创建图：

```
#encoding:utf-8
import networkx as nx
import matplotlib.pyplot as ply

import time

if __name__ == "__main__":
    digraph = nx.DiGraph()
    digraph.add_weighted_edges_from(ebunch = [(1,2,2),
                                              (1,4,4),
                                              (2,3,7),
                                              (4,5,3),
                                              (3,9,2),
                                              (5,6,3),
                                              (5,7,2),
                                              (6,9,2),
                                              (6,8,4),
                                              (7,8,6),
                                              (9,8,1)], weight='weight')

    for i in digraph.nodes():
        for j in digraph[i].items():
            digraph[i][j[0]]['capability'] = 1
```

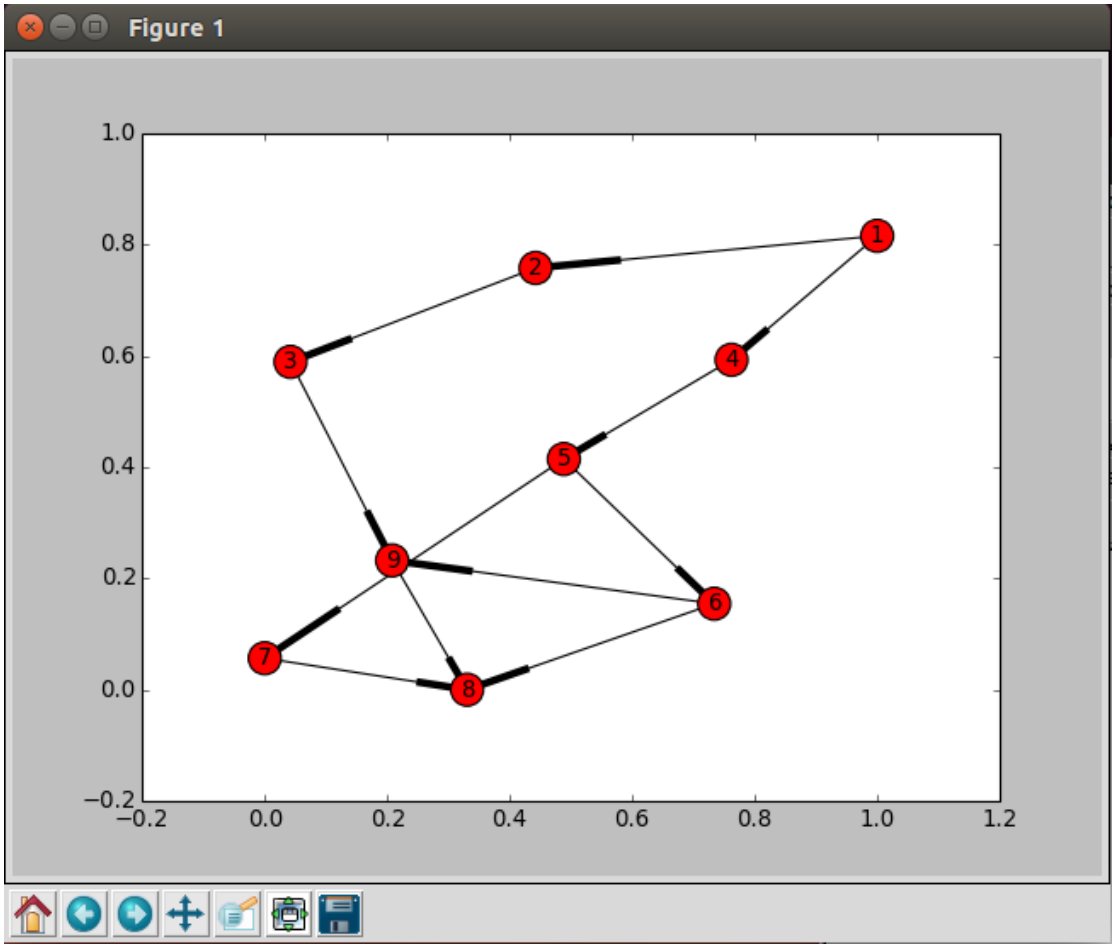
输出图：

```
    for i in digraph.nodes():
        for j in digraph[i].items():
            print i,'-->',j[0],digraph[i][j[0]]



    nx.drawing.draw_networkx(digraph)

    ply.show()
```

