Linear Optimization

Assignment 1

Sahil Goyal – 2020326

Anshak Goel – 2020283

Deeptorshi Mondal – 2020294

1.

Code for revised Simplex Method from scratch:

```
import numpy as np
def revised_simplex(c,A,b):
    m,n=A.shape
    basic_variables=np.arange(0,m)
    nonbasic_variables=np.arange(m,n)
    initial_soln=np.zeros(n)
    initial_soln[basic_variables]=np.linalg.solve(A[:,basic_variables],b)
    while True:
        A_b=A[:,basic_variables]
        yT=np.linalg.solve(A_b.T,c[basic_variables])
        temp=True
        k=0
        for i in nonbasic_variables:
            if np.dot(yT,A[:,i])>c[i]:
                temp=False
                k=i
                break
        if temp:
            break
        else:
            db=np.linalg.solve(A_b,-A[:,k])
            d=np.zeros(n)
            d[basic_variables]=db
            d[nonbasic_variables]=np.zeros(n-m)
            d[k]=1
            if np.all(d>=0):
                return "unbounded"
            else:
                lambd=np.min(-initial_soln[d<0]/d[d<0])
                j=np.argmin(-initial_soln[d<0]/d[d<0])
                initial_soln=initial_soln+lambd*d
                nonbasic_variables=np.append(nonbasic_variables,basic_variable
s[j])
```

```
                nonbasic_variables=np.delete(nonbasic_variables,np.where(nonba
sic_variables==k))
                nonbasic_variables=np.sort(nonbasic_variables)
                basic_variables[j]=k
                basic_variables=np.sort(basic_variables)
    print(initial_soln)
    return initial_soln

if __name__ == "__main__":
    A=np.array([[6,8,-1,0],[7,12,0,-1]])
    b=np.array([100,120])
    c=np.array([12,20,0,0])
    x=revised_simplex(c,A,b)
    if type(x)==str:
        print("The problem is unbounded")
    else:
        print("The Optimal Solution is: ",np.dot(c,x))
```

The function revised_simplex takes 3 input parameters: the cost vector c, a matrix A which represents the constraints, and a vector b which represents the RHS of the equations. It gives as output the solution vector which can be multiplied with the cost vector in order to get the optimal value.

As an example, we have solved a question from lecture slides:

$$\min 12x_1 + 20x_2$$

$$\text{s.t.} \begin{cases} 6x_1 + 8x_2 \geq 100 \\ 7x_1 + 12x_2 \geq 120 \\ x_1, x_2 \geq 0 \end{cases}$$

Ans : $x_1 = 15, x_2 = 5/4$, Optimal value=205.

```
A=np.array([[6,8,-1,0],[7,12,0,-1]])
b=np.array([100,120])
c=np.array([12,20,0,0])
x=revised_simplex(c,A,b)
if type(x)==str:
    print("The problem is unbounded")
else:
    print("The Optimal Solution is: ",np.dot(c,x))
```

Output:

```
[15.    1.25  0.    0.  ]
The Optimal Solution is:  205.00000000000003
```

2.

2 For a general graph for this problem ILP would be

maximize $\sum x_e$

such that

$\sum x_e \leq 1$ for all $v \in V$

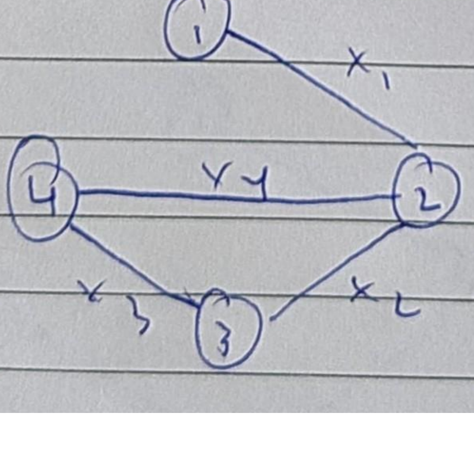
$x_e \in \{0, 1\}$ for all $e \in F$

Here V is set of vertices

F is set of edges

All edges are binary with 0 representing that they are not ~~involved~~ included and 1 representing that they are included

Graph:



The ILP will be

maximize $x_1 + x_2 + x_3 + x_4$

such that:

$$x_1 \leq 1$$
$$x_1 + x_2 + x_4 \leq 1$$
$$x_2 + x_3 \leq 1$$
$$x_3 + x_4 \leq 1$$

$$x_1, x_2, x_3, x_4 \in \{0, 1\}$$

```
c=np.array([-1,-1,-1,-1,0,0,0,0,0,0,0,0])
b=np.array([1,1,1,1,1,1,1,1])
A=np.array([
    [1,0,0,0,0,0,0,0,1,0,0,0],
    [1,1,0,1,0,0,0,0,0,1,0,0],
    [0,1,1,0,0,0,0,0,0,0,1,0],
    [0,0,1,1,0,0,0,0,0,0,0,1],
    [1,0,0,0,1,0,0,0,0,0,0,0],
    [0,1,0,0,0,1,0,0,0,0,0,0],
    [0,0,1,0,0,0,1,0,0,0,0,0],
    [0,0,0,1,0,0,0,1,0,0,0,0]
])
```

For the given adjacency matrix in network 1, we have solved the problem using the simplex code we have written. The output is:

```
[1. 0. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0.]
The Optimal Solution is:  2.0
```

3.

3) For a general graph for given problem, ILP would be of form

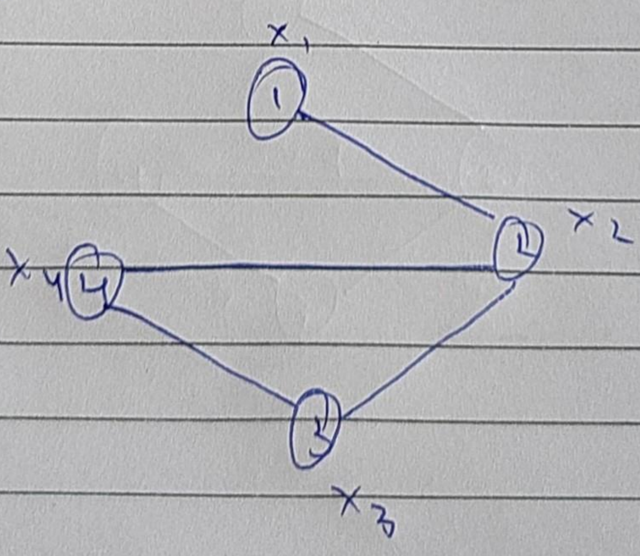
minimize $\sum x_i$

such that:

$$x_i + x_j \geq 1 \quad \text{for every } e_{ij} \in E$$

$$x_i \in \{0, 1\}$$

Here E is set of edges
All vertices (x_i) are binary with 0 representing that they are not included and 1 representing they are included
Graph:



The ILP will be:

minimise: $x_1 + x_2 + x_3 + x_4$

such that:

$$x_1 + x_2 \geq 1$$
$$x_2 + x_4 \geq 1$$
$$x_2 + x_3 \geq 1$$
$$x_3 + x_4 \geq 1$$

$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

```
c=np.array([1,1,1,1,0,0,0,0,0,0,0,0])
b=np.array([1,1,1,1,1,1,1,1])
A=np.array([[1,1,0,0,0,0,0,0,-1,0,0,0],
            [0,1,1,0,0,0,0,0,0,-1,0,0],
            [0,1,0,1,0,0,0,0,0,0,-1,0],
            [0,0,1,1,0,0,0,0,0,0,0,-1],
            [1,0,0,0,1,0,0,0,0,0,0,0],
            [0,1,0,0,0,1,0,0,0,0,0,0],
            [0,0,1,0,0,0,1,0,0,0,0,0],
            [0,0,0,1,0,0,0,1,0,0,0,0]])
```

For the given adjacency matrix in network 1, we have solved the problem using the simplex code we have written. The output is:

```
[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.  0.  0.  0. ]
The Optimal Solution is:  2.0
```

4.

