行“查购分开”、按需生产、增加“记忆功能、精简网页内容、增加透明度，加强管理，自觉接受监督等形成建议书，供中国铁路总公司参考，希望可以进一步完善网上订票系统。

**关键字：**参数估计    *Fibonacci* 级数     $M/M/n$  模型    排队论    愿望模型

# 火车票购票网站优化

## 一、问题重述

火车，是人们出行的重要的交通工具之一。中国从 2011 年开始实行网络售票，购买火车票的方式从只能到火车票售票口排队购票，逐步发展为可通过电话订票和网站订票等多种便利的途径。网络售票给人带来便利的同时，也出现了不少问题。典型的是 2013 年春节的时候，购票网站出现了包括登录、购买、付款等各个环节的问题。为了保证网站更稳定的工作，需要你帮忙提出优化意见。为了便于对问题的分析，现对网站的订票过程做如下简化：

- 1、登录网站：根据用户名和密码登录网站；
- 2、查询余票：查找某趟列车的剩余车票；
- 3、生成订单：锁定剩余车票中的一张作为购买的车票，根据车票信息和购票人信息生成支付订单；
- 4、付款完成：车票购买完成；付款失败：车票回收继续等待预定。

经分析，网站订票的瓶颈可能是以下两方面的原因：

第一，网站并发问题，也就是同一时刻订票人数过多的问题。在同一时刻（例如 1 秒之间）访问网站的人数过多，服务器无法响应所有人的请求，导致无法登录、无法查询票车票等问题。在春节期间网站的访问量是非常大的，网站的日访问量可以参考 alexa 的记录（见参考资料）。为了缓解同一时间段内网站访问量过大的问题，网站采取了分时购票的方法，相当于分散了请求量，减轻了同一时间段并发访问过大的问题。

第二，唯一资源问题，也就是唯一的一张车票。所有来订票的人先要锁定一张车票，如果出现多人同时请求订购同一张票，那么系统就无法判断这张车票该给谁锁定，这样系统就会让请求重试。如果一直无法锁定车票，就会导致系统死锁、订票失败。为了应对车票这唯一资源的分配问题，网站又采取了排队的方法，即请求到来先获取一个号码等待排队，这样就避免了对车票直接造成死锁。获取一个号码，入队列的过程相对于处理一张车票的过程（包括锁定车票、生成订单、检测支付等一系列操作）是非常快速的。（这个过程可以和银行的操作流程对比，先获取一个号码排队等待处理，服务窗口是处理器，按照队列先进先出的顺序依次处理）。

订票网站想要利用一种队列来满足现在的订票需要，附件给出了这个队列的测试数据：一个队列的入队列数据和一个队列的出队列数据，这些数据只是测试队列性能使用的，不是预测这一时间网站请求量的依据。根据资料回答以下问题：

- 1、根据队列的数据估计该队列每秒最多能处理多少数据，错误率是多少。
- 2、根据网站日累计访问量数据（见参考资料）和现行的网站分时策略，模拟 2013 年 1 月 17 日的网站订票请求量数据，计算需要多少个队列可以满足需求（总的票数用  $N$  表示，要求入队列错误率低于 0.1%，要求队列排队的时间不超过 30 分钟）。
- 3、评价现行的分时购票策略是否合理，可以如何优化（要求分时不超过 10 个）根据优化的分时策略重新计算第 2 问的问题。
- 4、给网站提出一个建议书，可以包括其他的方面，例如预售期、退票等，不需要建模模型说明。

## 二、模型假设

1. 假定模型引用的数据真实可靠，都可以用来计算；
2. 每个测试组内的时间都是连续无间断的；
3. 假设获取号码所耗时间为顾客到达的时间；获取号码即认为已经进入队列；
4. 假设出队列时间极短，可忽略不计，则题目所给出队时间为服务时间；
5. 假设 12306 网站每增加一个售票时间点即认为是增加一个服务台。

## 三、符号说明

$\sigma^2$	总体方差
$s^2$	样本方差
$\lambda$	单位时间平均到达的顾客数
$\mu$	单位时间被服务完成的顾客数
$L_s$	系统中的顾客数
$L_q$	系统中排队等候服务的顾客数
$W_s$	一个顾客在系统停留的时间
$W_q$	一个顾客在系统中排队等候的时间
$\rho$	服务强度或话务强度

## 四、问题分析

第一问：要求根据 10 组样本数据估计总体的每秒处理数据量及错误率。首先看到数据表中存异常数据，需要对异常数据进行处理；接着根据修正后的数据，利用 spss 软件得出频数分布表；其次以获取号码次数是 1、2、3 的为服务器成功处理的数据，获取号码次数是 4 次的为数据处理失败；然后根据已得数据得到每秒处理数据量的样本量和错误率的样本量；最后使用参数估计法得到总体的平均每秒处理的数据量和平均错误率。

第二问：要求根据局 Aleax 网站提供的 12306 网站日累计访问量，模拟出各个时间段的访问量，并根据题目要求求出需要的队列数。首先在给出的参考网站上找到一月份 12306 网站访问量趋势，并从图中进得出每百万人中日均访问量；接着从国家统计局网站找到最新的国家人口总数，根据同比列原则求出在 1 月 17 日当天的总访问量；其次考虑到网站日访问量为各个时间段的累计访问量这一特点，我们利用 *Fibonacci* 级数原理并在 VC6++ 环境下编写程序估计各个时间段的访问量；然后对各个时间段使用排队模型，并用 *matlab* 编程求出系统空闲的概率、队长、排队长、逗留时间、等待时间等 5 个数量指标，找出最符合生活实

