*May 4th, 2018*

**Exercise**

**Problem 2**

The problem 2 is described as the following figure:

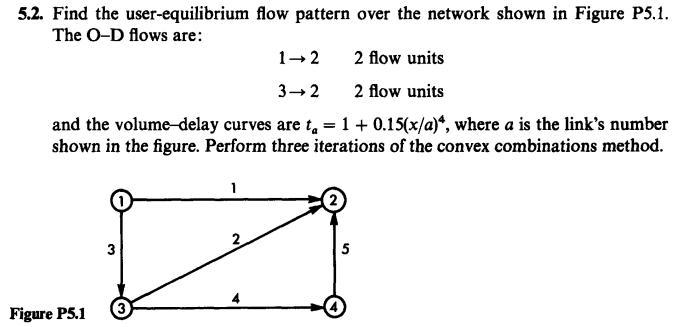
****

Figure 1 the description of problem 2

The problem is solved using convex combinations method with MATLAB. After three iterations, the flow and travel time on each link is shown as the following table:

Table 1 The flow and travel time on each link

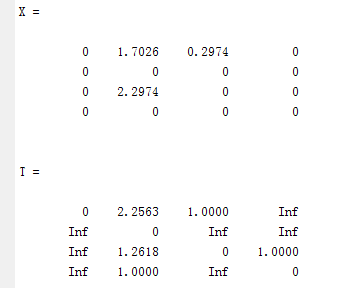
|  |  |  |
| --- | --- | --- |
| Link | Flow | Travel time |
| 1 | 1.7026 | 2.2563 |
| 2 | 2.2974 | 1.2618 |
| 3 | 0.2974 | 1 |
| 4 | 0 | 1 |
| 5 | 0 | 1 |

The flows and travel time on each path of each O-D pair is shown as follows:

Table 2 The flows and travel time on each path of each O-D pair

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| O-D (1-2) | Flow | Travel time | O-D (3-2) | Flow | Travel time |
| Path 1(1-2) | 1.7026 | 2.2563 | Path 1(3-2) | 2 | 1.2618 |
| Path 2(1-3-2) | 0.2974 | 2.2618 | Path 2(3-4-2) | 0 | 2 |
| Path 3(1-3-4-2) | 0 | 0 | The flow pattern is very close to equilibrium flow | | |

It can be concluded from Table 2 that the flow pattern in the network is very close to the user equilibrium flow pattern.

The MATAB code is shown in the end. For this code, the shortcoming is that it can't automatically find all the feasible path for each O-D pair and show its travel time and flow. The result it can be reached is the flow and travel time on each link. The result of this code in the command window of MATLAB software is shown as (X represents the flow pattern and T represents the travel time on each link):  


**MATLAB CODE**

(1)The function code for the objective function is shown as follows:  
function fun=originalfun(B)

syms x

fun=0;

T=[0 1+0.15\*(x/1)^4 1+0.15\*(x/3)^4 inf;

inf 0 inf inf;

inf 1+0.15\*(x/2)^4 0 1+0.15\*(x/4)^4;

inf 1+0.15\*(x/5)^4 inf 0];

for i=1:size(B,1)

for j=1:size(B,1)

s=T(i,j);

a=0;

b=B(i,j);

fun=int(s,x,a,b)+fun;

end

end

(2) The function code for label correcting method is shown as follows:  
function predecessor=labelcorrecting(A,X,T,r)

%利用label-correcting method 求解最短路

%即找出根点到任意点之间的最短路

%label list represent the distance from origin node to each node

label=1./zeros(1,4);

%predecessor list represent the predecessor node of each node that

%have yet to be examined as well as the nodes requiring further examination

predecessor=zeros(1,4);

%sequence list represent the nodes

sequence=[];

sequence=[r,sequence];

n=size(sequence,1);

label(r)=0;

while n~=0

r=sequence(1);

for j=1:size(A,1)

if A(r,j)~=0

temporary\_label=label(r)+T(r,j);

if temporary\_label<label(j)

label(j)=temporary\_label;

predecessor(j)=r;

sequence=[sequence,j];

end

end

end

sequence(sequence==r)=[];

n=size(sequence,2);

end

(3) The function code for the optima step size is shown as follows:

function a=stepsize(X,Y)

syms a;

B=X+a\*(Y-X);

phi=originalfun(B);

a=0;

b=1;

epsilon=0.001;

a=bisection(phi,a,b,epsilon);

(4) The function code for the bisection method is shown as follows:

function [s,phis,k,G,ds]=bisection(phi,a,b,epsilon)

h=b-a;

k=0;

d\_phi=diff(phi);

d=inline(d\_phi);

phi=inline(phi);

while h>(2\*epsilon)

s=(a+b)/2;

d\_value=feval(d,s);

k=k+1;

G(k,:)=[a,s,b];

if d\_value<=0

a=s;

else

b=s;

end

h=b-a;

end

s=(a+b)/2;

phis=feval(phi,s);

ds=(b-a)/2;

(5) The code for the bisection method is shown as follows:

clc;

clear;

%图的输入

A=[0 1 1 0;

0 0 0 0;

0 1 0 1;

0 1 0 0];

%the flow on each link

X=[0 0 0 0;

0 0 0 0;

0 0 0 0;

0 0 0 0];

%travel time on each link

T=[0 1+0.15\*(X(1,2)/1)^4 1+0.15\*(X(1,3)/3)^4 inf;

inf 0 inf inf;

inf 1+0.15\*(X(3,2)/2)^4 0 1+0.15\*(X(3,4)/4)^4;

inf 1+0.15\*(X(4,2)/5)^4 inf 0];

%flow on each O-D

f=[0 2 0 0;

0 0 0 0;

0 2 0 0;

0 0 0 0];

%存放最短路径

%第一次全有全无分配

for i=1:size(A,1)

s=[];

for j=1:size(f,1)

if f(i,j)~=0

%找出ij之间的最短路，最短路的中的每个路段都是f

q=f(i,j);

r=i;

predecessor=labelcorrecting(A,X,T,r);

k=j;

s=[j,s];

while predecessor(k)~=0

s=[predecessor(k),s];

k=predecessor(k);

end

end

end

%根据最短路径变更X的值,寻找Y

for i=1:(size(s,2)-1)

X(s(i),s(i+1))=q+X(s(i),s(i+1));

end

end

n=1;

while n<3

%update the travel time on each link

T=[0 1+0.15\*(X(1,2)/1)^4 1+0.15\*(X(1,3)/3)^4 inf;

inf 0 inf inf;

inf 1+0.15\*(X(3,2)/2)^4 0 1+0.15\*(X(3,4)/4)^4;

inf 1+0.15\*(X(4,2)/5)^4 inf 0];

%direction finding

Y=zeros(4,4);

for i=1:size(A,1)

s=[];

for j=1:size(f,1)

if f(i,j)~=0

%找出ij之间的最短路，最短路的中的每个路段都是f

q=f(i,j);

r=i;

predecessor=labelcorrecting(A,X,T,r);

k=j;

s=[j,s];

while predecessor(k)~=0

s=[predecessor(k),s];

k=predecessor(k);

end

end

end

%根据最短路径变更X的值,寻找Y

for i=1:(size(s,2)-1)

Y(s(i),s(i+1))=q+Y(s(i),s(i+1));

end

end

%find step size using bisection method

a=stepsize(X,Y);

%update x

for i=1:size(X,1)

for j=1:size(X,1)

X(i,j)=X(i,j)+a\*(Y(i,j)-X(i,j));

end

end

n=n+1;

end

%结果输出

X

T