contribution.m

```
disp('请输入评分集Q(即5*n的矩阵)和专家数n');
Q=input('Q=');
n=input('n=');
R=corrcoef(Q);%求出评分集Q的相关系数矩阵
[U,lamda]=pcacov(R);%求出相关系数矩阵的特征向量矩阵U和特征值
V=(lamda'/sum(lamda))*U';
F=V*Q';
A=F/sum(F);%各指标权重
Q=[Q(1,:)*(-1);Q(2,:);Q(3,:)*(-1);Q(4:end,:)];
disp('贡献度为')
alfa=A*sum(Q,2)/(5*n) %贡献度
```

change.m

```
function [ Mi ] = change(Mi, paths,Dj )%
i=size(paths,2);
z=5;
 xx=Dj(:,3);
while( paths(i-1)~=0)
    nm=find(xx==paths(i));
    op=find(xx==paths(i-1));
    %nm=paths(i);
    %op=paths(i-1);
    d1=Dj(nm,:);
    d2=Dj(op,:);
    if d1(1,2)== d2(1,2) %如果纵坐标相等，这意味着只能左右
        if d1(1,1)>d2(1,1)
            Mi(z)=1;%左为1
        else
            Mi(z)=2;%右为2

        end
    end
    if d1(1,1)==d2(1,1)%移动如果横坐标相等，这意味着只能上下移动
        if d1(1,2)> d2(1,2)
            Mi(z)=3;%下为3
        else
            Mi(z)=4;%上为4
        end
    end
    z=z+1;
     i=i-1;
end
Mi;
end
```

Floyd.m

```
function [ paths] = Flyod( sta,en,a)%sta为起点，en为终点，用paths记录起点到终点的最短时间的路径,a为各节点的距离矩阵
n=size(a);
%a(a==0)=inf;
%a(1:n+1:n^2)=0;
path=zeros(n);
for b=1:n
    for i=1:n
        for j=1:n
            if a(i,j)>a(i,b)+a(b,j)
                a(i,j)=a(i,b)+a(b,j);
                path(i,j)=b;
            end
        end
    end
end
i=1;%令paths(1)=0，帮助到达顶点时，以0为结束点
while( path(sta,en)~=0)%记录节点，从终点到起点倒过来记录
    i=i+1;
    paths(i)=en;
    en=path(sta,en);
end
 i=i+1;
    paths(i)=en;%将第一个转折点补录
    i=i+1;
        paths(i)=sta;%将起点补录
paths;
end
```

