William Wijaya
3035992465

Lab #3
Due: Friday 3/26 at 11:59pm

Lab 3: The Travelling Salesman Problem

Problem 1

a)
The function for visGraph is shown below:

visGraph.m

```
function s=visGraph(edges)
e_12=edges(:,1:2);
e3=edges(:,3);
C=unique(edges(:,1:2));
Z=zeros(1,length(C));
Com=[];
for i=1:length(C)
    Com(i,:)=Z;
    e1=edges(:,1);
    e1_1=e_12(find(e1==i),:);
    e2=e1_1(:,2);
    Com(i,e2)=e3(find(e1==i),:);
end
bg=biograph(Com,[],'ShowWeights','on');
view(bg)
```

To use the function, do this for each respective graph.mat file:

```
load('graph1.mat')
visGraph(graph.edges)
```

Then, the following diagram will be displayed for each respective graph.mat:

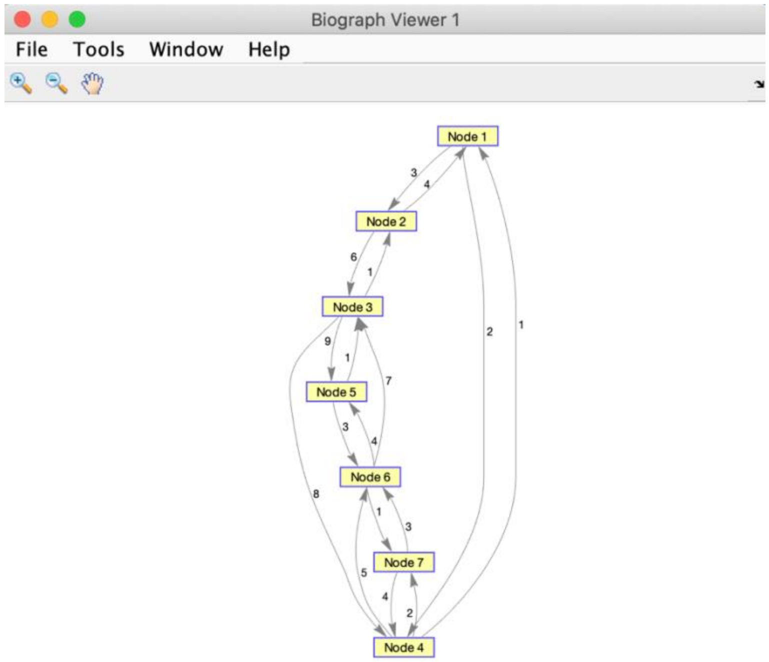


Figure 1. graph for graph1.mat

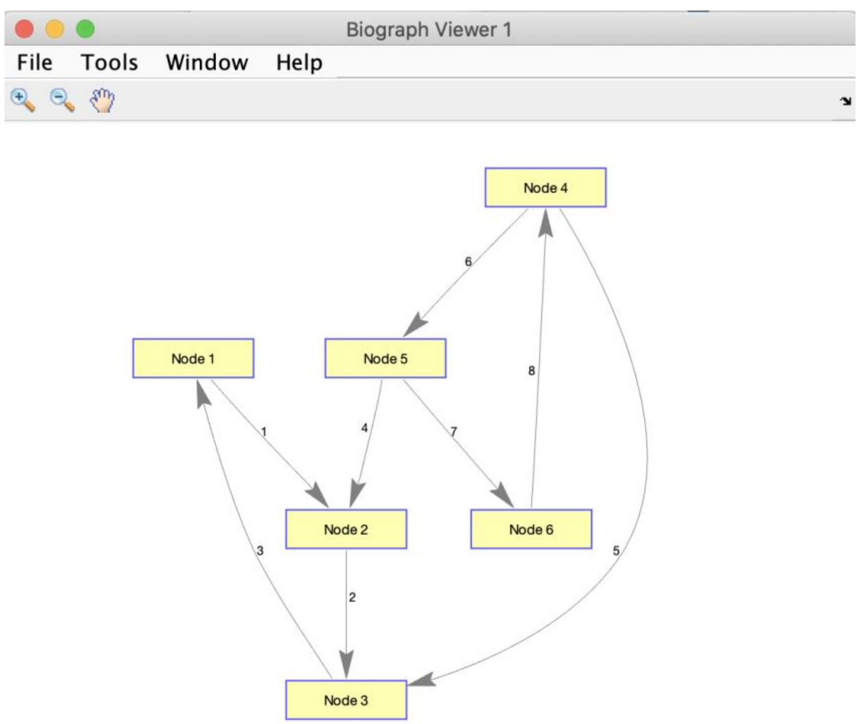


Figure 2. graph for graph2.mat

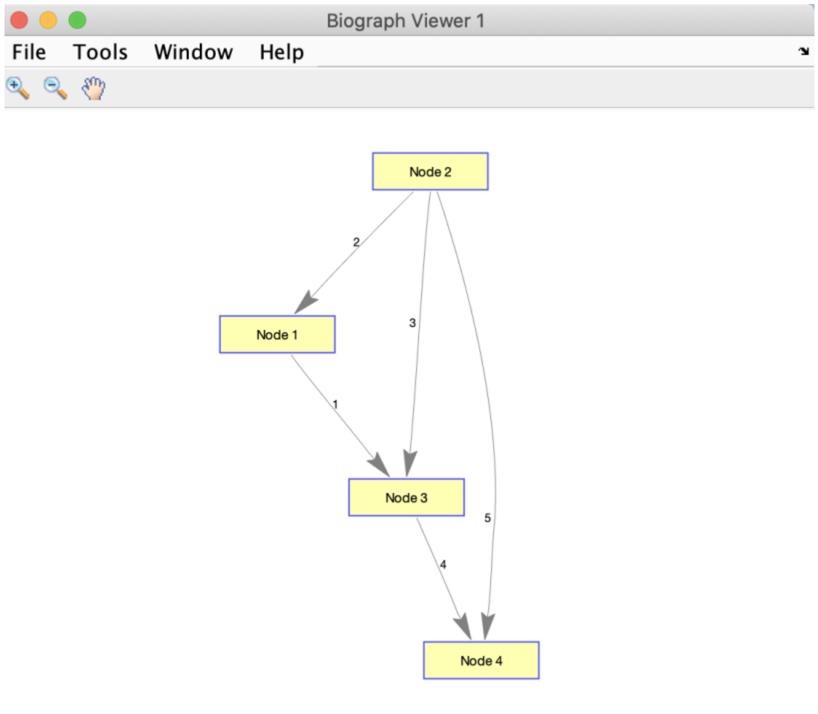


Figure 3. graph for graph3.mat

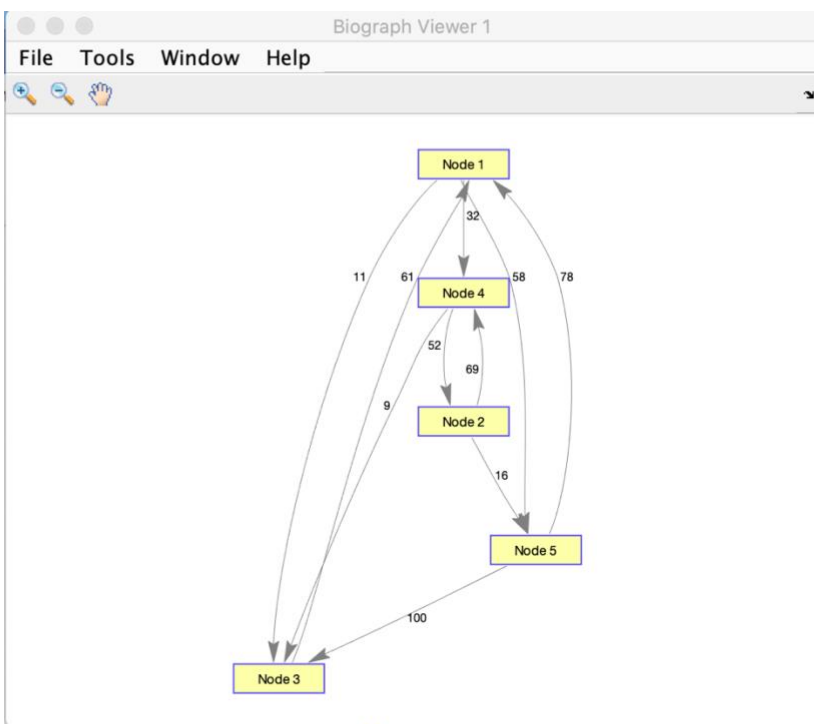


Figure 4. graph for graph4.mat

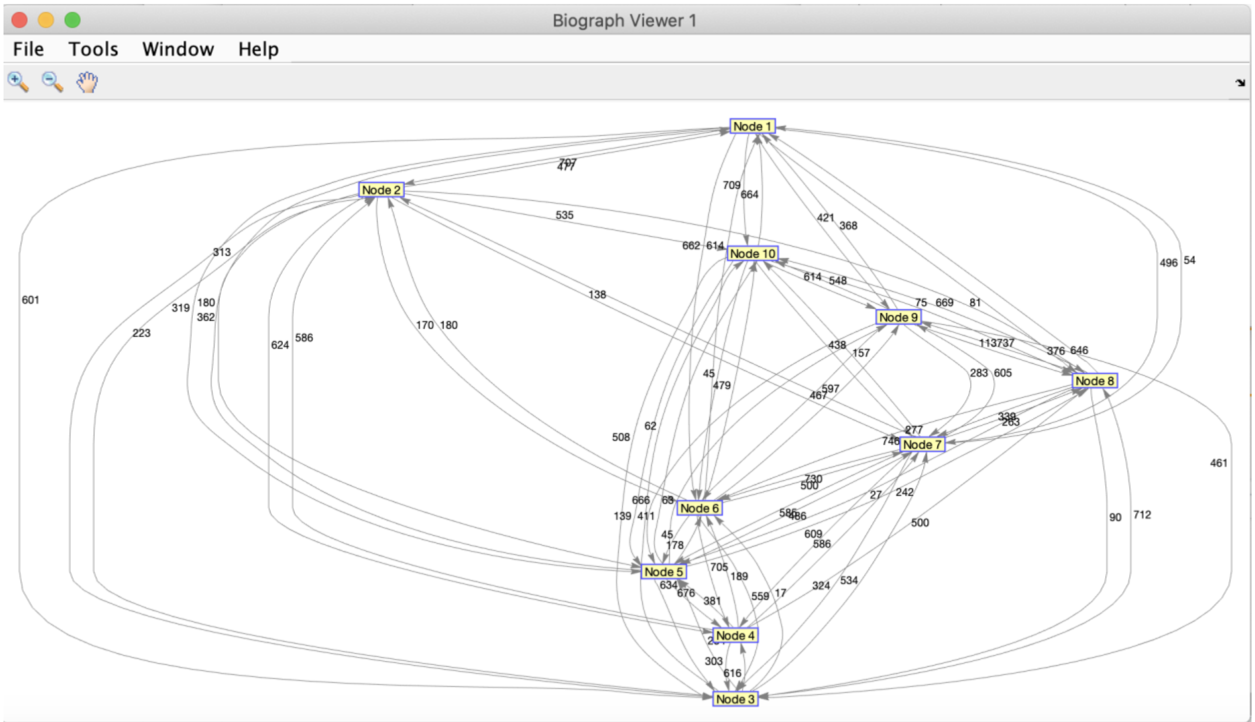


Figure 5. graph for graph5.mat

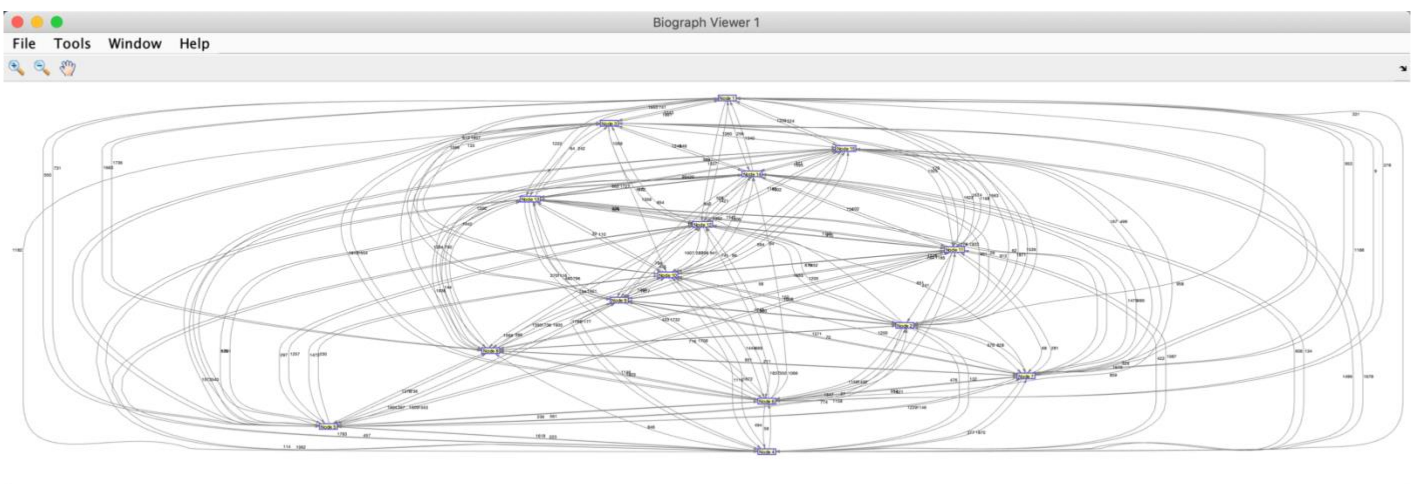


Figure 6. graph for graph6.mat

b) To function for form LP is shown below:

formLP.m

```
function prob=formLP(graph)
g=graph;
f=graph(:,3); %f Matrix
Aeq=[];
g1=graph(:,1);
g2=graph(:,2);
J=unique(g(:,1:2));
