4.3

这个就只是画出赋权图，我们直接调用prim算法

**代码如下：**

clc,clear

a = zeros(6);

a(1,[2,5]) = [20,15];a(2,[3,4,5]) = [20,60,25];

a(3,[4,5]) = [30,18];a(4,[5,6]) = [35,10];

a(5,6) = 15; a = a+a'; %建立邻接矩阵

s = cellstr(strcat('v',int2str([1:6]')));

G = graph(a,s,'upper');%画出无向赋权图

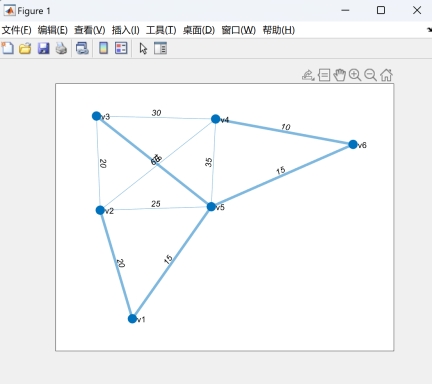
p = plot(G,'EdgeLabel',G.Edges.Weight)

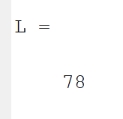
T = minspantree(G)%画出最小生成树

L = sum(T.Edges.Weight)%找出最小生成树的路径，并计算总和

highlight(p,T)%着重标记最小生成树

**结果如下：**





我们可知**最小生成树的长度为78**

4.8

由于我们无法做到如题所示的0.6概率的随机，我们可以调用随机函数矩阵来表示随机的概念，

**代码如下：**

clc,clear

a = rand(10);%构造概率矩阵

a = triu(a,1);%我们取上三角元素

w = randi(10,10);%构造了权重矩阵

W = (a>=0.4).\*w%生成无向赋权图邻接矩阵的上三角部分

W = W +W'%生成完全邻接矩阵

G = graph(W,'upper')

subplot(121),plot(G,'EdgeLabel',G.Edges.Weight)

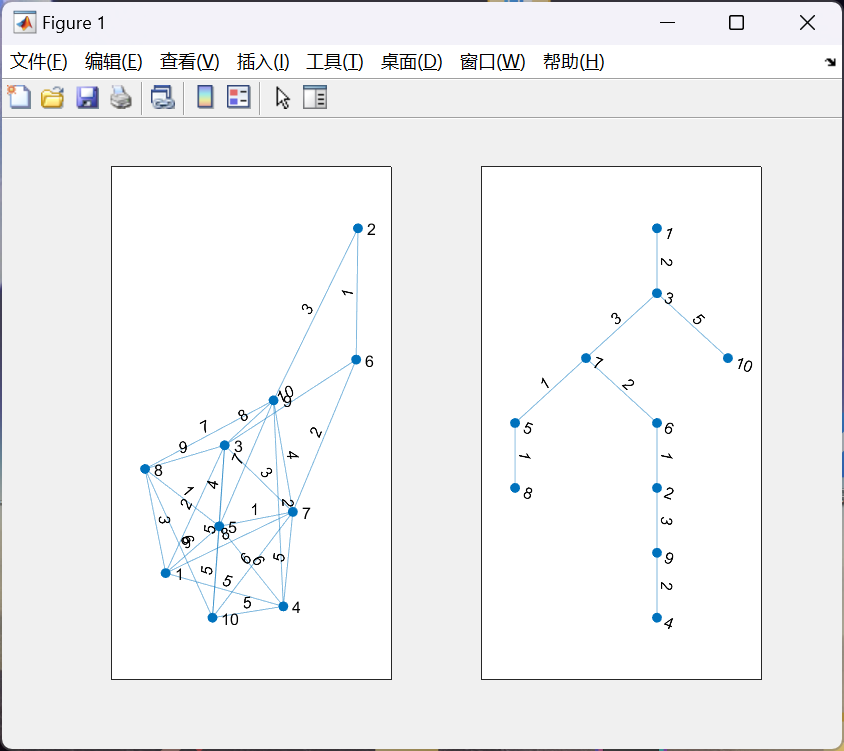
T = minspantree(G)%使用Prim算法求得最小生成树

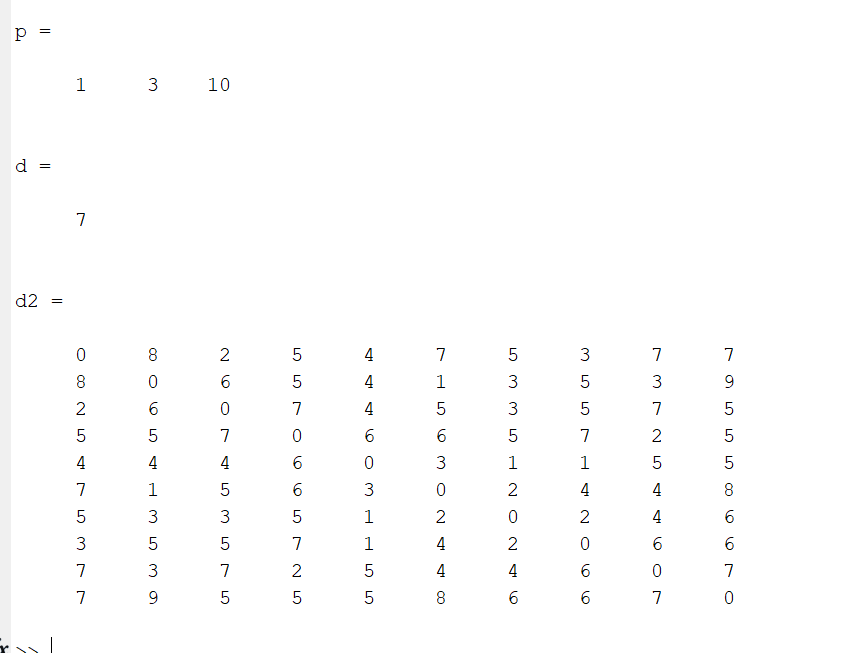
subplot(122),plot(T,'EdgeLabel',T.Edges.Weight)

