

2017 高教社杯全国大学生数学建模竞赛

承 诺 书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》(以下简称为“竞赛章程和参赛规则”,可从全国大学生数学建模竞赛网站下载)。

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章程和参赛规则的,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,严格遵守竞赛章程和参赛规则,以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为,我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会,可将我们的论文以任何形式进行公开展示(包括进行网上公示,在书籍、期刊和其他媒体进行正式或非正式发表等)。

我们参赛选择的题号是(从 A/B/C/D 中选择一项填写): 测试一

我们的参赛报名号为(如果赛区设置报名号的话): 2017015

所属学校(请填写完整的全名): 长江大学

参赛队员(打印并签名): 1. 孔庆亮

2. 陈梓欣

3. 王张弛

指导教师或指导教师组负责人(打印并签名): 数模指导组

(论文纸质版与电子版中的以上信息必须一致,只是电子版中无需签名。以上内容请仔细核对,提交后将不再允许做任何修改。如填写错误,论文可能被取消评奖资格。)

日期: 2017 年 8 月 4 日

赛区评阅编号(由赛区组委会评阅前进行编号):

2017 高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅编号(由赛区组委会评阅前进行编号):

赛区评阅记录(可供赛区评阅时使用):

评 阅 人										
评 分										
备 注										

全国统一编号(由赛区组委会送交全国前编号):

全国评阅编号(由全国组委会评阅前进行编号):

公交车调度

摘 要

本题的数据有一条线路在某一天的上下行中的所有站上下车人数的采集表,让我们确定一个便于操作的全天(工作日)的公交车调度方案。表格数据是离散的,想要使用数据相当不容易,我们使用 MATLAB 对表格数据进行初步的处理,得到每个时间段、位于每个站点公交车载有的乘客数量,对每个时间段可以寻找到每个时间段的最大载客量,将其看作均匀分布,即可采用三次样条插值法得出最大载客量的连续分布;对这条连续曲线进行分析:利用最大载客量的平均值,可以得到上行方向上的早高峰是 6 00 到 9 00,下行方向的早高峰是 7 00 到 10 00;利用 MATLAB 令发车间隔 t 从十不断减少一,使最大载客量插值线 $f(t)$ 在每个时间段的发车间隔 Δt 的变化不超过 120% 的满载率,利用搜索算法,可以求得不超过 120% 满载率的在每个时段都最大的发车间隔,从而确定发车时间表,

关键词: 三次样条插值、最优化搜索

一 问题重述

为了完善城市交通,改善居民出行状况,提高公交公司的经济与社会利益。基于某条公交线路客流调查和运营资料的数据,设计一个在这条线路上便于操作的全天公交车调度方案,包括两个起点站的发车时刻表;一共需要多少辆车;这个方案以怎样的程度照顾到了乘客和公交公司双方的利益,并把这个调度问题抽象成一个明确、完整的数学模型。

二 建模分析

2.1 问题分析

题中所给数据统计了一个工作日两个运行方向各站上下车的乘客数量,我们将对这组数据进行具体的分析处理,得到每一时间段所有公交车在某两个站点之间的累计人数,由此初步确定该时间段首发站要派出的公交车数量的大体范围。

2.2 模型假设

1. 首发站都在五点钟发第一班车;
2. 所有的公交车全程匀速行驶;
3. 公交车上乘客数量的变化,不会影响公交车的速度;
4. 假设表上所给数据能反映该段线路上的日常客流量;
5. 公交车到达终点站时,所有的乘客都会下车;
6. 乘客上行或是下行,无论经过几个站,车票价一定;
7. 公交车会按调度表准时到站和出站;
8. 公交车匀速行驶,不会出现故障,忽略公交车的停站时间
9. 每辆车经过各个车站时不会留有乘客。

定义 1 最大载客量 每一时间段载客量的最大值。

2.3 记号说明

2.4 建立模型

由于两条路线的对称性,这两条路线可以按相同的方式处理,因此我们对问题分析后,我们建立如下目标优化模型:

$$\max m_j \frac{T}{60} + m_i \frac{T}{60}$$

模型记号说明	
p_{max}	最大载客量
T	公交车走完全程所花的时间
m_i	第 i 时间段首站 A13 发配的车量
n_i	第 i 时间段首站 A0 发配的车量
a_i	第 i 时间段首站派出的车量
t_i	第 i 时间段的发车间隔

表 1: 模型记号说明

$$s.t. \begin{cases} \frac{a_i}{m_i} \leq 120 & i = 1, 2 \dots 18, j = 1, 2 \dots 13 \text{ or } 14 \\ \Delta t \leq \begin{cases} 10 & \text{非高峰时时间段} \\ 5 & \text{高峰时间段} \end{cases} \end{cases}$$