结果：

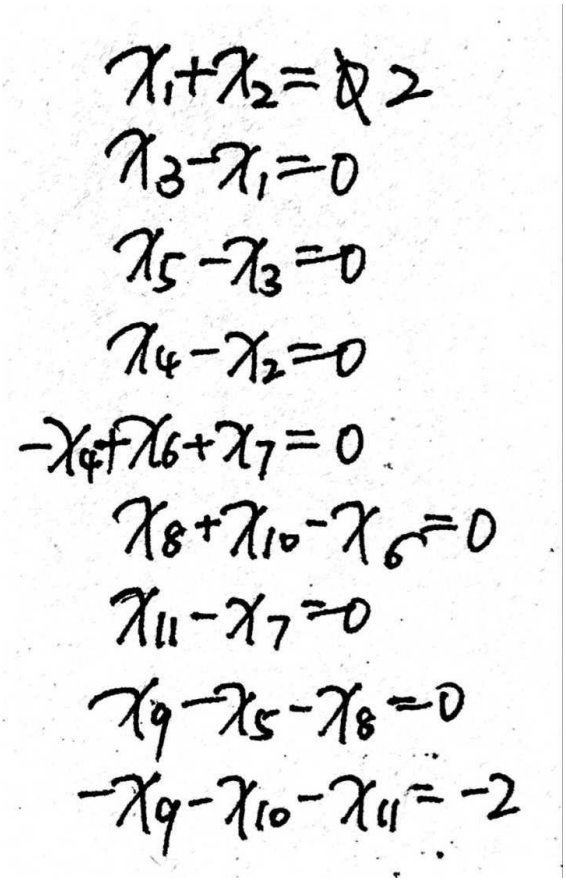
```
1 --> 2 {'capability': 1, 'weight': 2}
1 --> 4 {'capability': 1, 'weight': 4}
2 --> 3 {'capability': 1, 'weight': 7}
3 --> 9 {'capability': 1, 'weight': 2}
4 --> 5 {'capability': 1, 'weight': 3}
5 --> 6 {'capability': 1, 'weight': 3}
5 --> 7 {'capability': 1, 'weight': 2}
6 --> 8 {'capability': 1, 'weight': 4}
6 --> 9 {'capability': 1, 'weight': 2}
7 --> 8 {'capability': 1, 'weight': 6}
9 --> 8 {'capability': 1, 'weight': 1}
```

图的表示：



SDPP：

约束：



SDPP1：

增加其他约束

0<=Xi<=1

则输入 lingo：

```
model:
min=2*x1+4*x2+7*x3+3*x4+2*x5+3*x6+2*x7+2*x8+x9+4*x10+6*x11;
x1+x2=2;
x3-x1=0;
x5-x3=0;
x4-x2=0;
x6+x7-x4=0;
x8+x10-x6=0;
x11-x7=0;
x9-x5-x8=0;
-x9-x10-x11=-2;
0<=x1;
x1<=1;
0<=x2;
x2<=1;
0<=x3;
x3<=1;
0<=x4;
x4<=1;
0<=x5;
x5<=1;
0<=x6;
x6<=1;
0<=x7;
x7<=1;
0<=x8;
x8<=1;
0<=x9;
x9<=1;
0<=x10;
x10<=1;
0<=x11;
x11<=1;
end
```

输出结果：

```
 Global optimal solution found.
 Objective value:                              26.00000
 Infeasibilities:                              0.000000
 Total solver iterations:                             0


                     Variable           Value        Reduced Cost
                           X1        1.000000           0.000000
                           X2        1.000000           0.000000
                           X3        1.000000           0.000000
                           X4        1.000000           0.000000
                           X5        1.000000           0.000000
                           X6        1.000000           0.000000
                           X7        0.000000           0.000000
                           X8        0.000000           0.000000
                           X9        1.000000           0.000000
```

