

评委一评分，签名及备注	队号： 1259	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

中国自 2011 年起，开始实行网络售票。购买火车票的方式也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。然而，网络购票给人们带来了便利的同时，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。

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；第二，唯一资源问题，也就是唯一的一张车票。在同一时刻，可能有多个人在抢购一张票。

为了能够更好的解决订票中出现的问题，我们对现有的资料进行了分析并对火车票购票网站进行了深入的了解。我们了解到火车票购票网站采用的是排队的方法，并采用分时策略。来应对在购票时出现的问题。

针对问题一：我们对测试队列性能的入队列数据和出队列数据，运用 Excel 进行了分类汇总，借助 Matlab 工具，对数据进行了适当的处理，模拟数据的大致趋势。得到入队列速率，与出队列速率均有先增加后减少的趋势。为此我们又运用了 SPSS 软件分别对入队列数据与出队列数据进行回归分析，通过模型汇总的显示，我们得到了入队列速率与出队列速率的函数关系。从而进一步，求出了队列的数据处理能力，及其错误率。

针对问题二：为了缓解购票时出现的问题，增加队列也被认为是一种比较不错的方法。即增加队列数，可以分散客流量。为此我们对火车票购票网站的日访问量进行了合理的计算，并在现有分时策略的基础上，以及在解决第一题的过程中得到的一些结论，进行了合理的模拟。运用 SPSS 软件，对入队速率与错误率分别进行了一次多项式合和二次式拟合。根据拟合度最优原则，得到入队速率与错误率存在线性关系。结合实际情况，考虑到队列长度和入队速率，我们建议设置 7 个队列较为合适。

针对问题三，我们先对现有的分时策略进行了简单的优化。然后我们从分散访问量和降低系统内存浪费两个方面的因素进行考虑，建立模型。通过对模型的求解，我们得到了一个比较好的方案。即，分时段为 7 个，队列个数为 10 个。

这种方案下，系统内存的利用率最大，而且可以更好地应对售票起售时过大的访问量。

针对问题四，12306.cn 网站并不是完美的，所以还是有许多地方需要改进。我们通过查阅资料，进行分析，给出了 9 条建议。

关键词： 队列性能 回归分析 多项式拟合 网站优化策略

火车票购票网站优化

1.问题的重述

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。网络购票给人们带来了便利，得旅客可以不出家门在网络上就能买到车票；售票窗口也不用那么繁忙，节省了大量的人力物力。使但是随着购票人数的增加，网络购票也出现了不少问题。为了保证网站更稳定的工作，需要你帮忙提出优化意见。

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。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。

1.1 问题一

根据附件中给出的队列的测试数据，估计该队列每秒最多能处理多少数据，错误率是多少。

1.2 问题二

根据网站日累计访问量数据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）。

1.3 问题三

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。

1.4 问题四

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。

2.问题的分析

据了解,队列(Queue)是一种特殊的线性表,它是一种运算受限的线性表。它只允许在表的一端进行插入,而在另一端进行删除。允许删除的一端称为队头(front),允许插入的一端称为队尾(rear)。具体的操作方式为,队尾入队列,队首出队列。因此队列亦称作先进先出(First In First Out)的线性表,简称 FIFO 表。队列的存储结构分为顺序存储和链接存储。队列里的顺序存储需要使用一个数组和两个整数来实现,队列的链接存储用链接存储结构实现。

2.1 问题一分析

2.1 估计队列每秒最多能处理多少数据

首先分别对附件中的入队列和出队列数据进行分析,利用 Excel 对分别对入队的数据和出队列的数据进行分类汇总,得到如下的结论。

结论一: 相同的数据量的进入队列,所需要的时间不同,少则需要 9 秒,多则需要 67 秒,见表 1:

不同的时段 1000 个数据进入队列的总耗时										
测试组编号	1	2	3	4	5	6	7	8	9	10
所用时间 t(秒)	9	9	10	14	20	25	38	46	57	67

表 1

通过对每一测试组的数据进一步分析,我们发现对于测试组 1,在其第九秒时间内,只有一个数据进入队里,测试组 3 也存在同样的情况。考虑到在截取 1000 个数据时,存在开始时间控制不准,以及截取最后一个数据时,并不能把这一秒里的数据全部截取到。因此为了减少误差我们将开始的一秒内和结束的一秒内少于五个数据的时间舍去,忽略不计。得到如下表 2:

不同的时段进入队列的总耗时										
测试组编号	1	2	3	4	5	6	7	8	9	10
所用时间 t(秒)	8	9	9	14	19	25	36	45	56	66

表 2

结论二：不同时刻，同样一秒的时间，进入队列的数据也不同，多则 220 调数据进入队列，少则 1 条数据进入队列

结论三：入队列的时间 t'_λ 等于入队总耗时间 t_λ 与获取号码的时间 $t_\text{号}$ 之差。

即 $t'_\lambda = t_\lambda - t_\text{号}$ 。通过对入队列的数据分析，获取号码的时间 $t_\text{号}$ 是入队时间 t'_λ 的 2~10 倍。入队时间相对浮动不大，获取号码的时间相比上小浮动比较大，这与系统的分号的随机性是有关的。

结论四：对出队列的数据进行分析，我们发现同一秒的时间内出队列的数据是不同的。原因在于出队列的时候队列不一定有正在排队的数据所以每秒处理的数据多少是不一样的。对每一秒内的出队列数据的耗时进行加和，我们发现其总和少于 1 秒，多则为 0.9259 秒，少则为 0.0196 秒。

对一条数据的处理时间 $t_\text{总}$ 是包括入队总耗时间 t_λ 和出队列时间 $t_\text{出}$ 之和，

即 $t_\text{总} = t_\lambda + t_\text{出}$ 。那么队列的每秒处理的数据的数据量 $Q(t) = \frac{1}{t_\text{总}}$ 。

2.1.2 求错误率

由于出队列是不会发生错误的，因此错误率仅是由入队列决定的。入队列出错是指如果获取号码超过 3 次（即 4 次）的直接返回失败，不进行入队列。我们对每组的入队列出错数据进行统计，分类汇总得到如表 3：

测试组编号	总数据量	出错数据
1	1000	24
2	1000	52
3	1000	53
4	1000	8
5	1000	8
6	1000	2
7	1000	2
8	1000	3
9	1000	0
10	1000	0
总计	10000	152

表 3

表格可知，第九和第十组没有出现数据的错误。而出错比较多的来自于第一，二，三组。

2.2 问题二分析

根据对参考资料的分析和实际情况的了解，由 aleax 的统计图 1 可知，2013

年 1 月 17 日大概是进入了春运的高峰期，因此 12306. cn 网站的日访问量激增，为了缓解访问量过大，给系统带来的压力。火车票购票网站采用分时购票策略和排队的方法。



图 1[1]