We have written code for Interior Point Method from Scratch:

```
import numpy as np
import sympy

def ipopt(c,A,b):
    if np.linalg.matrix_rank(A)<np.min(A.shape):
        A=A[list(sympy.Matrix(A).T.rref()[1])]
    m,n=A.shape
    primal=np.ones(n,dtype=np.float64)
    dual=np.ones(m,dtype=np.float64)
    slack=np.ones(n,dtype=np.float64)
    flag=(abs(np.dot(primal,slack))>1e-6)
    while flag:
        sig=0.4
        mu=np.dot(primal,slack)/n
        b2=np.zeros(m+2*n,dtype=np.float64)
        b2[:m]=b-np.dot(A,primal)
        b2[m:m+n]=c-np.dot(A.T,dual)-slack
        b2[m+n:m+2*n]=sig*mu*np.ones(n,dtype=np.float64)-
np.dot(np.dot(np.diag(primal),np.diag(slack)),np.ones(n,dtype=np.float64))
        A2=np.zeros((m+2*n,m+2*n),dtype=np.float64)
        A2[:m,:n]=A
        A2[m:m+n,n:m+n]=A.T
        A2[m:m+n,n+m:m+2*n]=np.eye(n,dtype=np.float64)
        A2[m+n:m+2*n,:n]=np.diag(slack)
        A2[m+n:m+2*n,m+n:m+2*n]=np.diag(primal)
        delta=np.linalg.solve(A2,b2)
        dx,dl,ds=delta[:n],delta[n:m+n],delta[n+m:m+2*n]
        alpha_max=1
        neg_dx=np.where(dx<0)[0]
        neg_ds=np.where(ds<0)[0]
        if len(neg_dx)>0:
            alpha_max=min(alpha_max,np.min(-primal[neg_dx]/dx[neg_dx]))
        if len(neg_ds)>0:
            alpha_max=min(alpha_max,np.min(-slack[neg_ds]/ds[neg_ds]))
        alpha=min(0.99*alpha_max,1)
        primal+=alpha*dx
        dual+=alpha*dl
        slack+=alpha*ds
        flag=(abs(np.dot(primal,slack))>1e-6)
    return primal
```

```python
if __name__ == "__main__":
    A=np.array([[6,8,-1,0],[7,12,0,-1]],dtype=np.float64)
    b=np.array([100,120],dtype=np.float64)
    c=np.array([12,20,0,0],dtype=np.float64)
    x=ipopt(c,A,b)
    print(x)
    print("The Optimal Solution is: ",np.dot(c,x))
```

As an example, we have solved a question from lecture slides:

$$\min 12x_1 + 20x_2$$
$$\text{s.t.}$$
$$\begin{cases} 6x_1 + 8x_2 \geq 100 \\ 7x_1 + 12x_2 \geq 120 \\ x_1, x_2 \geq 0 \end{cases}$$

Ans : $x_1 = 15, x_2 = 5/4$, Optimal value=205.

```python
A=np.array([[6,8,-1,0],[7,12,0,-1]],dtype=np.float64)
b=np.array([100,120],dtype=np.float64)
c=np.array([12,20,0,0],dtype=np.float64)
x=ipopt(c,A,b)
print(x)
print("The Optimal Solution is: ",np.dot(c,x))
```

Output:

```
[1.50000004e+01 1.24999976e+00 6.43515942e-07 1.07252627e-07]
The Optimal Solution is:  205.00000032175794
```

We have written the LPP for the given flow network

4 LP for network 2:

max $x_1 + x_4 + x_6$

such that:

$x_1 + x_2 + x_3 - x_{12} - x_{14} = 0$

$x_4 - x_2 - x_7 = 0$

$x_6 + x_7 + x_8 - x_9 - x_{15} - x_{18} = 0$

$x_9 + x_{10} + x_{11} - x_8 - x_{16} - x_{19} = 0$

$x_{12} - x_{10} - x_{22} = 0$

$x_{14} - x_3 - x_{11} - x_{27} = 0$

$x_{15} + x_{16} + x_{17} - x_{20} - x_{23} - x_{24} - x_{28} = 0$

$x_{18} + x_{19} + x_{20} + x_{21} - x_{17} - x_{25} - x_{29} = 0$

$x_{22} + x_{23} - x_{26} - x_{30} = 0$

$x_{24} + x_{25} + x_{26} - x_{21} - x_{31} = 0$

$x_1 \le 11$, $x_2 \le 3$, $x_3 \le 12$, $x_4 \le 15$, $x_5 \le 8$, $x_6 \le 10$, $x_7 \le 5$, $x_8 \le 4$, $x_9 \le 6$, $x_{10} \le 3$, $x_{11} \le 4$, $x_{12} \le 18$, $x_{13} \le 16$, $x_{14} \le 4$, $x_{15} \le 3$, $x_{16} \le 17$, $x_{17} \le 4$, $x_{18} \le 11$, $x_{19} \le 6$, $x_{20} \le 4$, $x_{21} \le 2$, $x_{22} \le 13$, $x_{23} \le 9$, $x_{24} \le 4$, $x_{25} \le 5$, $x_{26} \le 7$, $x_{27} \le 21$, $x_{28} \le 3$, $x_{29} \le 4$, $x_{30} \le 9$, $x_{31} \le 15$

