



2017 湖南省研究生数学建模竞赛参赛承诺书

我们仔细阅读了湖南省研究生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。

我们授权湖南省研究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（从组委会提供的试题中选择一项填写）：

我们的队号为（填写完整的队号）：201718001002

所属学校（请填写完整的全名）：国防科学技术大学

参赛队员（打印并签名）：

1. 王佳雨
2. 王一雪
3. 王鑫

指导教师或指导教师组负责人(打印并签名)：王丹

日期：2017 年 4 月 24 日

评阅编号（由组委会评阅前进行编号）：

2017 湖南省研究生数学建模竞赛

编号专用页

评阅编号（由组委会评阅前进行编号）：

评阅记录（可供评阅时使用）：

[illegible]

湖南省第三届研究生数学建模竞赛

题 目 出租车合乘业务系统设计

摘 要：

在出租车合乘业务中，为充分调动乘客、司机等各方面的积极性，需要设计出合理的合乘业务系统。这种系统的设计涵盖乘客分配、车辆行驶路线优化以及车费计算方法等诸多方面的内容，期望能够保障司机和乘客双方的利益。

在本文中，我们设计了一种新颖的乘客与车辆的分配方案。因为每个出租车上最多只能乘坐 3 人，最少可以乘坐 0 人。该方案将编号为 1 到 1001 的乘客打乱顺序后再通过随机抽取个数的方法，抽取数量 0、1、2、3 中的任意一个随机数，将相应人的个数分配给车辆。相当于将 n 个小球随机放入 M 个箱子中的方法。思想上简单易懂，实现起来也很方便。

此外，我们使用遗传算法实现了车辆最短路径的选择问题。经过选择、交叉、变异、迭代逐步，并且根据每个乘客的出行路径产生出初始种群。利用 Floyd 算法结合遗传算法计算出每一辆车的最优路径，将这些所有车的路径相加就是一次人车匹配后获得的最短路径即所用时间最少的路径。将乘客和车辆按照上面描述的匹配方法进行 100 次随机匹配，从其中选出最短的总路径作为最后的实验结果。

关键字：车辆合乘；遗传算法；路径优化；乘客匹配

出租车合乘业务系统设计

1. 问题重述与分析

现有的两种出租车合乘模式，即“相同起点模式”和“一口价模式”，由于合乘条件难以满足，可能出现绕行耽误乘客时间，或者无法构成合乘影响司机收入等原因，存在一些问题。因此，题目希望设计出更优化的合成模式，以充分调动乘客、司机等各方面的积极性。

通过此次建模尝试解决以下问题：

问题1：请设计高效的合乘方案及其相应算法，使乘客等待时间尽量短，所需出租车数量尽量少。

分析：当所调动的出租车数目越少时，意味着在乘客数量一定的前提下，每辆出租车需要接送的乘客数量越大，这势必会造成乘客的等待时间变长。因此在“乘客等待时间”和“所需出租车数量”之间需要做出平衡；

另外，需要说明的是，在接下来的模型中，我们讨论的是乘客从呼叫出租车开始，到实际到达目的地这一过程中的全部时间，希望这个时间尽可能短。因为单纯希望乘客等待出租车的时间尽量短有可能会造成出租车司机“优先接客”的绕行行为。

问题2：请设计与合乘方案相应的合理的车费计算方法。

分析：更改了合乘模式，相应地必须设计出较为合理的车费计算方法，保障司机和乘客双方的利益。在新的合乘模式中，乘坐同一辆出租车的各路乘客上车位置不同，下车位置也不同，因此较为理想的计费方式是：单独乘车区间的车费由各乘客单独支付；共同乘车区间的车费，各乘客均按一定的百分比支付。总的原则是：对乘客而言所付车费应小于等于单独乘车到目的地的车费；对司机而言收入应大于等于只载一位乘客时的车费。