main.m

```
%在运行程序前需要进行三样数据准备，分别是节点间隔矩阵b,速度矩阵d,节点坐标矩阵Dj
%{现在利用开放后小区3的数据进行计算。先利用以下算式计算出tim和a矩阵的
%c=b;
%c=10*b;
%d=d/3.6;
%t=c./d;
%t(find(t==0))=inf;
%t=t+30*rand(9);
%t(1:10:9^2)=0;
%z=t./b;
%z(1:10:9^2)=0;
%}
clear
clc
tim=[
    0   NaN NaN NaN NaN 0.859091148 NaN NaN NaN
    NaN 0   NaN NaN NaN 1.832628509 NaN NaN 1.90367871
    NaN NaN 0   2.052194318 NaN NaN NaN 1.914113364 NaN
    NaN NaN 4.617542625 0   1.269501832 NaN NaN NaN NaN
    NaN NaN NaN 0.914542948 0   NaN NaN NaN 1.900425565
    2.022559573 2.141524696 NaN NaN NaN 0   1.339846262 NaN NaN
    NaN NaN NaN NaN NaN 4.487885049 0   3.587389222 NaN
    NaN NaN 3.090409776 NaN NaN NaN 3.499719802 0   1.796329296
    NaN 1.174778639 NaN NaN 1.814954787 NaN NaN 1.923858574 0
    ];%记录两个节点对应的行驶单位距离所需要的的时间
a=  [
    0   Inf Inf Inf Inf 20.61818756 Inf Inf Inf
    Inf 0   Inf Inf Inf 32.98731317 Inf Inf 30.45885936
    Inf Inf 0   10.26097159 Inf Inf Inf 24.88347373 Inf
    Inf Inf 23.08771312 0   22.85103298 Inf Inf Inf Inf
    Inf Inf Inf 16.46177306 0   Inf Inf Inf 36.10808574
    48.54142976 38.54744453 Inf Inf Inf 0   12.05861636 Inf Inf
    Inf Inf Inf Inf Inf 40.39096545 0   21.52433533 Inf
    Inf Inf 40.17532709 Inf Inf Inf 20.99831881 0   32.33392733
    Inf 18.79645823 Inf Inf 34.48414095 Inf Inf 34.62945434 0
    ];%记录两个节点之间的时间
Dj= [
    13  44  1
    57  44  2
    38  13  3
    38  7   4
    57  7   5
    38  44  6
    38  34  7
    38  27  8
    57  27  9
    ];%节点坐标


n=7;
zzzz=5*ones(1,7);%记录7个车辆的行驶状态
p=randperm(9);
sta=p(1:n);
en=p(10-n:9);%随机生成7辆汽车，起点和终点各不相同
Mj=Dj(sta,:);%Mj记录7个车的位置
Mi1=Mj(1,1:2);
Mi2=Mj(2,1:2);
Mi3=Mj(3,1:2);
Mi4=Mj(4,1:2);
Mi5=Mj(5,1:2);
Mi6=Mj(6,1:2);
Mi7=Mj(7,1:2);
Mj=[Mj(:,1:2),zeros(n,1),ones(n,2)];
paths1=Flyod(sta(1),en(1),a);%调用Flyod函数，求出每辆车的行驶路径
paths2=Flyod(sta(2),en(2),a);
paths3=Flyod(sta(3),en(3),a);
paths4=Flyod(sta(4),en(4),a);
paths5=Flyod(sta(5),en(5),a);
paths6=Flyod(sta(6),en(6),a);
```

```
paths7=Flyod(sta(7),en(7),a);
Mi1=change( Mi1,paths1,Dj );%调用change函数，进行即将行驶的路方向
Mi2=change( Mi2,paths2,Dj );
Mi3=change( Mi3,paths3,Dj );
Mi4=change( Mi4,paths4,Dj );
Mi5=change( Mi5,paths5,Dj );
Mi6=change( Mi6,paths6,Dj );
Mi7=change( Mi7,paths7,Dj );

while sum(Mj(:,5))>0
if Mj(1,5)~=0
    h=1;
    t=time(paths1,zzzz(h),tim);
    Mi1=move(Mi1,zzzz(h),t);
    Mj(1,1:4)=Mi1(1:4);
    f=Mj(1,1:2);
    b=Dj(:,1:2);
    if sum(ismember(b,f,'rows')==1)==1
        zzzz(h)=zzzz(h)+1;
        if zzzz(h)==(size(Mi1,2)+1)
            Mj(1,5)=0;
        end
    end

end
if Mj(2,5)~=0
    h=2;