我们将入队列数据表格导入到 Matlab 工作区间里，对 10000 条入队列数据进行统计分析，得到图 2，如下所示：

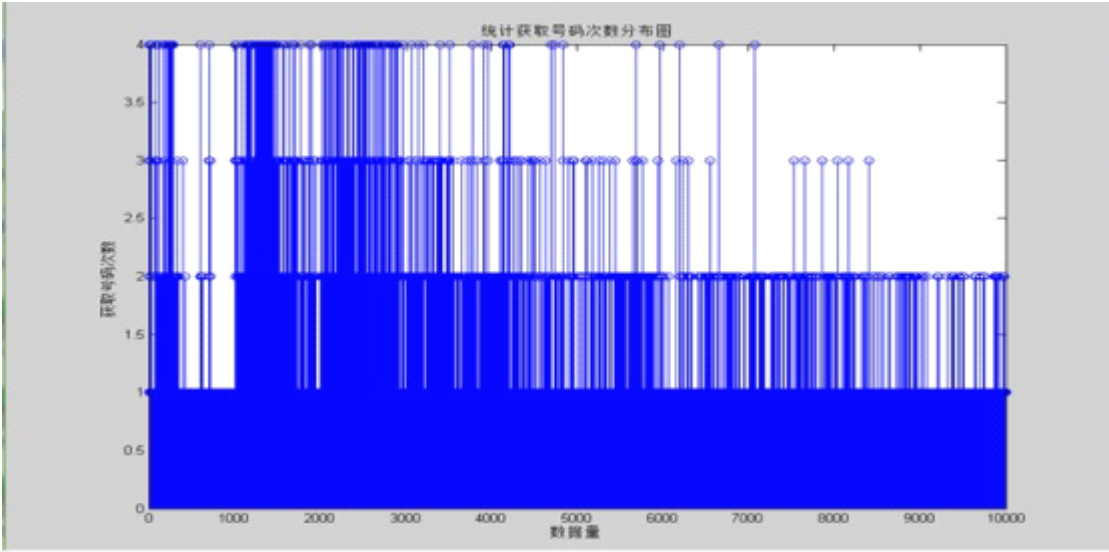


图 2

通过图的显示，此图分为了四个区域。蓝色最越密的区域为获取号码一次，就成功进去队列的数据。我们会发现在第四个区域里，第一，二，三组的蓝色密集程度较其它组更大。

我们对不同测试组下的错误率进行了绘图，见图 3：

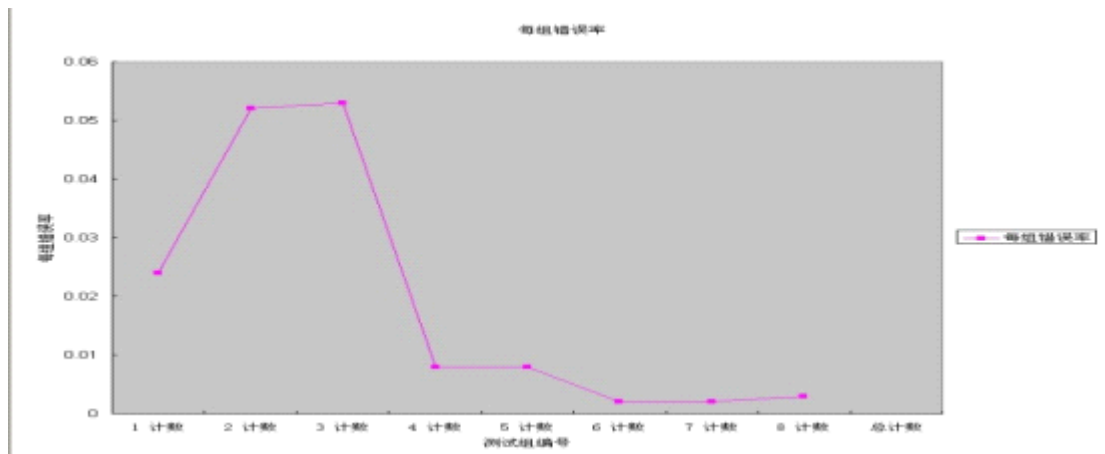


图 3

从图 2 中我们可以观察到，错误率比较的时，分别出现在第一，二，三组。由第一问分析的结果得出地一，二，三组的入队速率比较大。而后面的测试组相对前三组的入队速率比较小。以至于到第九，十组没有出现错误率。因此我们猜想当数据访问量达到一定程度时，才会出现入队不成功，即出现错误。

根据访问量可以理论的模拟出当日网站的订票的请求量，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。

访问量激增，如果只有一个队列供订票人员排队，那么平均每个人的排队等待时间就会相当大，错误率也会上升，严重影响到售票和购票效率，所以需要有多个队列同时进行排队，来分担巨大的订票请求量，这样在每个队列中排队的人，等待的时间就会减少并且错误率也会降低，从而提高售票和购票的效率。

求出的队列数需要满足两个条件：

①错误率低于 0.1%；

②队列排队的时间不超过 30 分钟；

根据提供的队列测试数据分析可以得到，申请入队的请求量越大，错误（即请求号码 4 次仍然没有成功）次数越多，可以初步估计为一个线性关系，如： $y=a*x+b$ ；其中 y 为一秒中错误次数， x 为一秒中的申请入队数， a, b 均为参数。因此减少入队的数据量可以减少错误率，增加队列数可以减少单个队列长度。

由题可得，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。这里可以忽略，也就是说，一个成功入队的用户的排队等待时间都是用在处理他前面的排队人数上，即前面排队人数的出队时间的总和。减少队列长度可以减少排队等待时间，而增加队列数可以减少队列的长度。

2.3 问题三分析

现行的分时购票策略，将大量的访问量分散，从而降低了系统处理购票的压力，同时也可以减少用户购票所用时间。而现行分时购票策略是将售票时间分为十段，是否为最优的分时策略？

分时策略所分时段多少，是可以建立数学模型的。现可做极限假设，当分时时段只有一个时，会造成大量用户购票拥挤进入队列里，造成系统阻塞。当分

时时段无限多时，会造成用户购票时查询困难，而且会给售票系统带来资源分配不便的问题。显然，分时时段不是越少越好，也不是越多越好。所以可以假设分时时段的多少对系统的利弊符合一种数学模型。

通过观察 12306.cn 所公布的数据，有的火车票销售时段所包含的火车站点过少；有的火车票销售时段所包含的火车站点则过多。因此，可以将火车站点重新分布，使得访问量的分布更加均匀，避免造成售票系统阻塞。

2.4 问题四分析

12306.cn 网站并不是完美的，所以还是有许多地方需要改进。

3.符号说明

$t_{\text{入}}$	入队列的时间	L	队列的长度
$t_{\text{号}}$	获取号码的时间	$G(t)$	出队列速率函数
$t_{\text{总}}$	入队列总耗时间	t_i	不同时段所用的时间
$Q(t)$	队列每秒处理数据的数据量，也称队列性能	$W(t)$	不同时刻队列里的数据量
$t_{\text{出}}$	出队列的时间	x_i	不同时段的错误率
$V_{\text{入}}$	入队列速率	z_i	不同时间段所占总时间的比重
$V_{\text{出}}$	出队列速率	η	内存的利用率

4.模型的建立与求解

4.1 问题一

4.1.1 估计队列每秒最多能处理多少数据

4.1.1.1 模型的分析

由于队列处理一条数据是要完成入队操作，和出队操作。而出队列的时候队列不一定有正在排队的人，所以每秒处理的数据（即出队列速率 $V_{\text{出}}$ ）是不一样。

经分析：当访问量比较小的时候，单位时间内进入队列的数据也比较小，即 V_{λ} 比较小，队列里的数据量 $W(t)$ 也就比较少。由于出队列速率 V_{μ} 是受队列里的数据量 $W(t)$ 的限制。因此，此阶段的出队速率 V_{μ} 是由入队速率 V_{λ} 决定，见图 4，也就是受访问量的限制的。所以在此阶段，队列的数据处理能力即 $Q(t)$ 不能达到最大值见图 4。