问题3：假设某城市的路网为正方形网格，网格边长500米，道路均可双向行驶。附件1是该城市某日某时刻之前3分钟内的打车需求数据，附件2是当前空驶出租车的位置信息，请按照你们设计的合乘方案给出具体的计算结果，并按附件3中指定的格式输出结果。假设，附件2中未涉及的出租车在计算合乘方案时不可用（非空驶，不考虑），并且合乘方案不要求用到附件2中的全部出租车，即，允许部分出租车仍保持空驶状态。乘客等待时间均以当前时刻为0时刻开始计时，即，不考虑之前3分钟内已经等待的时间。

分析：这一问需要我们根据所提供的数据计算并检验所设计的合乘方案能否实现预期目标，同时分析实现效果的优劣。

2. 假设和说明

(1)、题目中所给的一条假设是：假设不需要考虑对公司现有调度平台和手机打车软件的改造难度。

(2)、假设所有乘客的打车信息同时发布，所有车辆随时待命，等待系统分配乘客。

(3)、假设在行车过程中乘客分配不再发生变化，即每辆出租车司机在行车开始便已知晓自己此行需要接送哪几位乘客。

(4)、在司机和乘客双方达成乘坐协议后，双方均不再对行车路线抱有异议。

(5)、不考虑道路拥堵和司机目前的行驶方向。

(6)、司机一直保持匀速行驶状态，不将停车等时间计算在内。

3. 模型的建立、求解和分析

从出租车合乘方案设计中，可以分析出几个关键问题：乘客如何分配；每辆出租车的行车路线如何做到最优；目标函数的选取、设计以及最优解的计算方法等。下面各小结将对上述问题逐一进行探究。所建模型适用于所有出租车合乘问题，使用时只需将程序所需数据导入即可。在这里我们分析时采用的是问题三中给出的出租车和乘客的数据。

3.1 乘客分配问题

在合乘问题中，乘客人数多于出租车数量，在每位乘客独立发布自己的行程路线后，系统应能够根据出租车及乘客当前的位置信息，对乘客和车辆进行合理分配。

本文的分配方案是：设定每辆出租车内可随机分配0~3名乘客，将所有打车乘客位置随机排序，依次按照随机分配结果向总数为M的出租车内分配乘客，于是乘客将被顺序分配至一辆又一辆的出租车内。程序为Distribution.m。

上述程序的计算结果“R矩阵”即为计算所得的分配方案。它是一个664*100的矩阵，如图一所示。

	1	2	3	4	5	6	7	8	9	10	11
1	[609,23.55...	[609,23.55...	[609,23.55...	[609,23.55...	[609,23.55...		[609,23.55...	[609,23.55...	[609,23.55...	3x5 double	3x5 double
2	3x5 double		[123,21.33...				[123,21.33...			[448,22.25...	[448,22.25...
3	3x5 double	[452,24,27....	[448,22.25...	[123,21.33...	[123,21.33...	[609,23.55...	[872,21.20...		3x5 double		3x5 double
4	[174,20.50...	3x5 double		[872,21.20...	3x5 double			[452,24,27....	3x5 double	[872,21.20...	[746,20.50...
5					[99,19.500...	[452,24,27....	[772,24,27....		[174,20.50...		[333,22.87...
6	[333,22.87...	[99,19.500...	3x5 double	[99,19.500...	3x5 double		[99,19.500...	[123,21.33...	3x5 double	3x5 double	
7	[635,21.50...		[746,20.50...	[746,20.50...				[872,21.20...	[396,21.87...	3x5 double	
8		[746,20.50...		[333,22.87...		3x5 double	3x5 double			3x5 double	[635,21.50...
9		3x5 double	3x5 double	[556,22.74...	[635,21.50...		[556,22.74...	3x5 double			[556,22.74...
10	[671,21.22...	[671,21.22...	[671,21.22...	[671,21.22...		[609,23.55...		[174,20.50...		3x5 double	