际的排队模型并推广到整个售票时间段；最后根据推广的模型及相关数量指标求出需要的队列数。

第三问：要求简要评价第二问的模型并给出模型的优化方案。首先应该对第二问已经建立的模型简要评价，并指出在实际操作可能存在的一些问题，并就其中的一个问题展开研究；确定研究顾客等待时间成本之后，由于顾客消耗时间成本无法用金钱等实物衡量，故建立“愿望模型”，即顾客希望在进入网站后能在尽可能短的时间买到票。为实现这一目标，我们使用第二问中已经计算出的五个数量指标，重新比较，进一步抽想、概括出整体的队列模型，并计算整体队列的五个数量指标，并根据该类指标和前文估计的日均总访问量计算出能够满足需求的队列个数。

本文最后根据已经建立的数学模型及相关参考资料和生活经验，给出若干解决网站拥堵的参考意见！

## 五、问题一的模型建立与求解

### 5.1 数据处理

#### 5.1.1 异常数据的处理

从所给的参考数据及附表的备注，可以看出在入队列数据表中，单元格 E15 的数据应该为空，而所给数据为：0.0373（总入队时间），这一数据远大于获取号码耗时，是一个异常数据。为与前文假设 2 相吻合以及便于下文操作，这里将异常数据 0.0375 改为空白，与附表中的其他数据保持一致。

#### 5.1.2 频数分布表的生成

根据题目所给数据，将每 1000 个数据为一组分别得到频数分布表。利用 spss 软件以获取号码次数为因变量做频数分布表，并一步处理得到如下结果：

表 5.1 第一组获取号码次数频率表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1.00	911	91.1	91.1
2.00	51	5.1	96.2
3.00	14	1.4	97.6
4.00	24	2.4	100.0
合计	1000	100.0	

表 5.2 第二组获取号码次数频率表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1.00	697	69.7	69.7
2.00	182	18.2	87.9
3.00	69	6.9	94.8
4.00	52	5.2	100.0
合计	1000	100.0	

表 5.3 第三组获取号码次数频数分布表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1	656	65.6	65.6
2	208	20.8	86.4
3	83	8.3	94.7
4	53	5.3	100.0
合计	1000	100.0	

表 5.4 第四组获取号码次数频数分布表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1	792	79.2	79.2
2	164	16.4	95.6
3	36	3.6	99.2
4	8	0.8	100.0
合计	1000	100.0	

表 5.5 第五次获取号码次数频数分布表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1	855	85.5	85.5
2	115	11.5	97.0
3	22	2.2	99.2
4	8	0.8	100.0
合计	1000	100.0	

表 5.6 第六组获取号码次数频数分布表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1	882	88.2	88.2
2	103	10.3	98.5
3	13	1.3	99.8
4	2	0.2	100.0
合计	1000	100.0	

表 5.7 第七组获取号码次数频数分布表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1	944	94.4	94.4
2	50	5.0	99.4
3	4	0.4	99.8
4	2	0.2	100.0
合计	1000	100.0	

表 5.8 第八次获取号码次数频数分布表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1	918	91.8	91.8
2	76	7.6	99.4
3	3	0.3	99.7
4	3	0.3	100.0
合计	1000	100.0	

表 5.9 第九组获取号码次数频数分布表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1	945	94.5	94.5
2	52	5.2	99.7
3	3	0.3	100.0
合计	1000	100.0	

表 5.10 第十组获取号码次数频数分布表

获取号码次数	频率	有效百分比 (%)	累积百分比 (%)
1	970	97.0	97.0
2	30	3.0	100.0
合计	1000	100.0	

### 5.1.3 样本量的生成

根据频数分布表，得到每组成功处理的数据量及每组所用的时间：

$X_0 = (976, 948, 947, 992, 992, 998, 998, 997, 1000, 1000)$

$t_i = (8.0053, 8.0067, 9.0083, 13.1553, 19.0264, 24.0365, 37.0062, 45.0247, 56.0144, 66.0147)$

根据公式  $X_1 = \frac{X_{0i}}{t_i}$  ( $t_i$  为第  $i$  组的总时间) 得到每组成功处理的数据量的样本量:

$$X_1 = (121.92, 118.40, 105.13, 75.41, 52.14, 41.47, 26.97, 22.14, 17.85, 15.12)$$

根据频数分布表得到每组处理的数据的错误率:

$$X_2 = (0.024, 0.052, 0.053, 0.008, 0.008, 0.002, 0.002, 0.003, 0.00, 0.00)$$

## 5.2 估计总体的每秒处理数据量及错误率

### 5.2.1 检验收据的正态性

此处引用 matlab 工具箱中  $h = \text{normplot}(X_1)$  命令, 此命令可以显示数据矩阵的正态概率分布图。若数据符合正态分布, 则图形显示出直线形态, 而其他概率分布函数则显示出曲线形态。运行结果如图 1 所示:

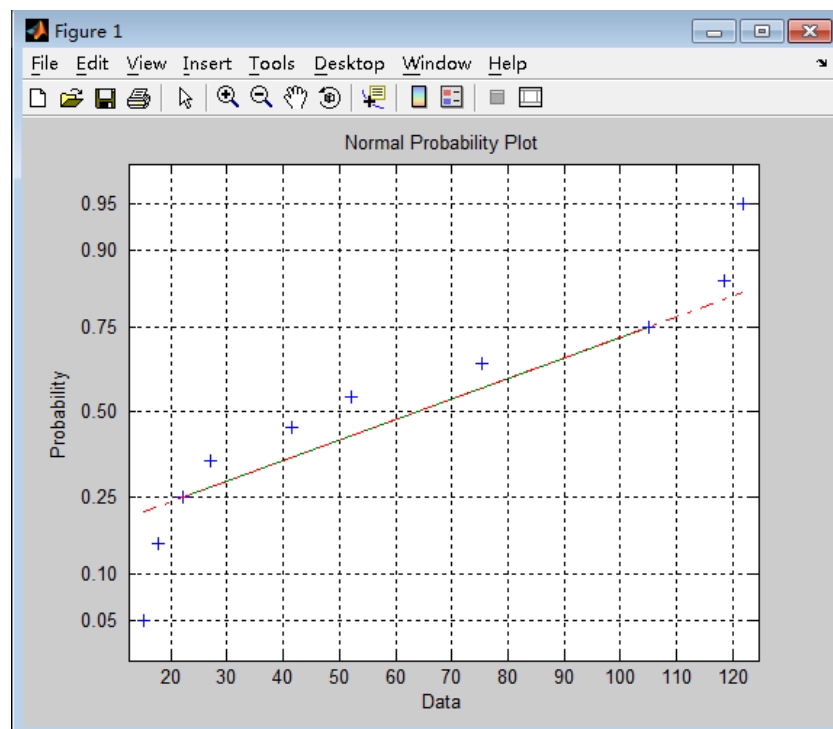


图 5.1 正态概率图

从图 1 可以看出数据分布基本为一条直线, 故可以初步断定每秒处理数据量符合正态分布, 因此可以使用正态参数估计。

同理可以得到错误率数据也符合正态分布, 故也可以使用正态估计。

### 5.2.2 总体的每秒处理数据量及错误率的估计

经检验该组数据符合正态分布, 由此可以给出该组数据的均值、标准差及置信区间。这里调用 matlab 程序中的运算指令:

$$[\text{Muhat}, \text{sigmahat}, \text{muci}, \text{sigmaci}] = \text{normfit}(X_1)$$

其中返回值 muhat 是  $X_1$  的均值的点估计值，sigmhat 是标准差的点估计值，muci 是均值的区间估计，sigmaci 是标准差的区间估计。

运行结果：每秒处理数据量的均值的点估计值为：59.6550，均值的置信区间：[29.2964,90.6406];每秒处理数据量的标准差估计值为：42.4716；标准差的置信区间为[29.2166,77.5488]。

在估计总体均值前先引入一个命题：设  $(X_1, X_2, \dots, X_n)$  是来自具有数学期望的任意总体  $X$  的一个样本，记  $E(X) = a$ ，则  $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$  是  $a$  的无偏估计。

证明：因为  $\hat{a} = \bar{X}$  的数学期望为  $E(\bar{X}) = E(\frac{1}{n} \sum_{i=1}^n X_i) = \frac{1}{n} \sum_{i=1}^n E(X_i) = \frac{1}{n} \sum_{i=1}^n a = a$ ，由无

偏估计的定义知： $\bar{X}$  是  $a$  的无偏估计。

根据上述命题，可以得出总体的均值的无偏估计为：59.6550，即服务器每秒处理的数据量是 59.6550。

同理可以得到总体错误率的均值无偏估计：0.0152，即总体的错误率是 1.52%。

### 5.3 模型的检验

从上文的计算过程可以看出样本的均值是总体均值的无偏估计，总体均值的精确度主要取决于样本均值的精度，因此只要保证样本均值的精确度就保证了总体均值的精确度。为进一步说明估计的可靠性，接下来对样本均值的可靠性进行检验。

首先计算样本  $X_1$  的均值： $\bar{X}_1 = 59.655$

本题中样本量  $n = 10 < 30$  属于小样本量；总体方差  $\sigma^2$  未知，则需要用样本方差  $s^2$  代替  $\sigma^2$ ，这时，样本均值经过标准化以后的随机变量则服从自由度为  $(n-1)$  的  $t$  分布即：

$$t = \frac{\bar{x} - \mu}{s / \sqrt{n}} \sim t(n-1)$$

可以得到总体均值  $\mu$  的置信区间：，其中  $s$  是样本的方差，计算公式： $s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$ ，

在 95% 的置信区间下，查表  $t_{\frac{\alpha}{2}}(10-1) = 2.2622$ ，计算总体均值  $\mu$  的置信区间为：



[29.2688,90.0415]。与前文计算的置信区间[29.2964,90.6406]误差极小，这一极小的误差可能是由于舍入无误差所造成的，因此有理由相信前文所估计得均值极具可信度。

## 六、问题二的模型建立与求解

### 6.1 车票请求量数据模拟

#### 6.1.1 获取数据

从题目所给的参考网址获取 2012 年 12 月至 2013 年 5 月每百万人中日平均访问人数走势图如下图：



图 6.1 每百万人中日平均访问人数走势图

从图中可以得到如下可能用到的信息：

- (1) 在 2013 年 1 月 13 日到 20 日每百万人中日平均访问人数走势图近似为抛物线；
- (2) 从图像中可以估计在 2013 年 1 月 17 日日均访问量达到高峰；