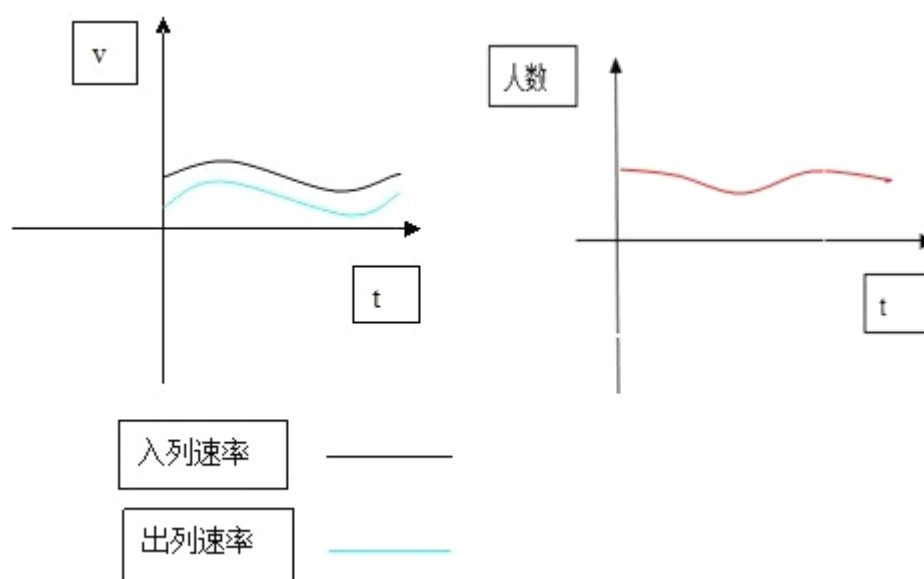


图 4

当访问量比较大的时候，入列速度 V_{λ} 开始时的基数比较大，这就好比火车票购票系统，在某个时间点开始出售某些站点的车票时，此时系统的访问量比较大。

随着访问量的继续增加，队列里面已有足够的数据，因而出列速率 V_{μ} 会随着队列里用户的增加而增加，逐渐达到最大值的。但是由于队列 L 是有限的，不可能无限制的有用户进入。此时队列处理数据的能力 $Q(t)$ 会受到出队速率 V_{μ} 的限制，因此入列速率 V_{λ} 也会慢慢的减小，降到与出列速率 V_{μ} 基本相等，见图 2。当入列速率 V_{λ} 与出列速率 V_{μ} 相同时队列中积攒的数据量达到最大。入列速率 V_{λ} 还会减小，出列速率 V_{μ} 会达到一个比较稳定的状态。此时为出列速率 V_{μ} 达到最大。队列处理数据的速率 $Q(t)$ 受到出列速率 V_{μ} 的限制。因此，此阶段的数据处理能力 $Q(t)$ ，见图 5，是由出队速率 V_{μ} 决定的。

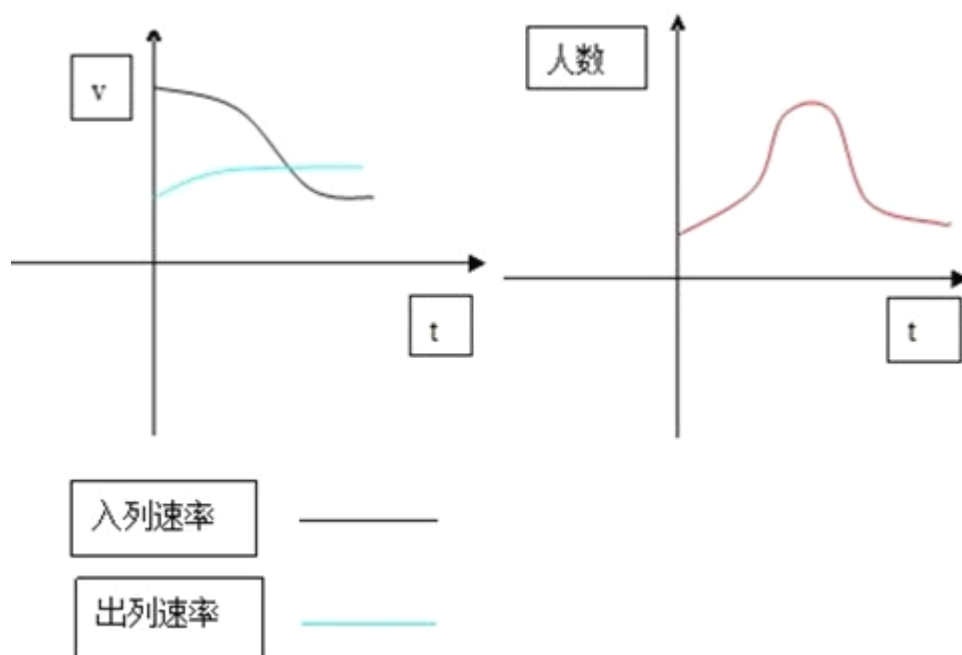


图 5

因此队列每秒处理数据的数量 $Q(t) = \begin{cases} V_{\lambda} & \text{访问量比较小时, 由 } V_{\lambda} \text{ 决定} \\ V_{\mu} & \text{访问量比较大时, 由 } V_{\mu} \text{ 决定} \end{cases}$ [4] (借

鉴了污水均流的设计一些思想)

4.1.1.2 模型的假设:

- (1) 假设队列的长度是有限的, 长队为 L ;
- (2) 假设所使用的实验数据是可靠的;
- (3) 假设出队列的速率规律是可以适用于不同的时间段;
- (4) 假设我们所研究的时段为高峰期;

4.1.1.3 模型的建立:

入队:

根据已给出的数据, 进行分类汇总, 我们判断入队的高峰时段为第一组、第二组数据, 我们用 Matlab 软件来模拟不同阶段的走势情况。

得到高峰期时段的走势图, 见图 6:

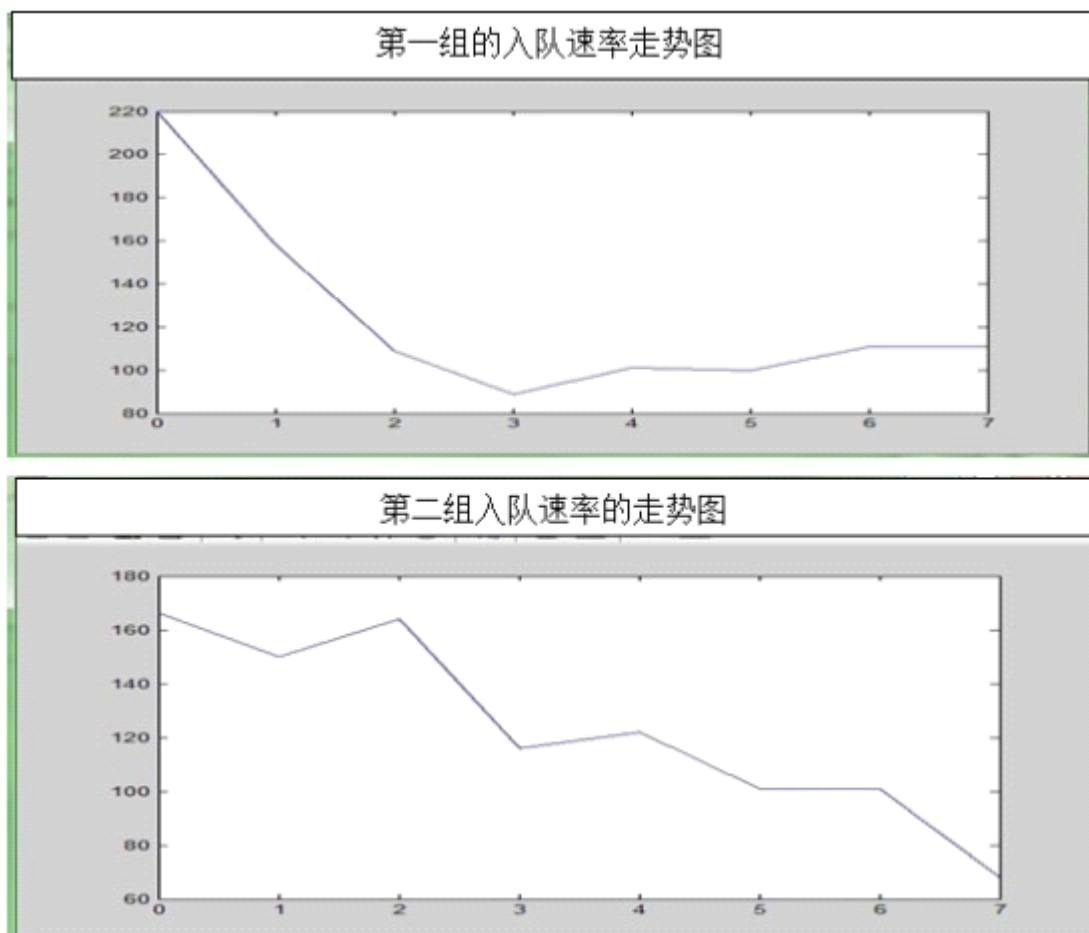


图 6

根据所给数据进行分析，出队列的高峰时段为 17:59。
得到高峰期时段出队列速率的走势图，见图 7：

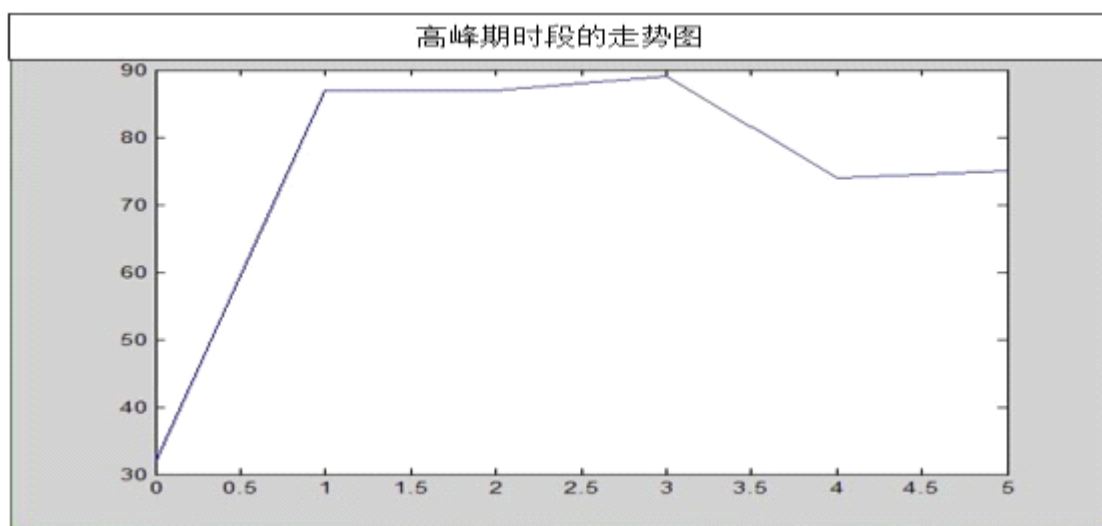


图 7

经比较模拟的数据的走势图与分析的走势情况基本相匹配。
对入队的第一组数据运用 SPSS 软件，进行二次回归拟合。
模型汇总与模型图像如下图 7：

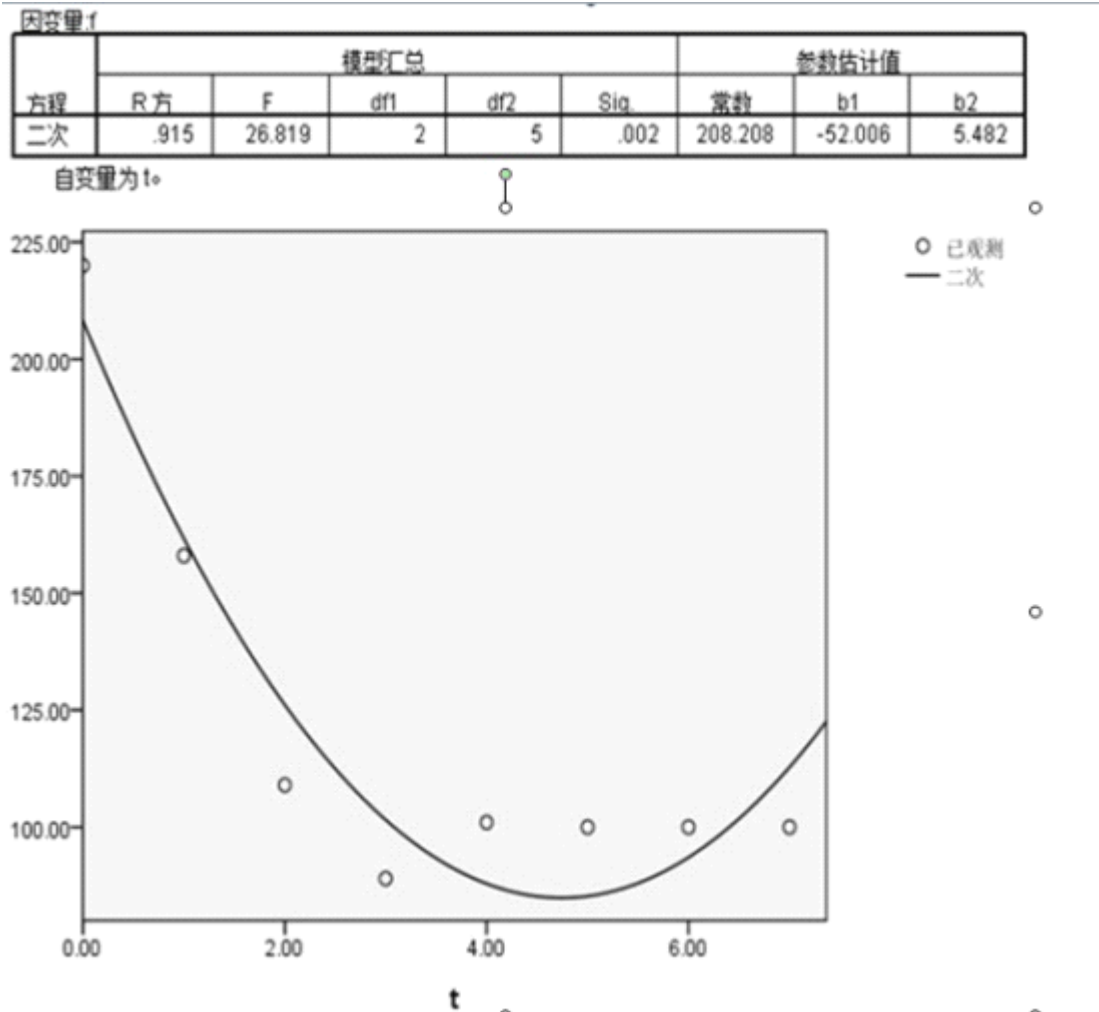


图 7

模型汇总情况：模型汇总表对回归方程进行拟合优度检验。由表可知拟合度 $R=0.915$ ，认为拟合优度较高。由于 $\text{Sig}=0.002<0.005$ ，因此关系显著。

得到入列速率的函数：

$$Q(t) = 5.5t^2 - 52t + 208 \quad 0 \leq t \leq 7$$

出队列：

通过分析我们已经知道，当访问量比较大的时候。出队列速率才有可能达到最大值，而此阶段的队列对数据的处理能力，是由出队列的速率决定的。因此我们选择 17；59 秒的数据，利用 SPSS 软件，进行二次回归拟合。