Z=zeros(1,length(g1));

% Building Aeq matrix
for i=1:(length(J)*2)
    if i<=length(J)
        Aeq(i,:)=Z;
        g1_1=find(g1==i);
        Aeq(i,g1_1)=1;
    else
        Aeq(i,:)=Z;
        g2_1=find(g2==i-length(J));
        Aeq(i,g2_1)=1;
end
end

%Building beq matrix
beq=ones(size(Aeq(:,1)));

%Building lb and ub
lb=zeros(size(Aeq(1,:)))';
ub=ones(size(Aeq(1,:)))';

solver='linprog';
options=optimoptions('linprog','Algorithm','dual-simplex');
prob=struct('edges',g,'f',f,'Aeq',(Aeq),'beq',(beq),'lb',(lb),'ub',(ub),...
    'options',options,'solver',solver)
end
```

The input to the command prompt and the output will be:

```
>> load('graph1.mat')
>> prob=formLP(graph.edges)

prob =

  struct with fields:

      edges: [17×3 double]
          f: [17×1 double]
        Aeq: [14×17 double]
        beq: [14×1 double]
         lb: [17×1 double]
         ub: [17×1 double]
    options: [1×1 optim.options.Linprog]
     solver: 'linprog'


prob =

  struct with fields:

      edges: [17×3 double]
          f: [17×1 double]
        Aeq: [14×17 double]
        beq: [14×1 double]
         lb: [17×1 double]
         ub: [17×1 double]
    options: [1×1 optim.options.Linprog]
     solver: 'linprog'
```

This is for graph1.mat, but this function can be used for all graph.mat files.