因此，在本文中我们可以估计 1 月 17 日每百万人中平均访问人数为：9000 人；根据国家统计局最新公布的人口普查数字 13.4 亿（此处为约数），可估计 1 月 17 日共有  $900000 \times 13.4 = 12060000 = 1206$  万人访问了该网站。

#### 6.1.2 数据模拟

根据 Alexa 网站提供的数据，我们已经粗略估计了 2013 年 1 月 17 日总访问量，虽然该数据不是很精确，但在本文能够说明问题，故认为该数据具有可靠性；另一方面，一个网站的日访问量应该是一天的累计访问量，是个总数，而不是某个时间段的访问量。考虑到这一点，我们自然想到了斐波那契（Fibonacci）级数，因为这一级数的基本特征是：前面相邻两项之和，构成了后一项，即：

$$F_n = F_{n-1} + F_{n-2} (n > 2), F_1 = F_2 = 1;$$

这一点正好符合一个网站的日累计访问量。在这里我们不妨设在原铁道部规定的放票时间点上，网络访问量达到最大；且忽略从 0 点到早上 8 点之间的访问量，这是因为在该时间段并不在网上放票，自然访问者也无法进入队伍。

有了 Fibonacci 级数的启发，接下来的工作应该是找到一个最接近我们已经估计得到的日累计访问量。这样再倒推就可以模拟出各个时间段的访问量。在此

我们在 VC6++ 的运行环境下，编写程序（见附录一），运行到第 36 个数时发现该数字与我们已估计的数据接近。表 6.1 罗列了第 27 到第 36 个 *Fibonacci* 数字。

表 6.1 *Fibonacci* 级数表

编号	25	26	27	28	29	30
数值	75025	121393	196418	317811	514229	832040
编号	31	32	33	34	35	36
数值	1346269	2178309	3524578	5702887	9227465	14930352

根据表 6.1，虽然第 36 个数字为 14930352，并不等于我们已经估计的 1 月 17 日的累计访问量，但已是最接近的数字了，接下来对 *Fibonacci* 进行简要修正，使之尽可能的接近我们所估计的数字。

数据的修正过程应当是减小误差，增大精确度的过程。在这里我们不妨直接将第 36 个数字更正为 1206 万，即相当于原数列减少了 20%，再根据 *Fibonacci* 级数递推公式倒推之后可以得到下表。

表 6.2 修正后的“*Fibonacci* 级数”表

编号	27	28	29	30	31
数值	60020	97112	157134	254249	411439
编号	32	33	34	35	36
数值	665632	1077015	1742647	2819662	4562310

严格意义上说，表 6.2 已经不是 *Fibonacci* 数列了，它是根据 *Fibonacci* 数列倒退之后得到的。因此我们可以得到各个阶段 12306 网站的访问量。事实上表 6.2 即是所要求模拟的数据，只是需要将其中的编号改为时间。见下表。

表 6.3 各个时间段网络的访问量

时间	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00
访问量	60020	97112	157134	254249	411439
时间	13:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00
访问量	665632	1077015	1742647	2819662	4562310

## 6.2 队列的计算与应用

排队论在日常生活中应用比较广泛，且已经建立起较为完善的排队论模型体系。本文主要应用排队论的基本模型解决提出的实际问题。

### 6.2.1 排队论简介

排队论的基本结构如下图所示：

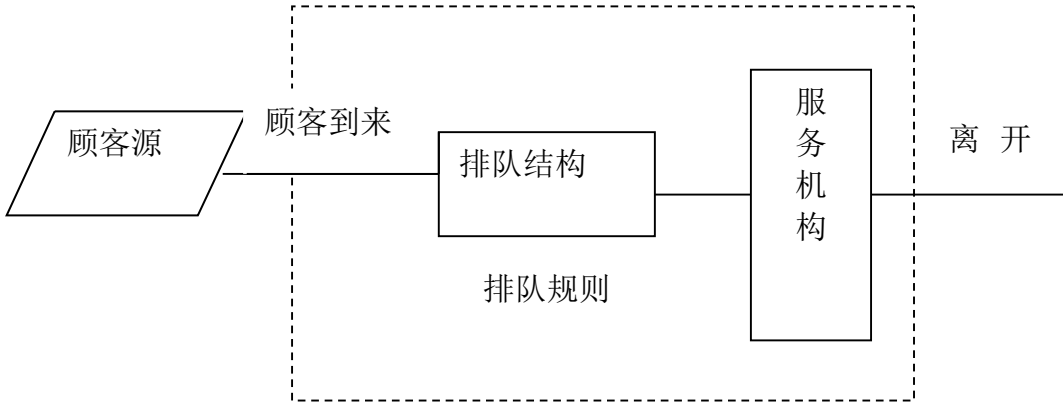


图 6.2 排队论结构

其中虚线框内是排队系统的核心，也是本文研究使用的重点。通俗的说排队应包括进入队伍（输入过程），被服务过程，离开队伍三个阶段。本文主要解决的是进入队伍、接受服务的过程，其中以进入队伍为重点；而离开的过程一般是不需要研究的，只要服务完毕自然就会离开。

1. 输入过程。顾客的到来完全随机的，需要对大量数据总结、整理并根据经验抽象、概括出顾客达到的数学模型，也就是排队论中所说的顾客到达的间隔时间分布。

2. 服务规则。在排队论中服务规则多种多样，如：先到先服务，后到先服务，随机服务等。在本文中我们使用先到先服务（FCFS）的服务规则，这也是社会普遍认同的服务方式。