得到的模型汇总表和函数图像如下图 8：

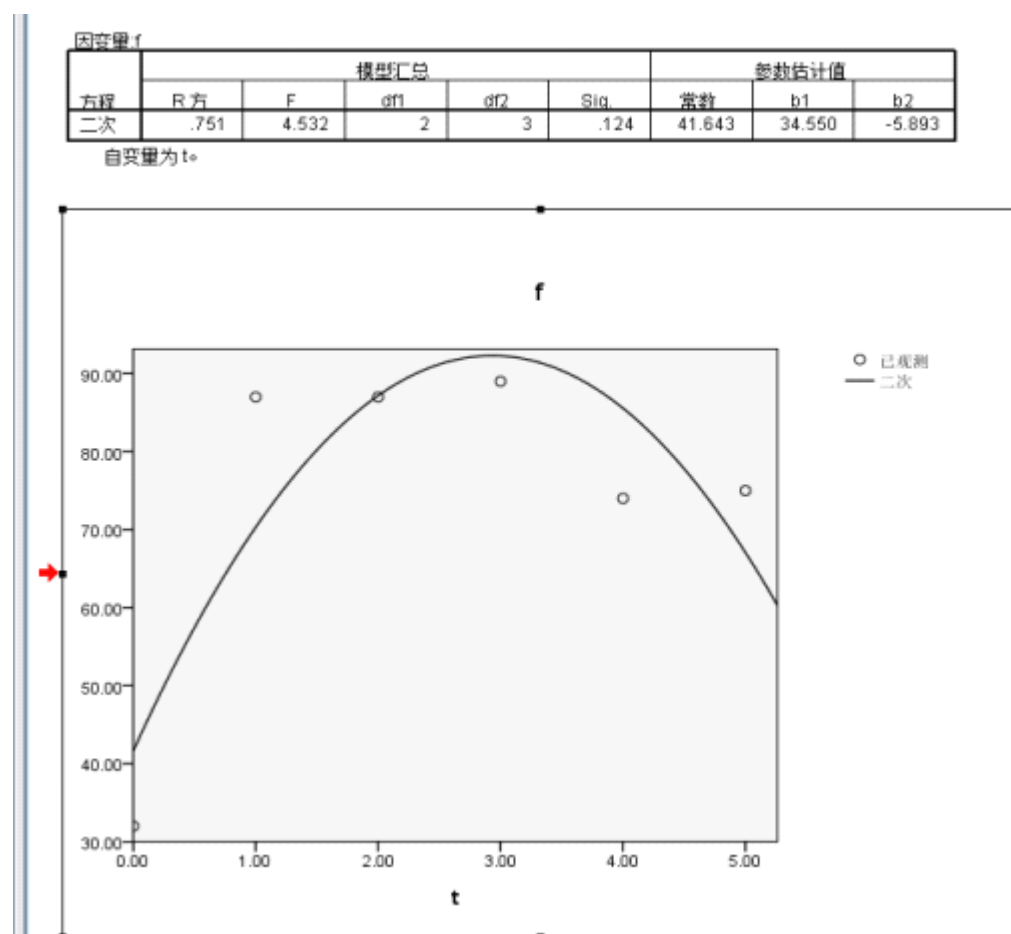


图 8

模型的汇总情况：模型汇总表对回归方程进行拟合优度检验。由表可知拟合度 $R=0.751>0.7$ ，认为拟合优度比较高。

得到出队列速率函数为：

$$G(t) = -5.9t^2 + 35t + 42 \quad 0 \leq t \leq 7$$

4.1.1.4 模型的求解：

当 $Q(t) = G(t)$ 时，队列中等待的人数达到最多。可以认为出列速率已经达到最大

得到等式 $11.4t^2 - 87t + 166 = 0$;

由于 $87^2 = 7569$ $4 * 11.4 * 166 \approx 7569$ 因此， t 只有一个解

解得 $t = 3.82s$

因此队列每秒最多处理的数据为： $G(3.82) = 90$ 人数/秒。

4.1.2 求错误率

错误率：

根据给出的数据，计算每组的错误率 x_i 与所用的时间 t_i 和每组时间所占的比重 z_i

(其中 $i=1, 2, 3, \dots$)，见表：

测试组编号	总数据量	出错数据	每组的错误率 x_i	每组所用时间 t_i	每组时间所占的比重 z_i
1	1000	24	0.024	8	0.02787
2	1000	52	0.052	9	0.03136
3	1000	53	0.053	9	0.03136
4	1000	8	0.008	14	0.04878
5	1000	8	0.008	19	0.06620
6	1000	2	0.002	25	0.08711
7	1000	2	0.002	36	0.12544
8	1000	3	0.003	45	0.15679
9	1000	0	0.000	56	0.19512
10	1000	0	0.000	66	0.22996
总计	10000	152	总计	287	

表 4

关于每组的错误率，可以发现单位时间内访问的数据量越大，也就是说当数据量相同的情况下，所用的时间越短其错误率越高。

由每组访问时间的长短，可以反映出当前时段页面访问量的大小。每组的时间越短其出错率越高，页面的访问量越大，反之页面的访问量越小，每组的时间越长，其出错率越低。我们认为单纯的用错误率加和或是求其期望值，均不是最好的错误。因此我们选择，通过对错误率的时间的比重作为权重值，可以得到准确的错误率。

错误率=Σ（错误次数/1000*每组时间/287）

即：

$$X = \sum (x_i * z_i)$$

将表 4 中的数据带入错误率公式
得到错误率

$$X = 0.00578$$

因此错误率为 0.578%

4.2 问题二

4.2.1 模型假设：

- 1、假设 2013 年 1 月 17 日网站的访问量非常大，从网站开始售票到结束，一直处于访问量的高峰期，此时火车票购票网站实行分时策略和排队方法。
- 2、假设全球网民大概有 22 亿，但是不是每天所有人都在上网，平均每天有 50% 的网民在上网，则访问量可估算为

“日访问量=当日每百万平均访问数*当天上网的全球网民数”。

其中“当天上网的全球网民数=22*50%（亿）”

- 3、因为 2013 年 1 月 17 日为春运高峰期，中国网民上网购票的频率急剧上升，可把中国网民的比例调整到 75%。
- 4、访问 12306 网站的网民并不是都是订票的，这里估计在 1 月 17 日（春运高峰

期) 总访问人数里有 98%的人是订票的, 则

“订票请求量=访问量*98%”。

5、假设队列排队时间都是用在等待前面排队用户出列上, 即

排队等待时间=前面排队用户出列时间的总和。

6、总票数 N 假定为当日网上售票总数, 当日总票数 N 会小于当日的订票请求量, 也就是说并不是所有订票的人都能订的上票。

4.2.2 模型建立与求解:

根据 alexa 的统计图显示, 2013 年 1 月 17 日当天的每百万人中大概有 9000 人在访问 12306 网站, 全球现在大约有 22 亿网民, 中国大约有 5.5 亿网民, 全球每天平均有大约 50%的网民在上网, 而在 2013 年 1 月 17 日正值春运高峰期, 中国网民上网访问的 12306 网站的比例要比平时高的多。

根据“日访问量=当日每百万平均访问数*当天上网的全球网民数”。

得:

当天上网的全球网民数= $((22-5.5)*50\%+5.5*75\%)*100*9000 \approx 11000000$

即为 1100 万访问量

而根据“订票请求量=访问量*98%”。

得:

订票请求量=1100 万*90% \approx 1000 万

即在 2013 年 1 月 17 日当日的订票请求量大概在 1000 万左右。

求每秒中错误次数和申请入队量之间的关系:

根据所给的测试数据, 对数据进行汇总整理, 获取其中的每一秒中错误次数和这一秒中申请入队的数, 适当去除一些无效的数据, 见表 5。

时间点(时:分:秒)	每秒申请入队数	每秒出错次数
19:47:00 计数	220	14
19:54:00 计数	166	10
19:54:02 计数	164	20
19:47:01 计数	158	8
19:54:01 计数	150	13
19:57:04 计数	139	9
19:57:05 计数	135	7
19:57:06 计数	135	8
19:57:03 计数	133	6
19:54:04 计数	122	3
19:57:00 计数	122	6
19:57:01 计数	121	9
19:54:03 计数	116	3
19:47:06 计数	111	0
19:47:02 计数	109	0
19:57:02 计数	106	5
19:47:04 计数	101	1
19:54:05 计数	101	1
19:54:06 计数	101	2
19:47:05 计数	100	1
19:47:03 计数	89	0
19:57:07 计数	89	3
20:20:03 计数	83	0
20:20:08 计数	82	0
20:20:01 计数	81	1

表 5

运用 SPSS 软件，对表 5 中的数据进行线性回归分析，模型汇总情况见表 6：

模型汇总 ^b										
模型	R	R 方	调整 R 方	标准 估计的误差	更改统计量					
					R 方更改	F 更改	df1	df2	Sig. F 更改	
1	.858 ^a	.735	.723	2.73532	.735	61.128	1	22	.000	
a. 预测变量: (常量), 访问人数。										
b. 因变量: 出错人数										

表 6

模型汇总：

模型汇总表对回归方程进行拟合优度检验。。由表可知拟合度 R=0.858，认为拟合优度较高。由于 Sig=0.000<0.005，因此线性关系显著。而拟合成二次方时，拟合度很低，所以排除二次函数的可能，因此我们选择一次函数。

曲线拟合：

模型汇总和参数估计值							
因变量: 出错人数							
方程	模型汇总					参数估计值	
	R 方	F	df1	df2	Sig.	常数	b1
线性	.735	61.128	1	22	.000	-10.286	.131
自变量为 访问人数。							

拟合的图像为：

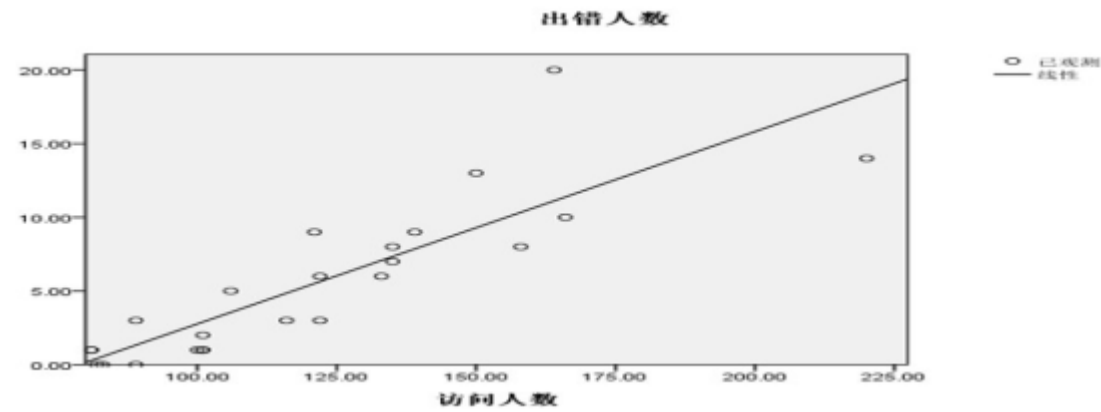


图 9

最终得出每秒申请入列次数 x 和错误次数 y 的函数关系为：

$$y=0.131x-10.286$$

又因为当 $y=0$ 时， $x \approx 78.5$ ，

根据数据得：

当 $x < 78.5$ 时，基本不出错，所以 $y=0$

综上得： $y = \begin{cases} 0 & x < 78.5 \\ 0.131x - 10.286 & x \geq 78.5 \end{cases}$

则错误率为：

$$y/x = \begin{cases} 0 & \\ (0.131x - 10.286)/x & \end{cases}$$

令错误率小于 0.1%，即 $(0.131x - 10.286)/x < 0.001$
得：

$$x < 79.1$$

由于 x 为人数，所以 x 只能取整数。

$$\text{即： } x \leq 79$$

即每秒申请入队人数小于等于 79 人，该队列的错误率就低于 0.1%。

根据 12306.cn 的实际分时起售车站情况，估计这一天的订票请求量大部分分布在这是几个时段内。

因此每个时段的订票请求量为 $1000 \text{ 万}/10=100 \text{ 万}$ ，每个时段大约有一个小时有过高的访问量，由于处于春运高分期，认为每个时段中访问量一直很高，则每个时段平均每秒的订票请求量为：

$$1000000/3600 \approx 277.8 > 79$$

这是模拟在一个队列中的情况，因为 $277.8 > 79$ ，所以需要增加队列数以减少每个队列的出错率，共需增加的队列数为：

$$277.8/79 \approx 3.5$$

则要是每个队列的出错率小于 0.1%，需要有 4 个队列同时开放。

根据排队等待时间不超过 30 分钟求队列长度：

由题目给出的数据得，高峰时段每个号码出队列的时间平均为 0.012s。

根据模型假设第 5 条得：

当队列排队时间为 30min 时，队列的长度为：

$$30 \times 60 / 0.012 = 150000$$

又因为上面已经算出每个时段大概有 100 万个订票请求量，所以要使队列的最长排队时间不超过 30min，需要的队列数为：

$$1000000 / 150000 \approx 6.7$$

即大概有 7 个队列就可以使队列排队时间不超过 30min。

由以上的结果可得：

要使队列错误率低于 0.1%，需要 4 个队列；要使队列排队时间不超过 30min，需要 7 个队列。而队列越多，错误率越低，队列排队时间越少，所以要同时满足这两个条件，则至少需要 7 个队列。

4.3 问题三

4.3.1 分时时段优化模型

根据 12306.cn 现行的分时售票策略，可以发现车站已经分为 10 个时段出售车票，由于题目中要求分时时段数不能超过 10 个，所以不能通过增加分时来进行优化。

但是通过通过观察 12306.cn 的关于调整部分车站互联网、电话订票起售时间的公告[2]（该调整从 1 月 26 日开始实施），由图 10，可以发现：

12: 00起售车站

阿克陶、阿图什、鞍山、巴楚、坂田、博乐、布列开、常州、潮州、大埔、丹东、丹阳、东莞、丰顺、佛山、福海、抚顺北、阜新、高州、格尔木、广元、广州、广州北、广州东、汉口、和静、和平、和什托洛盖、河源、菏泽、红果、华城、惠州西、揭阳、晋城、靖河、精河南、景德镇、克立玛依、库车、乐昌、雷州、龙川、轮台、玛纳斯湖、茂名东、梅州、绵阳、墨玉、南京、尼勒克、盘锦、皮山、坪石、曲靖、三水、沙湾县、莎车、鄯善、上饶、韶关东、神木、石河子、疏勒、绥芬河、天津、吐鲁番、乌西、五五、新和、新兴县、信宜、兴宁、徐闻、宣威、焉耆、阳春、叶城、伊宁东、英德、英吉沙、鹰潭、潯潭、泽普、湛江西、肇庆、镇江、中华门。