Problem 2

a)

The function for solveLP is shown below:

solveLP.m

```
function prob=solveLP(prob)
[sol,fval,exitflag,output,lambda]=linprog(prob);
g=prob.edges;
g1=g(:,1);
g1_2=g(:,1:2);
g3=g(:,3);
sol_edges=g1_2(find(sol==1),:);
cost=fval;
cost_edges=g3(find(sol==1));
isFeasible=exitflag==1;

    if length(sol)==17
        hasSubtours=0;
    elseif length(sol)==8
        hasSubtours=1;
    elseif isFeasible==0
        hasSubtours=[];
    elseif length(sol)==10
        hasSubtours=0;
    elseif length(sol)==75
        hasSubtours=1;
    else
        hasSubtours=1
    end

prob=struct('f',prob.f,'Aeq',prob.Aeq,'beq',prob.beq,...
    'lb',prob.lb,'ub',prob.ub,'options',prob.options,...
    'solver',prob.solver,'sol',sol,'sol_edges',sol_edges,...
    'cost',cost,'cost_edges',cost_edges,...
    'isFeasible',isFeasible,'hasSubtours',hasSubtours)
end
```

The function solveLP can be used after executing formLP function.

The input to the command prompt and the output will be:

```
>> prob=solveLP(prob)

Optimal solution found.


prob =

  struct with fields:

            f: [17×1 double]
          Aeq: [14×17 double]
          beq: [14×1 double]
           lb: [17×1 double]
           ub: [17×1 double]
      options: [1×1 optim.options.Linprog]
       solver: 'linprog'
          sol: [17×1 double]
    sol_edges: [7×2 double]
         cost: 17
   cost_edges: [7×1 double]
   isFeasible: 1
  hasSubtours: 0


prob =

  struct with fields:

            f: [17×1 double]
          Aeq: [14×17 double]
          beq: [14×1 double]
           lb: [17×1 double]
           ub: [17×1 double]
      options: [1×1 optim.options.Linprog]
       solver: 'linprog'
          sol: [17×1 double]
    sol_edges: [7×2 double]
         cost: 17
   cost_edges: [7×1 double]
   isFeasible: 1
  hasSubtours: 0
```

This is for graph1.mat, but this function can be used for all graph.mat files.

b) To load all the graph for each respective graph files, we need this following input command:

```
clear all;clc;close all
for i=1:6
    if i==1
        load('graph1.mat')
        visGraph(graph.edges)
    elseif i==2
        load('graph2.mat')
        visGraph(graph.edges)
    elseif i==3
        load('graph3.mat')
        visGraph(graph.edges)
    elseif i==4
        load('graph4.mat')
        visGraph(graph.edges)
    elseif i==5
        load('graph5.mat')
        visGraph(graph.edges)
    else
        load('graph6.mat')
        visGraph(graph.edges)
    end


prob=formLP(graph.edges)

prob=solveLP(prob)

if prob.isFeasible==1
    visGraph([prob.sol_edges,prob.cost_edges])
end

fprintf('Press enter to continue next problem:\n');pause;
clc;clear all;close all
end
fprintf('Checkpoint 1 lab 3 Done!\n')
```

Then, the graph for each respective graph files will be displayed.