3. 服务机构。排队论中有单队——单服务台、单队——多服务台（各服务台平行或各服务台串联）、多队——多服务台等多种情形，本文采用单队——多服务台（各服务台之间平行）。服务过程也即服务时间可以是随机型的也可以是确定型的，但大部分服务时间都是随机型的，这就需要找出随机型服务时间的概率分布。

4.  $M/M/n$  模型简介。 $D.G.Kendall$  提出在排队论中被广泛采用的“ $Kendall$  记号”。其一般格式为：

$$X/Y/Z/A/B/C$$

其中  $X$  表示顾客相继到达的时间间隔分布； $Y$  表示服务时间的分布； $Z$  表示服务台的个数； $A$  表示系统容量，即可容纳的最多顾客数（包括正在服务和排队等待的顾客）； $B$  表示顾客源的数目； $C$  表示服务规则。这种分类方法是以排队系统中的主要特征，即顾客相继到达间隔时间的分布、服务时间分布和窗口数对排队模型进行分类。

## 6.2.2 顾客到达的间隔时间和服务时间分布

设  $N(t)$  表示在时间区间  $[0, t)$  内到达的顾客数（ $t > 0$ ）

令  $P_n(t_1, t_2)$  表示在时间区间  $[t_1, t_2)$ （ $t_2 > t_1$ ）内有（ $n \geq 0$ ）个顾客到达（这是随机事件）的概率，即：

$$P_n(t_1, t_2) = P\{N(t_2) - N(t_1) = n\} (t_2 > t_1, n \geq 0)$$

当  $P_n(t_1, t_2)$  合于下列三个条件时，则认为顾客的到达形成泊松流，这三个条件是：

- (1) 在不重叠的时间区间内顾客到达数是相互独立的，称之为无后效性；
- (2) 在充分小的  $\Delta t$ ，在时间区间  $[t, t + \Delta t)$  内有 1 个顾客到达的概率与  $t$  无关，

而约与区间长度  $\Delta t$  成正比，即  $P_1(t, t + \Delta t) = \lambda \Delta t + o(\Delta t)$ ，其中  $o(\Delta t)$ ，当  $\Delta t \rightarrow 0$  时，是关于  $\Delta t$  的高阶无穷小。 $\lambda > 0$  是常数，他表示单位时间内有 1 个顾客到达的概率，称为概率强度。

- (3) 对于充分小的  $\Delta t$ ，在区间  $[t, t + \Delta t)$  内有 2 个或 2 个以上顾客到达的概率极

小，以至于可以忽略，即  $\sum_{n=2}^{\infty} P_n(t, t + \Delta t) = o(\Delta t)$

根据以上三条假设，我们认为本题中的输入过程服从泊松流，那么顾客相继到达的间隔时间  $T$  必须服从负指数分布（理由不再论述，见参考文献[3]），这里给出负指数分布的概率密度表达式：

$$f_v(t) = \begin{cases} \mu e^{-\mu t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

在输入过程服从泊松流的情况下，这里的  $\lambda$  有实际意义：单位时间内平均到达的顾客数。由前文已经模拟出的各阶段的网站访问量，可以得出个阶段的  $\lambda$  值：

表 6.4  $\lambda$  数值及服务台的个数数据表

时间段	$\lambda$	$n$ （服务台个数）
8:00——9:00	16.67	1
9:00——10:00	26.96	2
10:00——11:00	43.65	3
11:00——12:00	70.62	4
12:00——13:00	114.29	5
13:00——15:00	184.90	6
15:00——16:00	299.17	7
16:00——17:00	484.07	8
17:00——18:00	783.24	9
18:00——20:00	1267.31	10

对于服务时间的分布，我们依然采用负指数分布（理由见参考文献[3]），

$$f_v(t) = \begin{cases} \mu e^{-\mu t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

这里的  $\mu$  同样有实际意义：单位时间内能被服务完的顾客，称为平均服务率，

而  $\frac{1}{\mu} = E(v)$  表示一个顾客的平均服务时间，这里的平均就是期望值。根据题目所

给附表中的出对时间及前文的假设可以算的  $\mu = \frac{1}{E(v)} = 89.93364$ 。

### 6.2.3 $M/M/n$ 模型应用

承前文所述，这是一个到达时间间隔和服务时间分布均服从负指数分布，且有  $n$  个服务台的模型。在此之前先给出服务强度的  $\rho$  的定义。

$$\rho = \frac{\lambda}{\mu} = \frac{(1/\mu)}{(1/\lambda)}$$

它前一个表达式表示平均达到率与平均服务率之比，即在相同时区内顾客到达的平均数与被服务的平均数之比；后一个表达式表示一个顾客的服务时间与到

达间隔时间之比，称  $\rho$  为服务强度或话务强度。

令  $\rho = \frac{\lambda}{n\mu}$ ，只有当  $\frac{\lambda}{n\mu} < 1$  时才不会无限的队列，称它为这个系统的服务强度或服务机构的平均利用率。

根据  $\rho$  的定义我们先给出各个时间段内的  $\rho$  值见表 6.5

表 6.5  $\rho$  的数值表

时间段	$\lambda$	$\mu$	$n$	$\rho$
8:00-9:00	16.67222	89.93364	1	0.185384
9:00-10:00	26.97556	89.93364	2	0.149975
10:00-11:00	43.64833	89.93364	3	0.16178
11:00-12:00	70.62472	89.93364	4	0.196325
12:00-13:00	114.2886	89.93364	5	0.254162
13:00-15:00	184.8978	89.93364	6	0.342656
15:00-16:00	299.1708	89.93364	7	0.475225
16:00-17:00	484.0686	89.93364	8	0.672814
17:00-18:00	783.2394	89.93364	9	0.967676
18:00-19:00	1267.308	89.93364	10	1.409159