$x_i \ge 0$ for all $i \in \{1, 2, \ldots, 31\}$

We have solved the problem using the IPOPT code we have written. The output is as follows:

```
[1.10000000e+01 2.99999999e+00 5.77062147e-01 7.99999998e+00
 4.00000000e+00 9.99999999e+00 4.99999999e+00 7.60620746e-01
 5.16637612e+00 1.75998388e+00 6.07097689e-01 1.10945224e+01
 8.00000000e+00 3.48253975e+00 2.26546903e+00 5.72429862e+00
 2.99916057e+00 8.32877558e+00 1.04853832e+00 1.00083943e+00
 4.93910506e-01 9.33453851e+00 4.63588794e+00 3.02491004e+00
 4.37087364e+00 5.82785181e+00 2.29837991e+00 2.32729080e+00
 3.50202963e+00 8.14257464e+00 1.27297250e+01 8.97657061e-09
 8.97657063e-09 1.14229379e+01 7.00000002e+00 4.00000000e+00
 8.97657053e-09 8.97657054e-09 3.23937925e+00 8.33623878e-01
 1.24001612e+00 3.39290231e+00 6.90547762e+00 8.00000000e+00
 5.17460253e-01 7.34530971e-01 1.12757014e+01 1.00083943e+00
 2.67122442e+00 4.95146168e+00 2.99916057e+00 1.50608949e+00
 3.66546149e+00 4.36411206e+00 9.75089958e-01 6.29126363e-01
 1.17214819e+00 1.87016201e+01 6.72709199e-01 4.97970370e-01
 8.57425361e-01 2.27027502e+00]
The Optimal Solution is:  28.999999964093718
```

We have written the Dual of the LPP for the given flow network

We will now convert the primal LPP to the dual LPP by multiplying y_i with the primal LPP

$y_1(0 \le x_1 \le 11)$
$y_2(0 \le x_2 \le 3)$
$y_3(0 \le x_3 \le 12)$
$y_4(0 \le x_4 \le 15)$
$y_5(0 \le x_5 \le 8)$
$y_6(0 \le x_6 \le 10)$
$y_7(0 \le x_7 \le 5)$
$y_8(0 \le x_8 \le 4)$
$y_9(0 \le x_9 \le 6)$
$y_{10}(0 \le x_{10} \le 3)$
$y_{11}(0 \le x_{11} \le 4)$
$y_{12}(0 \le x_{12} \le 18)$
$y_{13}(0 \le x_{13} \le 16)$
$y_{14}(0 \le x_{14} \le 4)$
$y_{15}(0 \le x_{15} \le 3)$
$y_{16}(0 \le x_{16} \le 17)$
$y_{17}(0 \le x_{17} \le 4)$
$y_{18}(0 \le x_{18} \le 11)$
$y_{19}(0 \le x_{19} \le 6)$
$y_{20}(0 \le x_{20} \le 4)$
$y_{21}(0 \le x_{21} \le 2)$
$y_{22}(0 \le x_{22} \le 13)$
$y_{23}(0 \le x_{23} \le 9)$