    t=time(paths2,zzzz(h),tim);
    Mi2=move(Mi2,zzzz(h),t);
    Mj(2,1:4)=Mi2(1:4);
    f=Mi2(1:2);
    b=Dj(:,1:2);
    if sum(ismember(b,f,'rows')==1)==1
        zzzz(h)=zzzz(h)+1;
        if zzzz(h)==(size(Mi2,2)+1)
         Mj(h,5)=0;
        end
    end

end
if Mj(3,5)~=0
    h=3;
    t=time(paths3,zzzz(h),tim);
    Mi3=move(Mi3,zzzz(h),t);
    Mj(3,1:4)=Mi3(1:4);
    f=Mi3(1:2);
    b=Dj(:,1:2);
    if sum(ismember(b,f,'rows')==1)==1
        zzzz(h)=zzzz(h)+1;
        if zzzz(h)==(size(Mi3,2)+1)
        Mj(h,5)=0;
        end
    end
end
if Mj(4,5)~=0
    h=4;
    t=time(paths4,zzzz(h),tim);
    Mi4=move(Mi4,zzzz(h),t);
    Mj(4,1:4)=Mi4(1:4);
    f=Mi4(1:2);
    b=Dj(:,1:2);
    if sum(ismember(b,f,'rows')==1)==1
        zzzz(h)=zzzz(h)+1;
        if zzzz(h)==(size(Mi4,2)+1)
         Mj(h,5)=0;
        end
end
end
if Mj(5,5)~=0
    h=5;
    t=time(paths5,zzzz(h),tim);
    Mi5=move(Mi5,zzzz(h),t);
    Mj(5,1:4)=Mi5(1:4);
    b=Dj(:,1:2);
    if sum(ismember(b,f,'rows')==1)==1
        zzzz(h)=zzzz(h)+1;
        if zzzz(h)==(size(Mi5,2)+1)
        Mj(h,5)=0;
      end
    end
end
if Mj(6,5)~=0
    h=6;
    t=time(paths6,zzzz(h),tim);
    Mi6=move(Mi6,zzzz(h),t);
    Mj(6,1:4)=Mi6(1:4);
    f=Mi6(1:2)  ;
    f=Mj(6,1:2);
    b=Dj(:,1:2);
    if sum(ismember(b,f,'rows')==1)==1
        zzzz(h)=zzzz(h)+1;
        if zzzz(h)==(size(Mi6,2)+1)
            Mj(h,5)=0;
        end
    end
end
if Mj(7,5)~=0
    h=7;
    t=time(paths7,zzzz(h),tim);
    Mi7=move(Mi7,zzzz(h),t);
    Mj(7,1:4)=Mi7(1:4);
    f=Mi7(1:2)  ;
    b=Dj(:,1:2);
    if sum(ismember(b,f,'rows')==1)==1
        zzzz(h)=zzzz(h)+1;
        if zzzz(h)==(size(Mi7,2)+1)
        Mj(h,5)=0;
        end
    end

end


hold on
plot(Mi1(1,1),Mi1(1,2),'*',Mi2(1,1),Mi2(1,2),'<',Mi3(1,1),Mi3(1,2),'p',Mi4(1,1),Mi4(1,
2),'x',Mi5(1,1),Mi5(1,2),'>',Mi6(1,1),Mi6(1,2),'o',Mi7(1,1),Mi7(1,2),'+')
axis([0 70 0 70]);
end
```

move.m

```
function [ Mi] = move(Mi,z,t)%每驾驶1单位长度，就停顿t时刻，记录move程序所运行的时间，
可以判断总共汽车到达目的地所使用的时间
if Mi(1,z)==4%上走一步
    Mi(1,2)=Mi(1,2)+1;
    Mi(1,4)=1;
    Mi(1,3)=0;
  pause(t)
end
if Mi(1,z)==3%下走
    Mi(1,2)=Mi(1,2)-1;
    Mi(1,3)=1;
    Mi(1,4)=0;
  pause(t)
end
if Mi(1,z)==2%右走
    Mi(1,1)=Mi(1,1)+1;
    Mi(1,4)=1;
    Mi(1,3)=0;
  pause(t)
end
if Mi(1,z)==1%左走
    Mi(1,1)=Mi(1,1)-1;
    Mi(1,3)=1;
    Mi(1,4)=0;
  pause(t)
end
end
```

time.m

```
function [ t] = time( paths1,zzzz,tim)%获得当前时刻，所行驶单位距离所需要的的时间
    time1=paths1(1,size(paths1,2)-zzzz+5);
    time2=paths1(1,size(paths1,2)-zzzz+4);
    t=tim(time1,time2)/100;
end
```