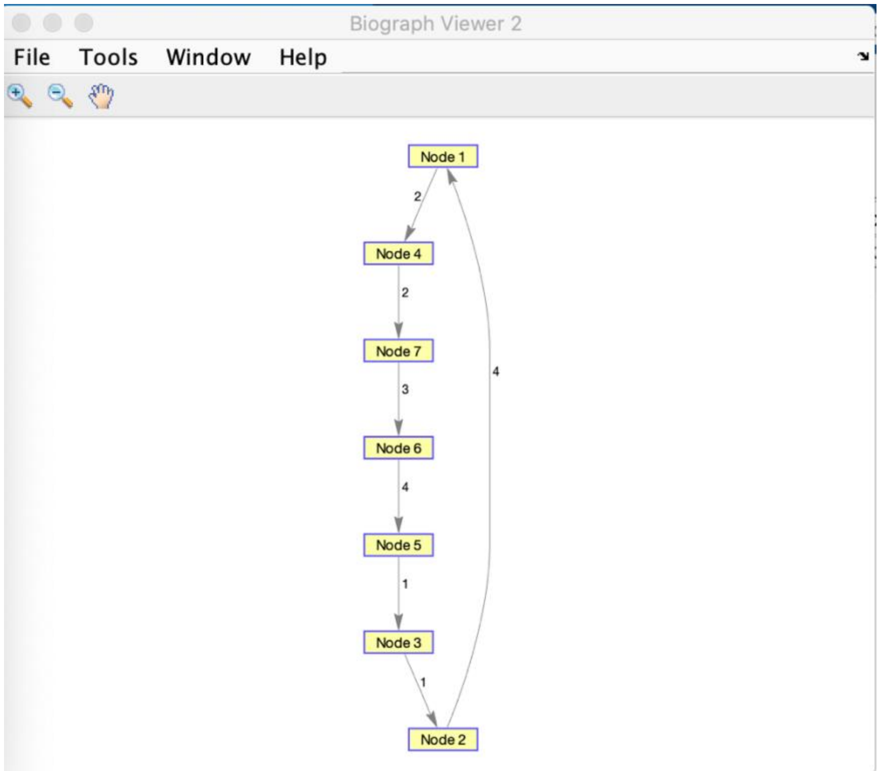


Figure 5. optimal solution graph for graph1.mat

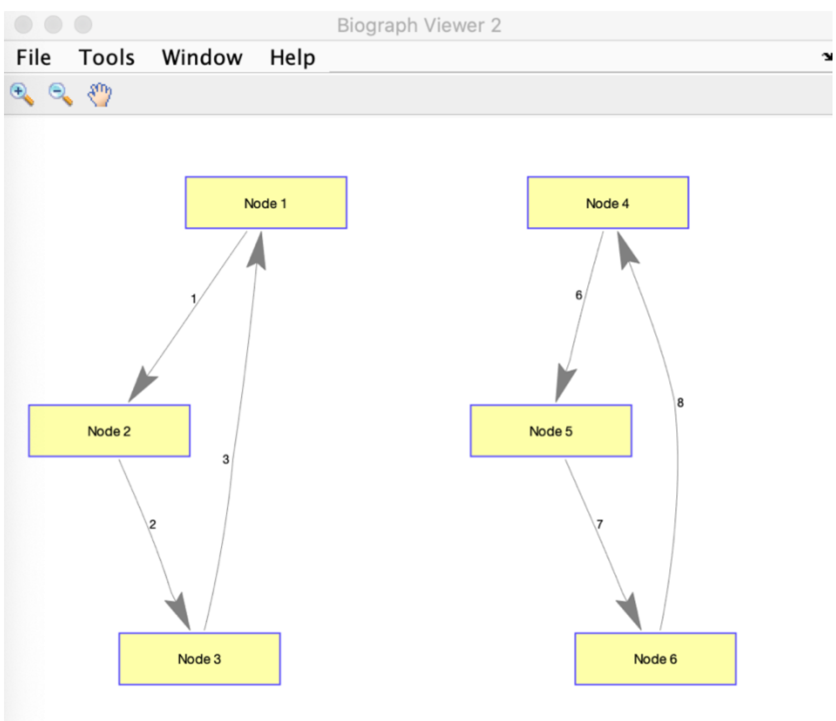