2.5 模型求解和分析

2.5.1 模型的求解

第一步通过对本题所给数据的分析,我们将全天的行车时间依照原来的数据按小时分为 24 段,得出每个时间段原始数据统计的公交车上每两个站点之间的载客量,第二步求出载客量的最大值,据此判断出每个时间段首站派出公交车数量的下限。乘客在非高峰期等待的时间一般不超过 10 分钟,在高峰期等待的时间一般不超过 5 分钟,从而知晓各个时间段首站派出车量的最小值作为硬性条件处理第三步由第二步确定各个时间段发车时间间隔和以及车次后,首先算出首发站到终点站的距离,,进而具体确定发车具体时刻。

2.5.2 结果分析

结果比实际值会偏大

2.6 模型评价

没有考虑车次跨时段的情况,每个时间段每站经过的车次数都是不一样的,而本模型直接忽略了这一点,存在很大的漏洞每个站经过的车次应该尽可能的细化。

2.6.1 模型优点

1) 模型简单、高效

2.6.2 模型缺点

1) 以最大载客量为基准,得出来的车次偏大,对公司是比较不利的。

2.7 模型改进

1. 考虑跨时段车次,按各站到首战的时间具体考虑各时间段发配车次。

2. 考虑乘客等车时间,并把该因素结合到乘客满意度中。
3. 按小时分段不一定最合理,可以采用聚类分析将其进一步分类。经检验,下行方向上车人数与下车人数不符,可以通过延长某日调查时段范围全面调查数据。

参考文献

附 录

0.1 MATLAB 源代码

```
clc, clear all
```

```
% 读取文件
```

```
% 如要修改, 将***.xlsx文件路径填入, 或者放在此m文件相同目录下, 使用xlsread('***.xlsx')
```

```
data1 = xlsread('D:\Mcm\Test1\上行上下车.xlsx');
```

```
data2 = xlsread('D:\Mcm\Test1\下行上下车.xlsx');
```

```
% 上行为1, 下行为2
```

```
% 上车下车人数分别存储在矩阵 up 和 down 中
```

```
for i = 1:2:36
```

```
    up1((i+1)/2,:) = data1(i,:);
```

```
end
```

```
for i = 2:2:36
```

```
    down1(i/2,:) = data1(i,:);
```

```
end
```

```
for i = 1:2:36
```

```
    up2((i+1)/2,:) = data2(i,:);
```

```
end
```

```
for i = 2:2:36
```

```
    down2(i/2,:) = data2(i,:);
```

```
end
```

```
% 每站上车人数与下车人数之差
```

```
A1 = up1 - down1;
```

```
A2 = up2 - down2;
```

```
B1 = max(A1');
```

```
B2 = max(A2');
```

```
% 每个时间段需要的车辆数下限
```

```
m1 = B1 ./ 120;
```

```
m2 = B2 ./ 120;
```

```

% 车辆数有小数部分处理
m1 = floor(m1 + 1);
m2 = floor(m2 + 1);
% 至少发6辆车
m1(find(m1 < 5)) = 6;
m2(find(m2 < 5)) = 6;

save('m1','m1');
save('m2','m2');
% 寻找一天所要的最少车辆数
max(m1)
max(m2)
%save( 'upHt.met','E');
%save( 'downHt.met','E');

% 每段时间的发车时间间隔
Dt1 = 60 ./ m1;
Dt2 = 60 ./ m2;

% 保存
save('Dt1','Dt1');
save('Dt2','Dt2');

`clc, clear all

% 读取文件
data1 = xlsread('D:\Mcm\Test1\上行各站距离.xlsx');
data2 = xlsread('D:\Mcm\Test1\下行各站距离.xlsx');

t1 = data1 ./ (20 ./ 60)
t2 = data2 ./ (20 ./ 60)

```



```

T1 = sum(t1)
T2 = sum(t2)

clc,clear all

load('m1')
load('m2')
load('Dt1')
load('Dt2')

temp1 = 0;
for i = 1:length(Dt1)
    num1(i) = 60 ./ floor(Dt1(i));
    if isinteger(num1(i))
    else num1(i) = round(num1(i));
    end
    temp1 = (60 + temp1) - num1(i) .* Dt1(i);
end

temp2 = 0;
for i = 1:length(Dt1)
    num2(i) = 60 ./ floor(Dt2(i));
    if isinteger(num2(i))
    else num2(i) = round(num2(i));
    end
    temp2 = (60 + temp2) - num2(i) .* Dt2(i);
end

num1
num2

```