图一 人车分配结果显示

其中每一列表示667辆出租车中顺序第一辆车、第二辆车、第三辆车……中的乘客分配情况，如R{1, 1}为：

客人编号	初始位置 x/km	初始位置 y/km	目的地位置 x/km	目的地位置 y/km
970	22	26.5633	23.9131	27
864	20.8469	25	21.8440	25.5000

表示第970位和第864位乘客被分配到第一辆车；

R{2, 2}是个空矩阵，表示没有乘客被分配到第二辆车。

每一行表示新一次的分配结果，即这样的人车分配计算我们进行了100次。

3.2 路径优化问题

不同乘客的上、下车位置均不同，司机为实现合乘目标，势必会比单独接送一位乘客时行驶过更为复杂的路线，这便存在绕行的隐患。因此，需要设计出较为合理的行车路径，使得对每一位乘客而言，绕行路程在可以接受的范围内。在此使用了遗传算法和Floyd对路径进行优化。在尽量不绕路的情况下，走最短的路径并保证把每位乘客送达目的地。之前我们假设司机是匀速行驶，路径最短情况在匀速行驶下相当于时间最短，时间短对于用户来说相当于花费少。所以我们选择路径来做优化。具体编程见附件min.m。

3.3 目标函数的选取与计算

我们把从乘客呼叫出租车开始，到实际到达目的地这一过程中的全部时间设为目标函数，希望这个时间尽可能短。目标函数中的时间成本由系统中全部合乘出行者的时间成本共同构成。这里的时间成本主要指的是车辆运行中的速度^[2]：

$$\sum_{k \in K} \sum_{u \in H} \sum_{v \in H} \sum_{i \in G} \sum_{j \in G} x_{ij}^k t_{ij} q_{uv}^k$$

式中：k属于K，K为车辆的集合；u, v属于H，H为乘客初始点和目的地的位置；i, j属于G，G为人车匹配中所有坐标点的集合；t为（i-j）的运行时间；q为u点上第k台车去v的客流量。

4. 结论

- （1）、通过编写程序，解决了乘客和车辆之间的分配问题。
- （2）、使用Folyd算法，能够计算出最短路径。
- （3）、使用遗传算法，计算了目标函数的最优解。

5. 模型的优缺点

5.1 优点

- （1）、将问题简化，考虑在比较简单的情况下实现合乘系统的基本功能；
- （2）、所采用的人车分配方法简单明了，因而相应的程序编写也较为直观。

5.2 缺点

- （1）、本文只适用于人车分配为静态的情形，且在车辆开始行使之后，分配不再发生变化，灵活性较差，导致行驶的路径可能比动态分配的行驶路径要长；
- （2）在人车分配的方法上，可以选择其他思路，尝试用其他算法实现；
- （3）编程过程中使用较多for循环，造成程序运行时间比较长，不适合作为APP等快捷方式来运行，编码还需要改进。

参考文献

- [1] 于匡员 基于预约模式的出租车合乘路径优化 哈尔滨工业大学硕士学位论文
- [2] 周和平，钟壁樯，彭霞花，夏西 出租车合乘路径选择与费率优化模型 长沙理工大学学报 Vol.8 No.1 Mar. 2011
- [3] 覃运梅，石琴 出租车合成模式的探讨 合肥工业大学学报 Vol. 29 No. 1 Jan. 2006

附录

Distribution.m

```
close all;clear all;clc;
load('*.mat');
ii=size(requests,1);%乘客个数
jj=size(taxi,1);%出租车个数
A=randperm(ii);%随机的“乘客个数”个数的排列（整数 1~ii）
requests1=requests(A,:);%随机排列的乘客
n=zeros(jj,100);
j=0;
while(j<100)%迭代次数为 100 次
    j=j+1;
    for i=1:jj
        B=0:1:3;
        C=randperm(4);
        n(i,j)=B(C(1:1));%随机生成 0~3 个乘客。
    end
    if sum(n(1:jj,j))~= ii
        j=j-1;%使得分配给 jj 辆车的乘客数量为 ii
    end
end
m=sum(n(1:jj,1:100));%验证给 jj 辆车是否安排的 ii 个乘客
for k=1:jj
    for l=1:100
        R{k,l}=requests1((sum(n(1:(k-1),l))+1):(sum(n(1:k,l))),1:5);
    end
end%为每辆车随机分配 0~3 位乘客
```