从表中计算的  $\rho$  值看：在 18:00—19:00 之间  $\rho=1.4>1$ ，可能会排成无限的队列。这一点与实际相符合，在 18:00 时网上放票量最大，自然购票人就增加，因此在该阶段出现无限排队的现象在情理之中。

接下来我们需要计算每个队列空闲的概率、顾客需要等待的概率、平均等待顾客人数、平均顾客人数、顾客平均逗留时间、顾客平均等待时间等 6 个指标，这些指标的计算对于  $M/M/1$  模型很好计算，但对于  $Z \geq 2$  的计算就相当困难了，本文使用 matlab 程序（见附录二）计算，在计算结果（见附录三）中发现：除了  $n=9$  这一时间段（17:00—18:00）比较符合生活实际，因此我们大胆推测将该情况下的模型推广到整个队列中；虽然这一推断无法用严格的数学逻辑进行推理证明，但符合事实就是最好的证明。摘取  $n=9$  如下表：

表 6.6  $n=9$  数量指标计算结果

	$P_0$	$L_q$	$L_s$	$W_s$	$W_q$
$n=9$	0.0000	26.6834	35.3925	0.0452	0.0341

该表显示：在春运期间，火车票需求量较大，队列的所有服务台都处于繁忙状态，都得到充分利用；系统中的顾客数达 35 人每秒；其中正在排队等候的顾客达 27 人每秒；一个顾客在系统中逗留的时间达 0.0452 秒；其中在系统中排队等候的时间 0.0341 秒；在顾客进入队列后，等待的时间极短只有 0.0111 秒。

由前文模拟的数据 1 月 17 日访问量为 1206 万，按失误率不低于 0.1%、总票数为  $N$  的要求，并设（1）有  $x$  个队列，（2）根据网站公布的售票时间，设一天只有 16 个小时能买到票；（3）在一天内在这 16 个小时（早 8 点到 24 点）之

外的访问视为无效。基于以上 3 点可得到下列方程（或不等式）：

$$\begin{cases} \frac{35 \times 3600 \times 16x}{10000} = 1206 \times (1 - 0.1\%) \\ 1206 \times (1 - 0.1\%) > \frac{N}{10000} \end{cases}$$

解出它的临界值： $x = 6$ 。即需要六个这样的队列才能满足需求。

## 七、问题三的模型建立与求解

### 7.1 对现行分时购票策略的评价

从第六部分计算的数量指标积分此结果看，分时策略可以分散请求量，缓解了访问量过大的问题，尽可能的使更多的人能够从网上买到票，可以说分时策略是一个很好的解决春运购票难的问题。但是还存在一些问题：（1）没有考虑成本；（2）在现行的分时策略下很容易出现大量抢票软件，构成新的社会不公平，甚至引发社会稳定问题；（3）现行分时策略下，在各时间段开始售票的火车站数量不尽相同，且大站与小站的搭配不尽合理问题等。这些都是导致网络阻塞，购票困难原因。本文接下来就问题（1）进行研究解决。

### 7.2 购票策略的优化与计算结果

#### 7.2.1 愿望模型的提出

这里的成本不一定是用钱来衡量，12306 网站增设一个服务器可以很容易用钱衡量，但顾客因买票而等待的时间是难以用钱来衡量的。因此在这里我们不研究 12306 网站增设服务器的成本，而着重研究顾客因买票等待时间的“成本”。这里的成本不能用钱等其他实物来衡量，我们提出使用“愿望模型”来解决该问题。所谓愿望模型就是顾客希望在最短时间内购到所需火车票的愿望。用排队论中的术语就是平均逗留时间（排队等待时间和服务时间之和）最短。

#### 7.2.2 模型的优化

本文第二问仅从生活实际的角度考虑选取第九组数据的排队论模型作为整体的队列，而没有充分考虑顾客的愿望，因此在这里模型的选取不仅要考虑生活实际更要加入顾客愿望这一要素。

我们现已第八组数据为基础，分别计算有 5、6、7、8、9 个服务台是 5 个数量指标，并使用 *matlab* 画出服务台个数与话务强度的关系图。

表 7.1 数量指标数据表

	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$
$\rho$	1.0765	0.8971	0.7689	0.6728	0.5981
$P_0$	-0.0024	0.0022	0.0038	0.0043	0.0045
$L_q$	-16.8781	6.3902	1.4079	0.4721	0.1737
$L_s$	-11.4956	11.7727	6.7904	5.8546	5.5562
$W_s$	-0.0237	0.0243	0.0140	0.0121	0.0115
$W_q$	-0.0349	0.0132	0.0029	0.0010	0.0004