13: 00起售车站

安化、安庆、安庆西、亳州、长春、长沙、长治北、常德、巢湖、郴州、辰溪、春湾、慈利、低庄、东方、敦煌、阜南、阜阳、哈尔滨、海口、汉寿、汉中、合肥、合肥西、衡山、衡阳、怀化、淮南、黄梅、会同、吉首、济南、嘉峪关、靖州、耒阳、冷水江东、澧县、道源、临澧、临湘、六安、娄底、庐江、路口镇、麻阳、猛洞河、汨罗、宁乡、祁东北、祁阳北、全椒、三堂集、三亚、韶山、邵东、邵阳、石家庄、石家庄北、石门县、舒城、水家湖、松桃、宿松、太湖、太原、天柱山、通道、桐城、铜仁、乌海西、武汉、厦门、厦门高崎、湘潭、湘乡、新化、新晃、淑浦、叶集、岳阳、永州、岳阳、张家界、重庆北、株洲。

15: 00起售车站

滨江、达州、大理、德州、赣州、广水、哈尔滨东、海湾、邯郸、吉安、吉林、集宁南、嘉善、井冈山、昆山、丽江、芦港、麻城、内江、南宁、商丘、上海、上海南、十堰、松江、苏州、威海、无锡、西宁西、香坊、信阳、延安、昭通。

18: 00起售车站

阿城、阿尔山、阿金、阿里河、阿龙山、阿木尔、安达、安广、安家、安口窑、安龙、安陆、安平、安顺、安塘、安固、安溪、昂昂溪、敖力布告、八达岭、八角台、八面城、八面通、八仙洞、巴东、巴林、巴山、巴彦高勒、巴中、霸州、白壁关、白河、白河东、白河县、白河、白奎堡、白狼、白泉、白沙、白石山、白水江、白音察干、白音胡硕、白音地拉、白银山、白银山、白云鄂博、百色、柏果、蚌埠、宝坻、宝拉格、宝林、宝龙山、宝泉岭、保定、保康、北安、北戴河、北海、北京东、北流、北马圈子、北票南、北台、北营、臂膀河、贵红、本溪、本溪湖、笔架山、泊头、勃利、博克图、博兴、布海、蔡家沟、蔡家坡、沧口、沧州、苍石、曹县、草海、草河口、草市、册亨、岑溪、查布嘎、蔡家河、慈江、柴沟堡、柴河、昌乐、昌黎、昌平、昌平北、昌图、长春南、长甸、长葛、长岭子、长山屯、长寿、长汀、长汀镇、长阳、长垣、长征、长治、朝阳、朝阳川、朝阳地、朝阳镇、辰清、陈相屯、晨明、成都南、成高子、成吉思汗、承德东、城固、城阳、池州、赤壁、赤峰西、崇仁、崇左、滁北、楚山、楚雄、创业村、春阳、磁山、磁县、磁窑、嵯岗、达家沟、达拉特旗、达拉特西、大安、大安北、大巴、大板、大堡、大成、大关、大官屯、大红旗、大虎山、大林、大陆号、太平房、大青沟、大石桥、大石头、大石寨、大屯、大武口、大兴、大兴沟、大雁、大杨树、大英东、大营、大营镇、大营子、大竹园、代县、岱岳、带岭、丹凤、当阳、刀尔登、到保、道清、遼州、得耳布尔、德安、德伯斯、德昌、德惠、德阳、灯塔、登沙河、邓州、低窝铺、滴道、甸心、定边、定南、定陶、定西、定襄、定州、东安、东边井、东二道河、东方红、东丰、东光、东海、东津、东京城、东来、东明村、东明县、东胜、东通化、东乡、东辛庄、东营、东源、东镇、东至、东庄、都匀、豆罗、独立屯、独山、杜家、对青山、兑镇、敦化、峨边、峨边、额济纳、鄂州、二道湾、二连、二龙、二龙山屯、二密河、发耳、繁峙、范家屯、防城、防城港、费县、分宜、汾阳、丰城、丰城南、丰乐镇、丰镇、凤陵渡、冯屯、凤凰城、凤县、凤阳、凤州、扶绥、扶余、涪陵、福利屯、福泉、抚宁、抚顺、抚远、抚州北、富川、富海、富锦、富拉尔基、富县、富县东、富裕、富源、嘎什甸子、盖州、干沟、甘谷、甘河、甘洛、甘旗卡、甘泉、甘泉北、赶水、皋兰、高碑店、高村、高老、高密、高平、高桥镇、高山子、高台、高滩、高邑、葛城、葛镇堡、葛根庙、蛤蟆塘、根河、工农港、公庙子、公营子、公主岭、珙义、共青城、沟帮子、孤家子、花园、古城镇、古东、古文、古浪、古莲、古田、谷城、固安、固始、固原、固镇、瓜州、关林、官厅、官厅西、官字井、冠豸山、灌水、光泽、广安、广德、广汉、广宁寺、广通、归流河、贵定、贵定南、贵港、贵溪、郭家店、郭磊庄、果松、哈拉海、哈拉苏、海北、海城、海林、海龙、海伦、海坨子、海阳、寒葱沟、寒岭、汉沽、汉阴、汉源、浩良河、合川、合阳、和龙、河边、河唇、河津、河口南、贺州、鹤北、鹤壁、鹤岗、鹤立、鹤庆、黑河、黑井、黑水、黑台、横道河子、横峰、衡水、红安、红安西、红花沟、红山、红寺堡、红兴隆、红星、红彦、

.....

图 10

其中 14:00 没有起售车站，也就是说 13: 00 起售车站后 2 个小时内没有新的起售车站，除了 14:00，其它分时时段（从 8:00 到 18:00）相差基本都是 1 个

小时，而且起售的车站的数量相差的不多，但是 18:00 有大量的起售车站。12306.cn 网站的开通时间是每天的 7:00 到 23:00[3]，也就是说 18:00 到 23:00 中间没有新的起售车站。

我们会发现，18:00 相对于其它分时段，起售车站的数量远远大于其他时段的起售车站，这样势必会造成 18:00 点有大量的订票请求进入队列，从而导致错误率激增和排队时间的增加，不利于高效的售票。

我们可以这么优化：既然 13:00 到 15:00 之间有两个小时没有新增的起售车站，我们可以将 18:00 起售的车站中的一部分分配到 13:00 起售车站中。通过这样来使起售车站的分配更加均衡，进而可以更好的分散了网站全天的请求量，减轻了同一时间段并发访问过大的问题。

4.3.2 模型的优化

4.3.2.1 模型分析：

对系统的优化，可以看作是两个指标的优化，即：分散访问量；降低系统内存浪费。而系统内所分队列的多少也会对这两个因素产生影响。分时段数量的变化以及队列数量的变化，都可以对系统的性能产生影响。

通过分析，分时段与队列数量可以建立数学模型。因此我们通过建立模型，分析系统性能，找到一个最优的方案。

4.3.3.2 模型的假设：

- (1) 假设总的访问量按照时段平均分配。
- (2) 假设总体访问量 $N=1000$ 万。

4.4.2.3 模型建立与求解：

设分时段段的个数为 x ($x \leq 10$)，系统内所分队列的个数为 y 。

每个队列的等待时长不能超过 30 分钟，每个数据的出队平均时间为 0.012s。所以 $30 \times 60 / 0.012 = 150000$ 。由假设 (1)，可知每个队列的长度最大为 150000。

可以通过 $\frac{N}{150000x}$ 计算出系统队列个数的理论的最小值，得到关于系统内队列的个数 y 的不等式：

$$\frac{N}{150000x} \leq y$$

由 $N=1000$ 万，得到不等式如下：

$$xy \geq 67$$

由题意可知 $x \leq 10$ ，根据火车票售票官网信息[5]显示：售票节点是由原来 4 个调整到了 10 个，就说明系统在节点为 4 或 4 个以下时会使系统造成阻塞，或队列数量无法满足超大访问量的需求。所以 4 个或 4 个以下的节点不予考虑。