X10	1.000000	0.000000
X11	0.000000	1.000000

Row	Slack or Surplus	Dual Price
1	26.00000	-1.000000
2	0.000000	-14.00000
3	0.000000	-11.00000
4	0.000000	-4.000000
5	0.000000	-10.00000
6	0.000000	-7.000000
7	0.000000	-4.000000
8	0.000000	-5.000000
9	0.000000	-2.000000
10	0.000000	0.000000
11	1.000000	0.000000
12	0.000000	1.000000
13	1.000000	0.000000
14	0.000000	0.000000
15	1.000000	0.000000
16	0.000000	0.000000
17	1.000000	0.000000
18	0.000000	0.000000
19	1.000000	0.000000
20	0.000000	0.000000
21	1.000000	0.000000
22	0.000000	0.000000
23	0.000000	0.000000
24	1.000000	0.000000
25	0.000000	0.000000
26	1.000000	0.000000
27	1.000000	0.000000
28	0.000000	1.000000
29	1.000000	0.000000
30	0.000000	0.000000
31	0.000000	0.000000
32	1.000000	0.000000

SDPP2

添加整数约束：

```
model:
min=2*x1+4*x2+7*x3+3*x4+2*x5+3*x6+2*x7+2*x8+x9+4*x10+6*x11;
x1+x2=2;
x3-x1=0;
x5-x3=0;
x4-x2=0;
x6+x7-x4=0;
x8+x10-x6=0;
x11-x7=0;
x9-x5-x8=0;
-x9-x10-x11=-2;
0<=x1;
x1<=1;
0<=x2;
x2<=1;
0<=x3;
x3<=1;
0<=x4;
x4<=1;
0<=x5;
x5<=1;
0<=x6;
x6<=1;
0<=x7;
x7<=1;
0<=x8;
x8<=1;
0<=x9;
x9<=1;
0<=x10;
x10<=1;
0<=x11;
x11<=1;
@gin(x1);@gin(x2);@gin(x3);@gin(x4);@gin(x5);@gin(x6);
@gin(x7);@gin(x8);@gin(x9);@gin(x10);@gin(x11);
end
```

输出结果：

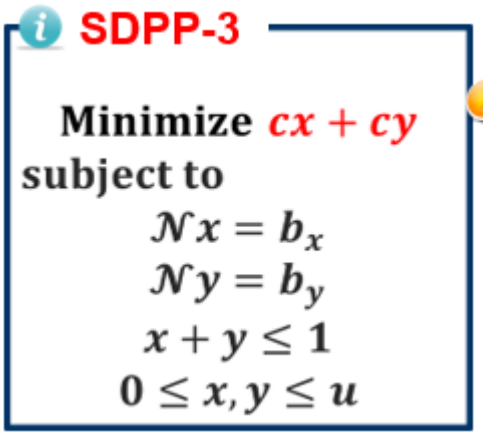
```
 Global optimal solution found.
 Objective value:                              26.00000
 Objective bound:                              26.00000
 Infeasibilities:                              0.000000
 Extended solver steps:                               0
 Total solver iterations:                             0


                     Variable           Value        Reduced Cost
                          X1        1.000000            2.000000
                          X2        1.000000            4.000000
                          X3        1.000000            7.000000
                          X4        1.000000            3.000000
                          X5        1.000000            2.000000
                          X6        1.000000            3.000000
```

X7	0.000000	2.000000
X8	0.000000	2.000000
X9	1.000000	1.000000
X10	1.000000	4.000000
X11	0.000000	6.000000

Row	Slack or Surplus	Dual Price
1	26.00000	-1.000000
2	0.000000	0.000000
3	0.000000	0.000000
4	0.000000	0.000000
5	0.000000	0.000000
6	0.000000	0.000000
7	0.000000	0.000000
8	0.000000	0.000000
9	0.000000	0.000000
10	0.000000	0.000000
11	1.000000	0.000000
12	0.000000	0.000000
13	1.000000	0.000000
14	0.000000	0.000000
15	1.000000	0.000000
16	0.000000	0.000000
17	1.000000	0.000000
18	0.000000	0.000000
19	1.000000	0.000000
20	0.000000	0.000000
21	1.000000	0.000000
22	0.000000	0.000000
23	0.000000	0.000000
24	1.000000	0.000000
25	0.000000	0.000000
26	1.000000	0.000000
27	1.000000	0.000000
28	0.000000	0.000000
29	1.000000	0.000000
30	0.000000	0.000000
31	0.000000	0.000000
32	1.000000	0.000000