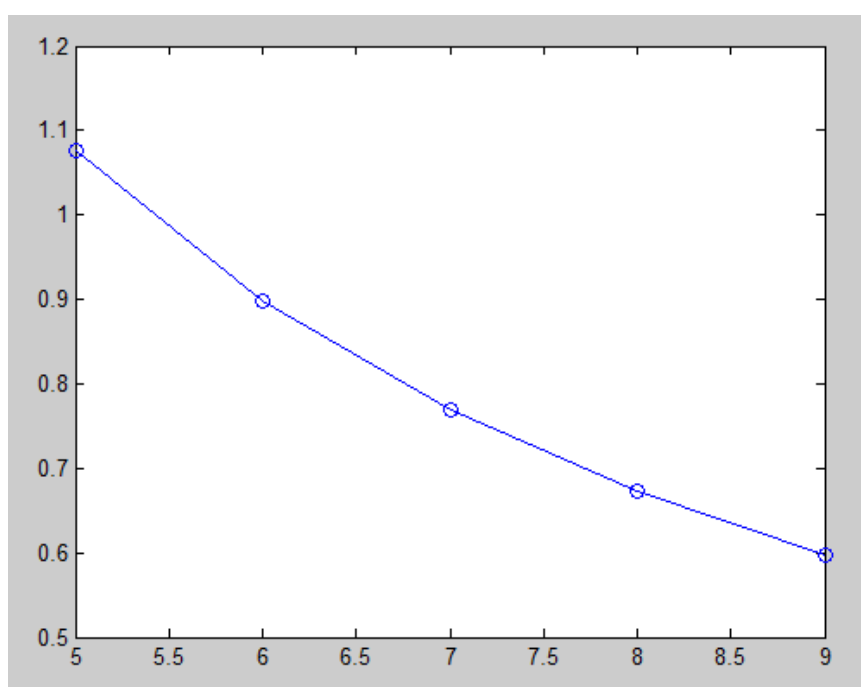


图 7.1 服务台数与话务强度的关系

从上表可以得出：当服务台个数为 5 时，各项指标为负，与实际情况不符，故直接舍去不予考虑，另一方面当服务台个数为大于 9 时，前四个数量指标均无变化，只有最后一个指标有微小变化（这里没有给出  $n > 9$  的数量指标），因此也不再考虑。

接下来就是在充分考虑生活实际与顾客的愿望模型来敲定选取几个服务台。事实正这里还涉及到互为矛盾的两项指标：

- (1) 顾客在系统中平均等待的时间
- (2) 服务台的空闲时间比例  $I$ 。对这两项指标，我们可以分别确定其意向水平  $\alpha$  和  $\beta$ ，作为上述两项指标的上界值。即满足： $W_q \leq \alpha, I \leq \beta$

其中服务台的空闲比例可以用话务强度衡量，话务强度越大则空闲比例越小。从表中可以看出  $n=6$  时服务台得到充分利用达 89.71%，比其他情形都能充分使用资源，因此选取 6 个服务台。

因此应该在 16:00—17:00 之间开设 6 个服务台，这样既能缩减网站的成本，又能是各个服务台得到充分利用，不会造成资源浪费。进而将这一模型推广，即得到优化后分时方法：8:00、11:00、15:00、17:00、18:00、20:00。

### 7.2.3 计算结果

根据第二问的模型

$$\begin{cases} \frac{11 \times 3600 \times 16x}{10000} = 1206 \times (1 - 0.1\%) \\ 1206 \times (1 - 0.1\%) > \frac{N}{10000} \end{cases}$$

我们直接可以算的需要约 18 个这样的队列。

## 八、关于第四问

针对春运一票难求的问题结合本文的模型和数量指标提供如下意见供参考：

(1) 合理分配网络放票量与车站售票量。虽然网络售票能给消费者带来极大的便利，减少不必要的开支（如打的到车站等），但应注意在我国仍有不少人不会上网或不具备上网购票的条件，因此从社会公平角度考虑应合理分配网络售票与车站售票的比例，并根据实际情况适时做出调整。

(2) 实行“查购分开”。即火车票信息查询与车票购买分开，可以适当放权，让一些具有公信力的网站缓存查询结果，减少对 12306 网站的查询压力，进而增大网友购票成功的机会。

(3) 按需生产。允许网友（消费者）提前预订车票，铁路总局再根据预订信息来安排春运与其他节假日的运营能力。

(4) 建议网站增加“记忆功能”。也就是说让网站变得聪明一些，能够记住网友的购买信息。一些网友在购票失败后会重新登录，再重新走一遍购票的全流程，这无疑增大了服务器的工作量（相同的工作可能要做好几遍），因此增加记忆功能，记录网友的乘车区间，再次登录后可以减少对其他网页的点击，直接进入付账购票环节。

(5) 减少一些购票过程中不必要的累赘网页，精简网页内容。从现行的网站售票系统看，购买一张票需要点击多个网页，不如减少步骤，按网络的“思维”设置购票程序，而不是照搬车站的购票方式；精简网页内容可以防止网友点击与购票无关的页面。

(6) 增加透明度，加强管理，自觉接受监督。每年春运“黄牛党”等非法倒票现象屡禁不止，即使在实名制下，一些精明的“黄牛党”仍获利不少。因此铁路总局应在不涉及国家或政府秘密的前提下公开总售票量，自觉接受社会各界监督，加大市场监管力度，压力打击黄牛党等非法倒票人员。

## 九、模型的总结与推广

### 9.1 模型的总结

本文以 12306 网站售票系统为背景，主要利用排队论知识对购票系统建立相应的数学模型，并对模型相关参数进行计算和分析，并根据题目要求完成了任务，为网络售票系统提供了理论依据。