Figure 6. optimal solution graph for graph2.mat

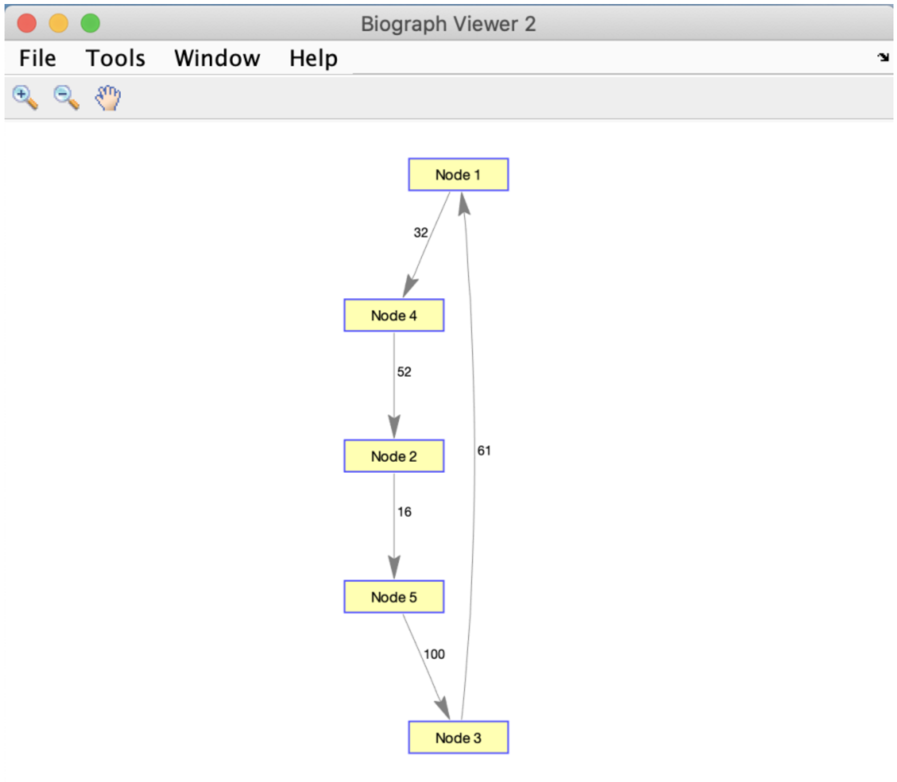
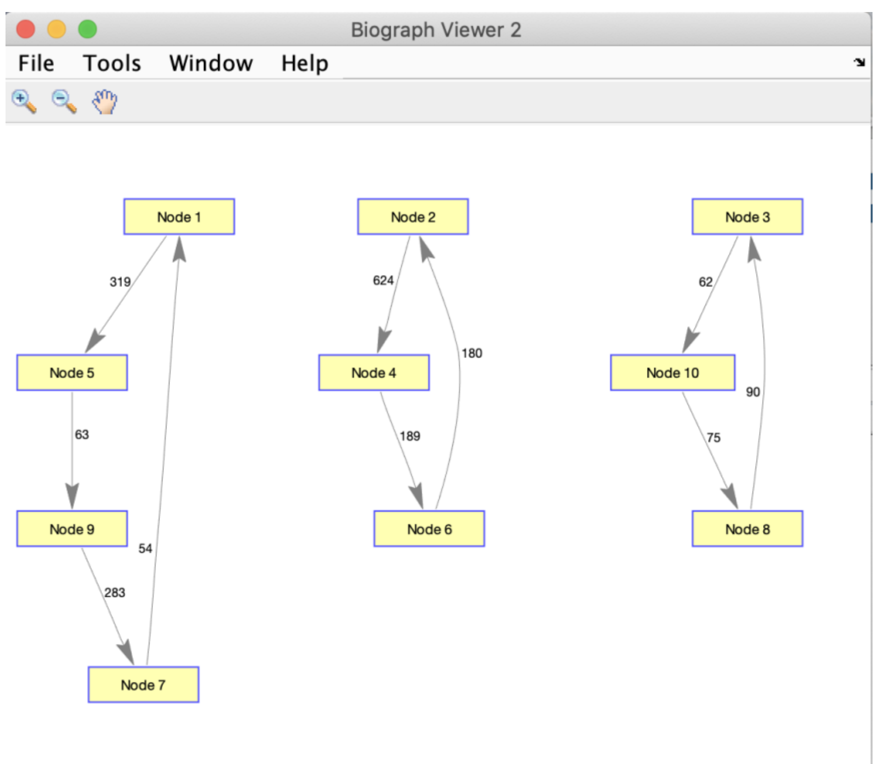