SDPP3：

添加约束：



输入:

```
model:
ret1=2*x1+4*x2+7*x3+3*x4+2*x5+3*x6+2*x7+2*x8+x9+4*x10+6*x11;
ret2=2*y1+4*y2+7*y3+3*y4+2*y5+3*y6+2*y7+2*y8+y9+4*y10+6*y11;
min=ret1+ret2;
x1+x2=1;
y1+y2=1;
x3-x1=0;
y3-y1=0;
x5-x3=0;
y5-y3=0;
x4-x2=0;
y4-y2=0;
x6+x7-x4=0;
y6+y7-y4=0;
x8+x10-x6=0;
y8+y10-y6=0;
x11-x7=0;
y11-y7=0;
x9-x5-x8=0;
y9-y5-y8=0;
-x9-x10-x11=-1;
-y9-y10-y11=-1;
0<=x1;y1<=1;
0<=x2;y2<=1;
0<=x3;y3<=1;
0<=x4;y4<=1;
0<=x5;y5<=1;
0<=x6;y6<=1;
0<=x7;y7<=1;
0<=x8;y8<=1;
0<=x9;y9<=1;
0<=x10;y10<=1;
```

```
0<=x11;y11<=1;
x1+y1<1;x2+y2<1;x3+y3<1;x4+y4<1;x5+y5<1;x6+y6<1;
x7+y7<1;x8+y8<1;x9+y9<1;x10+y10<1;x11+y11<1;
end
```

输出结果：

```
  Global optimal solution found.
  Objective value:                              26.00000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
```

Variable	Value	Reduced Cost
RET1	14.00000	0.000000
X1	0.000000	0.000000
X2	1.000000	0.000000
X3	0.000000	0.000000
X4	1.000000	0.000000
X5	0.000000	0.000000
X6	1.000000	0.000000
X7	0.000000	0.000000
X8	0.000000	0.000000
X9	0.000000	0.000000
X10	1.000000	0.000000
X11	0.000000	0.000000
RET2	12.00000	0.000000
Y1	1.000000	0.000000
Y2	0.000000	0.000000
Y3	1.000000	0.000000
Y4	0.000000	0.000000
Y5	1.000000	0.000000
Y6	0.000000	0.000000
Y7	0.000000	0.000000
Y8	0.000000	0.000000
Y9	1.000000	0.000000
Y10	0.000000	0.000000
Y11	0.000000	0.000000

Row	Slack or Surplus	Dual Price
1	0.000000	-1.000000
2	0.000000	-1.000000
3	26.00000	-1.000000
4	0.000000	-15.00000
5	0.000000	-10.00000
6	0.000000	-13.00000

7	0.000000	-8.000000
8	0.000000	-5.000000
9	0.000000	0.000000
10	0.000000	-11.00000
11	0.000000	-6.000000
12	0.000000	-8.000000
13	0.000000	-3.000000
14	0.000000	-5.000000
15	0.000000	0.000000
16	0.000000	-6.000000
17	0.000000	-1.000000
18	0.000000	-3.000000
19	0.000000	2.000000
20	0.000000	0.000000
21	0.000000	5.000000
22	0.000000	0.000000
23	0.000000	0.000000
24	1.000000	0.000000
25	1.000000	0.000000
26	0.000000	0.000000
27	0.000000	0.000000
28	1.000000	0.000000
29	1.000000	0.000000
30	0.000000	0.000000
31	0.000000	0.000000
32	1.000000	0.000000
33	1.000000	0.000000
34	0.000000	0.000000
35	1.000000	0.000000
36	0.000000	0.000000
37	1.000000	0.000000
38	0.000000	0.000000
39	0.000000	0.000000
40	1.000000	0.000000
41	1.000000	0.000000
42	0.000000	0.000000
43	1.000000	0.000000
44	0.000000	0.000000
45	0.000000	0.000000
46	0.000000	1.000000
47	0.000000	0.000000
48	0.000000	0.000000
49	0.000000	0.000000
50	1.000000	0.000000

51	1.000000	0.000000
52	0.000000	2.000000
53	0.000000	1.000000
54	1.000000	0.000000

比较求解结果以及时间：

时间太短无法度量，推测 SDPP1-3 运算时间以此从低到高。
运算结果 SDPP3 可以很直观看出 c1x1(ret1)与 c2x2(ret2)的值也可以很清楚看出流的路径。