由于 x, y 均是大于零的整数，所以可以将 x 进行分类，当队列的个数 y 取得最小值（满足 $xy \geq 67$ ）时，系统对内存的利用率最大。

$$y \geq \frac{67}{x} \quad (\text{取满足不等式 } y \text{ 的最小值})$$

由于每个队列的入列速率 V_λ 不能超过 79 人数/秒，则可以看作当每个队列的入列速率 V_λ 为 79 人数/秒时队列的利用率为 1。

则计算系统内存利用率的公式如下：

$$V_\lambda = \frac{N}{3600 * xy} \quad (\text{入队速率 } V_\lambda)$$

$$\eta = \frac{V_\lambda}{79} \quad (\text{系统内存利用率})$$

将两式合并的得：

$$\eta = \frac{N}{3600 * 79 * xy}$$

通过计算得到数据表 7，如下：

x 的取值	y 的取值	系统内存的利用率
5	14	50.2%
6	12	48.8%
7	10	50.2%
8	9	48.8%
9	8	48.8%
10	7	50.2%

表 7

根据结果进行分析得出：

分时时段为 10，队列个数为 7；分时时段为 7，队列个数为 10；分时时段为 5，队列个数为 14。系统内存的利用率最大。

但考虑到分段时段不能过少，和队列不宜过多，所以排除分时时段为 5，队列个数为 14 的方案。

当每个时段开始售票的初始时段，入队速率 V_λ 会达到一个比较大的值。

因此，应该加强处理短时间内入列速率 V_λ 急剧增加的能力。而多个队列的平摊，无疑是可以加强处理短时间内入列速率 V_λ 急剧增加的能力最有效的方法。

因此，最优的方案为：分时时段为 7 个 队列个数为 10 个。

4.4 问题四

给 12306 网站的建议书

介于 12306 网站在最近几年春运期间的表现,在很多方面令广大用户不满意。还有很多地方需要进行改进,在此我们提出一些自己的想法,可能我们的一些建议不是很合理,但是我们愿为系统性能的提升尽自己微薄的力量,如有不道之处,敬请谅解。

我们的建议如下:

建议一:

当我们访问 12306.cn 网站,来到主界面后发现,目前的导航栏和重要位置字体过小,布局也不合理,应该将用户的核心操作放在关键位置, 预订车票的按钮位置不突出、不明显,需要用户思考,如下图 11[3]所示



图 11

建议二:

如下图所示,登陆后起始页操作冗余,没必要让用户再点一次。可以直接到车票预定页,如果有未完成订单就直接显示未完成订单页。

欢迎您登录中国铁路客户服务中心网站。

如果您需要预订车票,请您点击[车票预订](#)。

图 12[3]

建议三：

目前用户进入车票预定页面所默认的起始时间是当日，其实在春运期间，可以默认的是最早可订到车票的日期，更好地服务那些急于订票的用户。

建议四：

在 12306 的乘车人姓名、座位席不要在预定了再选，建议预先选定，并且自动记忆，这样用户在上次没买到的情况下，由于系统还记得用户信息，直接点击确认就可以继续购票。

建议五：

往返车票的订购需要进行两次预订的流程，乘车人，输入两次验证码，进行两次验证，建议可以合并为一次预订

车次	查询区间			余票信息											购票
	发站	到站	历时	商务座	特等座	一等座	二等座	高级软卧	软卧	硬卧	软座	硬座	无座	其他	
K7742	保定 22:02	北京 00:01	01:59	--	--	--	--	--	无	无	--	有	无	--	预订
K220	保定 22:22	北京西 00:08	01:46	--	--	--	--	--	12	有	--	有	有	--	预订
G616	保定东 22:28	北京西 23:09	00:41	16	--	有	有	--	--	--	--	--	--	--	预订
G528	保定东 22:33	北京西 23:14	00:41	--	15	有	有	--	--	--	--	--	--	--	预订
K7726	保定 22:34	北京西 00:21	01:47	--	--	--	--	--	5	--	无	有	--	--	预订
G668	保定东 22:39	北京西 23:20	00:41	--	14	有	有	--	--	--	--	--	--	--	预订

图 13[3]

建议六：

网上购票流程，查询-》预定-》确认-》完成[3]，这样的操作逻辑更适合售票员，不适合网民的习惯，完全是照搬实体窗口售票。建议做一下数据挖掘，把用户常买的车次记录下来，然后用户登录后，就应该显示用户最常买的那些车次。

建议七：

目前并没有针对弱势群体购票的专区，特殊票种的购票流程也很复杂繁琐，应该在明显的地方为学生等特殊群体做单独区分，方便特殊用户查找。

建议八：

由于 12306 在节假日买票具有高并发性，可以引导用户不要在出行前几天再去买票。在主流节假日前可以考虑允许提早 30 天买票，并且在价格上给以一定的优惠政策，来鼓励大家提早买票。把短时间的并发分散到这 30 天，让系统瞬时负荷峰值尽可能的下降。

建议九：

现行的排队方法在春运高峰期的排队时间过长，用户在屏幕前很难一直盯着屏幕等着买票，所以可以在即将排到票时给用户发个短信提示其准备购票。

我们能想到的建议就是以上这些了，也许并不是很专业，但是这是我们思考的结果，如有不到之处，敬请谅解。谢谢！

五、模型的评价与推广

5.1 模型的优点

- (1) 对于入队列速率和出队列速率减与时间的关系,以及错误率与入队列速率,我们均采用了多项式拟合的方法,简化问题,简单易懂。避免从技术角度对队列性能的研究。更简单的求出队列的数据处理能力,错误率和所需队列数。
- (2) 综合运用了理论推导,建立模型,并借助不同的工具软件,增加了可信度。
- (3) 所建立的模型在各自研究的时段内是有效的。
- (4) 我们充分考虑了网站的日访问量的实际的情况,对访问量做了合理计算。

5.2 模型的缺点

- (1) 由于测试数据的由来,并不是在连续的时间内得出的,因此我们研究的只是分段区间上的队列的入队出队情况。这样结果势必会有一定的误差。
- (2) 由于对函数类型种类了解不够全面,可能还存在更好的函数模型。
- (3) 由于模型建立与求解时所用的数据,均来自于实验测试数据,与真实情况会有一定偏差。

5.3 模型的推广

随着科学技术的不断发展,网络购票是必会成为众多人的首选购票方式。如何利用网络购票的方式,更好的给用户带来便利,已成为现在人们研究的热点。

由于局部范围内入队速率和出队速率对队列的性能的影响,均存在最优值。而我们对队列的数量的确定,以及对现行排队方法和分时策略的优化,均对实际情况进行了充分的考虑得而出的。因此具有可行性。

相信通过对火车票购票网站的进一步研究,一定会更好的处理现行网站存在的问题,并进一步减少火车票网上购票出现的各种问题。

参考文献：

- [1]网站日累计访问量参见：http://alexa.chinaz.com/Show_Cache.asp?Url=12306.cn
- [2]关于调整部分车站互联网、电话订票起售时间的公告：
http://www.12306.cn/mormhweb/zxdt/tlxw_tdbtz56.html
- [3]12306 网站介绍：<http://baike.baidu.com/view/3188461.htm>
- [4]姜启源，谢金星，叶俊 高等教育出版社 《数学模型第四版》2011 年
- [5]火车票购票分时节点调整：
http://www.12306.cn/mormhweb/zxdt/tlxw_tdbtz53.html