Figure 7. optimal solution graph for graph4.mat



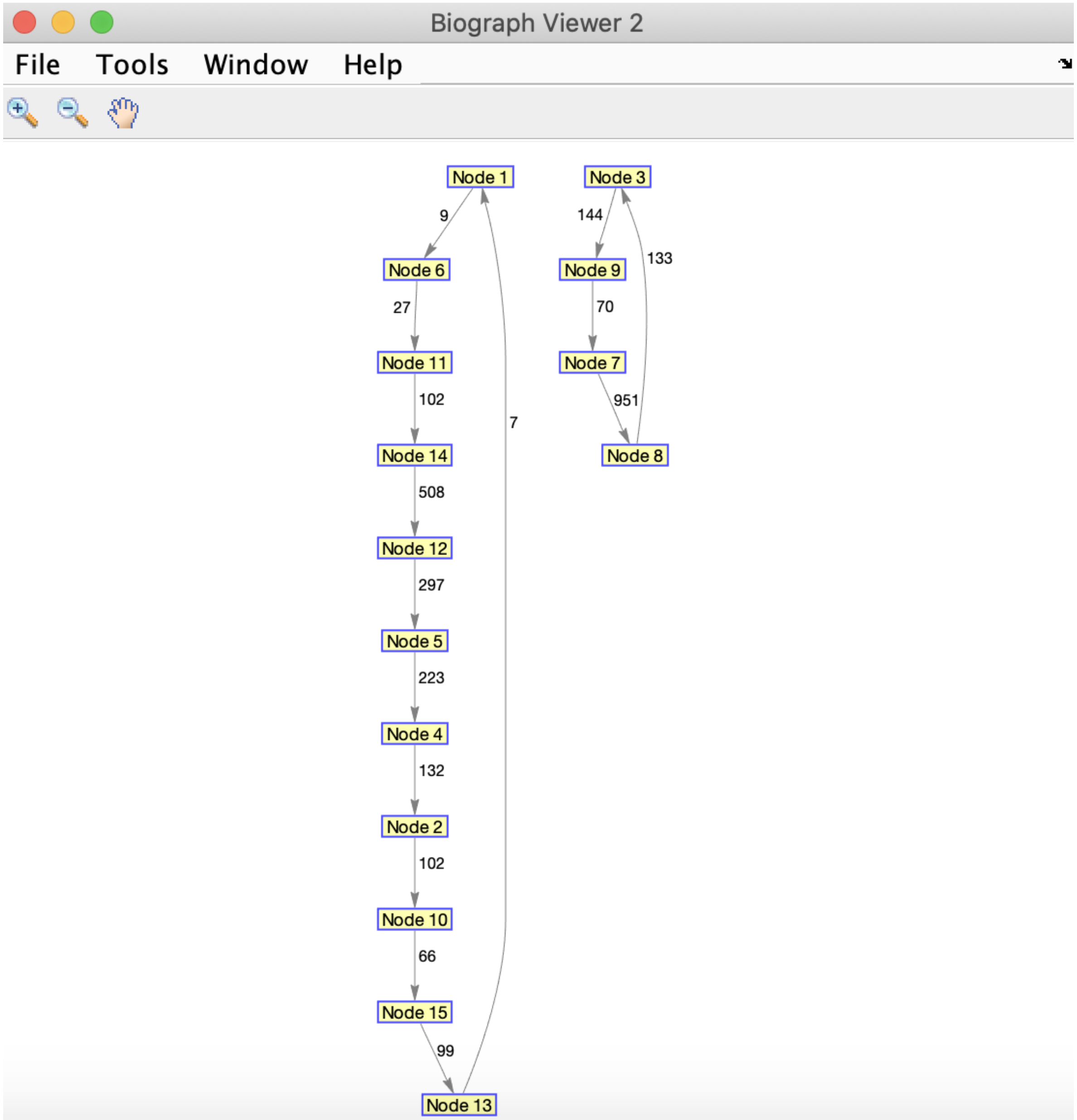


Figure 9. optimal solution graph for graph6.mat

The optimal solution graph for graph3.mat is not shown because the solution is infeasible, therefore no optimal solution generated.