[p,d] = shortestpath(G,1,10)%q求得1-10的最短距离及最短路径；

d2 = distances(G)

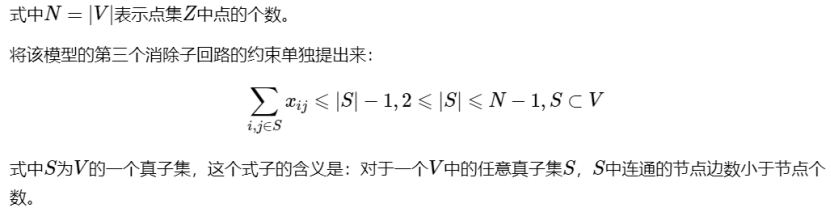
**结果如下：**





1. 最小生成树如上图所示
2. 路径为1→3→10，最短路径长度为7
3. 每个点的最短距离如上

4.13



该问题可以转化为0—1整数规划类问题，具体问题可以转化为如下



**代码如下：**

clc, clear, close all, n = 9;

nod =cellstr(strcat('v',int2str([0:n-1]')));%运用cellstr进行标号

G = graph(); G = addnode(G,nod); %定好无序图

ed ={ 'v0','v1',2;'v0','v2',1;'v0','v3',3;'v0','v4',4

'v0','v5',4;'v0','v6',2;'v0','v7',5;'v0','v8',4

'v1','v2',4;'v1','v8',1;'v2','v3',1;'v3','v4',1

'v4','v5',5;'v5','v6',2;'v6','v7',3;'v7','v8',5};

G = addedge(G,ed(:,1),ed(:,2),cell2mat(ed(:,3)));%无序图确认

p = plot(G,'EdgeLabel',G.Edges.Weight) %做出9个村庄道路及道路长度图

w = full(adjacency(G,'weighted')); %做出边权矩阵

w(w==0) = 1000000; %充分大的正实数，让所有不能直接到达的两个村庄改为足够大

prob = optimproblem;

x = optimvar('x',n,n,'Type','integer','LowerBound',0,'UpperBound',1);

prob.Objective = sum(sum(w.\*x));

prob.Constraints.con1 = 1<=sum(x(1,:));%条件1

prob.Constraints.con2 = sum(x(:,2:end))'==1; %条件2

con3 = [];

for q = 2:n-1

a = zeros(q);

for m = 1:100 %100次足够精度

b = randperm(n);%随机对n数进行排序

c = b(1:q); %相当于从n中随机抽取p个数

a = x(c,c);

con3 = [sum(sum(a)) <= q-1;con3];

end

end

prob.Constraints.con3 = con3;%条件3

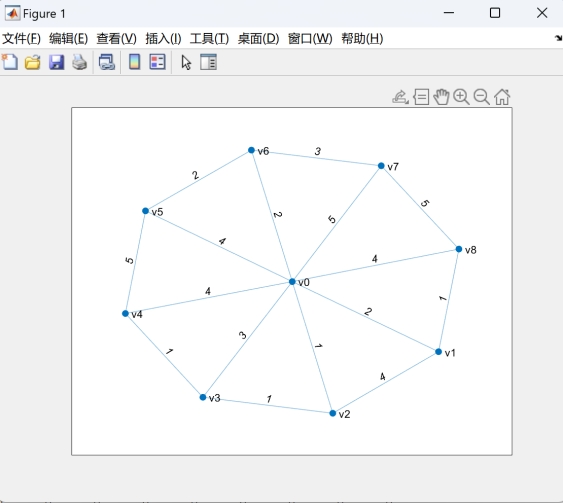
[sol,fval,flag,out] = solve(prob)

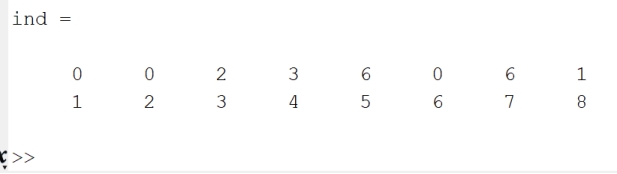
xx = sol.x

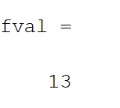
[i,j]=find(sol.x);

ind = [(i-1)'; (j-1)'] %输出树的顶点编号

**结果如下：**







**即最小生成树的坐标和权重都可得到**