min.m

```
function [path,lmin]=ga(data,d) %data 为点集,d 为距离矩阵, 即赋权图
load('smallcase.mat')
tic
%=====
sj0=data;%开环最短路线
%=====
x=sj0(:,1);y=sj0(:,2);
N=length(x);
%=====
L=N; %sj0 的长度
w=800;dai=1000;
%通过改良圈算法选取优良父代 A
for k=1:w
c=randperm(L-2);
c1=[1,c+1,L];
flag=1;
while flag>0
flag=0;
for m=1:L-3
for n=m+2:L-1
if d(c1(m),c1(n))+d(c1(m+1),c1(n+1))<d(c1(m),c1(m+1))+d(c1(n),c1(n+1))
flag=1;
c1(m+1:n)=c1(n:-1:m+1);
end
end
end
end
J(k,c1)=1:L;
end
J=J/L;
J(:,1)=0;J(:,L)=1;
rand('state',sum(clock));
%遗传算法实现过程
A=J;
for k=1:dai %产生 0~1 间随机数列进行编码
B=A;
c=randperm(w);
%交配产生子代 B
for i=1:2:w
F=2+floor(100*rand(1));
temp=B(c(i),F:L);
B(c(i),F:L)=B(c(i+1),F:L);
```

```

B(c(i+1),F:L)=temp;
end;
%变异产生子代 c
by=find(rand(1,w)<0.1);
if length(by)==0
by=floor(w*rand(1))+1;
end
C=A(by,:);
L3=length(by);
for j=1:L3
bw=floor(1+fix(rand(1,3)*N)); %产生 1-N 的 3 个随机数
bw=sort(bw);
C(j,:)=C(j,[1:bw(1)-1,bw(2)+1:bw(3),bw(1):bw(2),bw(3)+1:L]);
end
G=[A;B;C];
TL=size(G,1);
%在父代和子代中选择优良品种作为新的父代
[dd,IX]=sort(G,2);
temp=[];
temp(1:TL)=0;
for j=1:TL
for i=1:L-1
temp(j)=temp(j)+d(IX(j,i),IX(j,i+1));
end
end
[DZ,IZ]=sort(temp);
A=G(IZ(1:w),:);
end
path=IX(IZ(1),:)
% for i=1:length(path)
% path(i)=path(i)-1;
% end
% path=path(2:end-1);
lmin=0;l=0;
for j=1:(length(path)-1)
t1=path(j);t2=path(j+1);
l=d(t1,t2);
lmin=lmin+l;
end
xx=sj0(path,1);yy=sj0(path,2);
plot(xx,yy,'r-o');
for i = 1:size(smallcase,1)
text(smallcase(i,1),smallcase(i,2),[' ' num2str(i)]);
end

```

```
axis equal  
toc
```

```

floyd.m
function [dist,mypath]=myfloyd(a,sb,db)
% 输入: a—邻接矩阵(aij)是指 i 到 j 之间的距离, 可以是有向的
% sb—起点的标号; db—终点的标号
% 输出: dist—最短路的距离; % mypath—最短路的路径
n=size(a,1); path=zeros(n);
for i=1:n
for j=1:n
if a(i,j)~=inf
path(i,j)=j; %j 是 i 的后续点
end
end
end
for k=1:n
for i=1:n
for j=1:n
if a(i,j)>a(i,k)+a(k,j)
a(i,j)=a(i,k)+a(k,j);
path(i,j)=path(i,k);
end
end
end
end
dist=a(sb,db);
mypath=sb; t=sb;
while t~=db
temp=path(t,db);
mypath=[mypath,temp];
t=temp;
end
return

```