本文所用模型不多，且为常见模型这正是本文的特点。我们认为对于实际数学问题，所用的数学方法越简单越有利用相关部门采纳使用，也可增加文章的可读性。

本文所用模型的主要特征：

(1) 第一问中主要使用无偏估计、区间估计等传统数学模型。这类模型发展比较成熟完善，且已形成了使用该类模型的基本步骤。在现有的数学建模问题中，大数据量已成为其基本特征，因此样本估计总体的思路处处可见。本文主要是在前人的研究基础上恰当的估计了每秒最多能处理的数据量。

(2) 第二问中分别使用了 *Fibonacci* 级数原理和排队论模型，且排队论模型贯穿于全文。*Fibonacci* 级数在数学模型中的使用还不是很多，本文主要是使用 *Fibonacci* 级数的原理来近似模拟各个时间段网站的累计访问量；排队论模型是本文的主要模型，使用该模型主要是为了计算各个数量指标并进行计算分析，顺



利解决题目中提出的要求;。

(3) 排队论本身并不是很难,主要工作应该在从已经搜集的数据中概括抽象出到达的时间间隔分布和服务时间分布,因此本文化了大量篇幅来证明这个问题,只有抽象的时间间隔分布、服务时间分布的模型合理,排队论的后续工作才有意义,才能使后续工作顺利进展。

## 9.2 模型的推广

*Fibonacci* 级数源于兔子的繁殖,但其基本原理在实际生活有着极大的用途;*Fibonacci* 级数的使用也使得 VC6++ 应用实际数学中,因此在学习数学,特别是在数学模型的建立过程抽出其原理、本质,用大家耳熟能详模型解决实际问题,使文章更具可读性。

排队论模型的应用非常广泛。小到在菜市场买菜、食堂打饭大到网络访问、医院就医等,排队论处处可见。排队论的使用应该跟加注意数据的搜集、整理过程,这一过程的合理性直接影响排队论模型的可用性。因此在以后的使用过程中应更加注重数据来源的可靠性、数据处理的合理性。

本文主要使用  $M/M/n$  模型,该模型的研究使用比较完善,在许多领域都有广泛的应用。特别是  $M/M/n$  模型的优化设计更具有使用价值,但现有的资料对其研究并不深刻,这可能是今后的一个重要研究课题;另一方面对于一个队列多个服务台串联或是多个队列多个服务台的模型则需要进一步研究,相信若这些模型应用到本文将能更好的解决相关问题。

## 十、参考文献

- [1] 叶慈南, 曹伟丽. 应用数理统计[M]. 北京: 机械工业出版社, 2004.6.
- [2] 王雪萍. 排队论在体检系统中的应用研究[D]. 华中师范大学, 2008.6.
- [3] 钱颂迪等. 运筹学(第三版)[M]. 北京: 清华大学出版社, 2005:312-314, 324-326.
- [4] 谷海兰, 张威. 应用 matlab 软件解决排队论模型问题[J]. 河北企业, 2011, 第四期.
- [5] 赵静, 但琦. 数学建模与数学实验[M]. 北京: 到高等教育出版社, 2008.3.
- [6] 郑莉, 董渊, 何江舟. C++语言程序设计[M]. 北京: 清华大学出版社, 2010: 97.
- [7] 解争龙, 李向军. 基于排队论模型的网络拥塞率研究[A]. 计算机工程与设计, 2007, 28(17)
- [8] 贾俊平, 何晓琼, 金勇进. 统计学[M]. 北京: 中国人民大学出版社, 2012.6.

## 附录

附录 1: *Fibonacci* 级数程序 (在 VC6++ 环境下运行):

```
#include <iostream>
using namespace std;
long int fib(int n)
{int f=1,f2=1;
  If (n==2||n==1)
    f=1;
  else f=fib(n-1)+fib(n-2);
  return f;
}
int main()
{int n;
  cout<<"请输入一个正整数:";
  cin>>n;
  long int y=fib(n);
  cout<<y<<endl;
  return 0;
}
```

附录 2: 计算排队论数量指标的 matlab 源程序: (本程序中用 s 表示服务台个数, 只需在 command window 中输入相应的参数并调用 f(lambda,mu,s))

```
function x=f(lambda,mu,s)
ro=lambda/mu;ros=ro/s
sum1=0;
for (i=0:s-1)
sum1=sum1+ro.^i/factorial(i);
end
sum2=ro.^s/factorial(s)/(1-ros);
p0=1/(sum1+sum2);
p=ro.^s.* p0/factorial(s)/(1-ros);
Lq=p.*ros/(1-ros);
L=Lq+ro;
W=L/lambda;
Wq=Lq/lambda;
x(1)=p0 ;
x(2)=p ;
x(3)=Lq ;
x(4)=L ;
x(5)=W;
x(6)=Wq;
end
```

附录 3.

表 1 单个或多个服务台数量指标计算结果

	1	2	3	4	5	6	7	8	9	10
$P_0$	0.8146	0.7392	0.6151	0.4558	0.2804	0.1277	0.0356	0.0043	0.000	-0.000
$L_q$	0.0422	0.0069	0.0027	0.0022	0.0035	0.0106	0.0550	0.4721	26.6834	-10.3220
$L_s$	0.2276	0.3069	0.4880	0.7875	1.2743	2.0666	3.3816	5.8546	35.3925	3.7696
$W_s$	0.0136	0.0144	0.0112	0.0112	0.0112	0.0112	0.0113	0.0121	0.0452	0.0030
$W_q$	0.0025	0.0003	0.0001	0.0000	0.0000	0.0001	0.0002	0.0010	0.0341	-0.0081