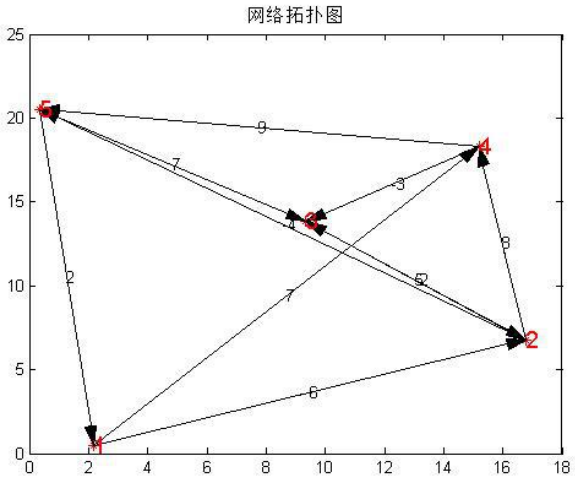
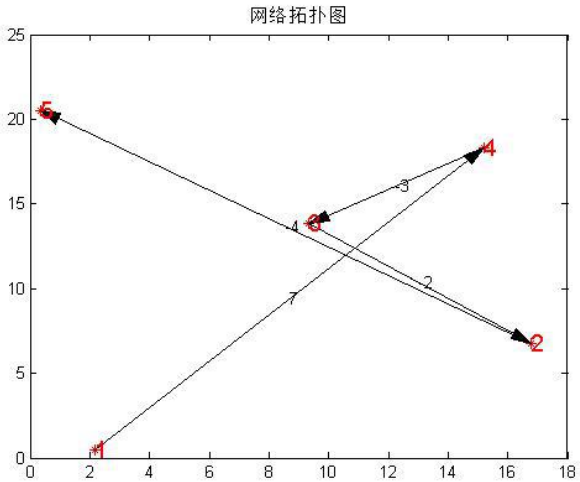
Bellman 最短路径编程实验

该算法可以用来解决一般（边的权值为负）的单源最短路径问题，而 dijkstra 只能解决权值非负的情况。

此算法使用松弛技术，对每一个顶点，逐步减少源到该顶点的路径的估计值，直到达到最短的路径。

算法运算结果：





matlab 代码如下，netplot 函数在这里，不过当时函数中表示两节点没有路径用的是 0，而现在需要改成 inf：

```
clear all;close all;clc
%初始化邻接压缩表
b=[1 2 6;
    1 4 7
    2 3 5;
    2 4 8;
    2 5 -4;
    3 2 -2;
    4 3 -3;
    4 5 9;
    5 1 2;
    5 3 7];

m=max(max(b(:,1:2)));                %压缩表中最大值就是邻接矩阵的宽与高
A=compresstable2matrix(b);   %从邻接压缩表构造图的矩阵表示
netplot(A,1)                         %形象表示

S=inf(1,m);                          %源到其他节点的最短距离，开始为 inf
S(1)=0;                              %源点到自己的距离为 0
pa=zeros(1,m);                       %寻找到的节点的前趋
pa(1)=1;                             %源点的前趋是自己

pre_pa=ones(1,m);while sum(pre_pa==pa)~=m      %终止条件，判断终止的
方法很多，这个应该不是最佳实践
    pre_pa=pa;
    for k=1:m
        if pre_pa(k)~=0                      %对每一个已搜寻到的节点，从此
节点寻找后继节点
            i=k;
            for j=1:m
                if A(i,j)~=inf
```

```
                    if S(j)>S(i)+A(i,j)
                        S(j)=S(i)+A(i,j);          %边缘松弛，取两节点间最
小权值作为实际权值
                        pa(j)=i;                   %寻找前趋
                    end
                end
            end
        end
    endend
%最终我们需要的就是这两个值
S       %源点到其他每一点的距离
pa      %其他每一节点的前趋

%算法到此结束，下面只是为了形象的表示而写的。
re=[];for i=2:m
    re=[re;pa(i) i A(pa(i),i)];end
A=compresstable2matrix(re);  %从邻接压缩表构造图的矩阵表示
figure;
netplot(A,1)                 %形象表示
```

compresstable2matrix.m

```
function A=compresstable2matrix(b)
    [n ~]=size(b);
    m=max(max(b(:,1:2)));
    A=inf(m,m);

    for i=1:n
        A(b(i,1),b(i,2))=b(i,3);
    end
end
```