My NoteBook

$y_{24}\ (0 \le x_{24} \le 4)$

$y_{25}\ (0 \le x_{25} \le 5)$

$y_{26}\ (0 \le x_{26} \le 7)$

$y_{27}\ (0 \le x_{27} \le 21)$

$y_{28}\ (0 \le x_{28} \le 3)$

$y_{29}\ (0 \le x_{29} \le 4)$

$y_{30}\ (0 \le x_{30} \le 9)$

$y_{31}\ (0 \le x_{31} \le 15)$

$y_{32}\ (x_1 + x_2 + x_3 - x_{12} - x_{14}) = 0$

$y_{33}\ (x_4 - x_2 - x_7) = 0$

$y_{34}\ (x_6 + x_7 + x_8 - x_9 - x_{15} - x_{18}) = 0$

$y_{35}\ (x_9 + x_{10} + x_{11} - x_8 - x_{16} - x_{19}) = 0$

$y_{36}\ (x_{12} - x_{10} - x_{22}) = 0$

$y_{37}\ (x_{14} - x_3 - x_{11} - x_{27}) = 0$

$y_{38}\ (x_{15} + x_{16} + x_{17} - x_{20} - x_{23} - x_{24} - x_{28}) = 0$

$y_{39}\ (x_{18} + x_{19} + x_{20} + x_{21} - x_{17} - x_{25} - x_{29}) = 0$

$y_{40}\ (x_{22} + x_{23} - x_{26} - x_{30}) = 0$

$y_{41}\ (x_{24} + x_{25} + x_{26} - x_{21} - x_{31}) = 0$

Dual form of LPP

min: $11y_1 + 3y_2 + 12y_3 + 15y_4 + 8y_5 + 10y_6 + 5y_7 + 4y_8 + 6y_9 + 3y_{10} + 4y_{11} + 18y_{12} + 16y_{13} + 4y_{14} + 3y_{15} + 17y_{16} + 4y_{17} + 11y_{18} + 6y_{19} + 4y_{20} + 2y_{21} + 13y_{22} + 9y_{23} + 4y_{24} + 5y_{25} + 7y_{26} + 21y_{27} + 3y_{28} + 4y_{29} + 9y_{30} + 15y_{31}$

such that :

$y_1 + y_{32} \geq 1$

$y_2 + y_{32} - y_{33} \geq 0$

$y_3 + y_{32} - y_{37} \geq 0$

$y_4 + y_{33} \geq 1$

$y_5 \geq 0$

$y_6 + y_{34} \geq 1$

$y_7 - y_{33} + y_{34} \geq 0$

$y_8 + y_{34} - y_{35} \geq 0$

$y_9 - y_{34} + y_{35} \geq 0$

$y_{10} + y_{35} - y_{36} \geq 0$

$y_{11} + y_{35} - y_{37} \geq 0$

$y_{12} - y_{32} + y_{36} \geq 0$

$y_{13} \geq 0$

$y_{14} - y_{32} + y_{37} \geq 0$

$y_{15} - y_{37} + y_{38} \geq 0$

$y_{16} - y_{35} + y_{38} \geq 0$

$$y_{17} + y_{38} - y_{39} \geq 0$$
$$y_{18} - y_{34} + y_{39} \geq 0$$
$$y_{19} - y_{35} + y_{39} \geq 0$$
$$y_{20} - y_{38} + y_{39} \geq 0$$
$$y_{21} + y_{39} - y_{41} \geq 0$$
$$y_{22} - y_{36} + y_{40} \geq 0$$
$$y_{23} - y_{38} + y_{40} \geq 0$$
$$y_{24} - y_{38} + y_{41} \geq 0$$
$$y_{25} - y_{39} + y_{41} \geq 0$$
$$y_{26} - y_{40} + y_{41} \geq 0$$
$$y_{27} - y_{37} \geq 0$$
$$y_{28} - y_{38} \geq 0$$
$$y_{29} - y_{39} \geq 0$$
$$y_{30} - y_{40} \geq 0$$
$$y_{31} - y_{41} \geq 0$$

$y_i \geq 0$ for all $i \in \{1, \ldots, 31\}$

$y_i \in \mathbb{R}$ for all $i \in \{32, \ldots, 41\}$

We have solved the problem using the IPOPT code we have written. The output is as follows:

```
[9.99999986e-01 9.99999988e-01 1.06485393e-09 1.73790465e-09
 3.04133280e-09 9.99999977e-01 9.99999978e-01 3.75080501e-09
 1.45828538e-08 9.75275038e-09 3.58562266e-09 1.76110165e-09
 1.52066636e-09 2.35501611e-08 1.67185302e-08 1.07855052e-09
 1.21148800e-08 4.54881940e-09 2.45563813e-09 4.06077923e-09
 8.07556373e-09 3.32342894e-09 2.78563084e-09 1.24431587e-08
 1.93613008e-08 1.03953197e-08 6.50573432e-10 1.79372791e-08
 2.44104928e-08 1.41686711e-08 5.36544017e-09 2.19887428e+06
 2.19887428e+06 2.19887433e+06 2.19887333e+06 2.19885193e+06
 2.19885193e+06 2.19881280e+06 2.19881280e+06 2.19891275e+06
 2.19891275e+06 2.19888741e+06 2.19888741e+06 2.19876228e+06
 2.19876228e+06 2.19883363e+06 2.19883363e+06 2.19889273e+06
 2.19889273e+06 2.19890445e+06 2.19890445e+06 1.10593911e-09
 4.05510956e-09 2.11353054e-08 1.52066615e-09 3.04133280e-09
 1.21653300e-09 2.43306593e-09 1.60781953e-08 2.35608019e-09
 6.94087262e-09 2.00353560e-08 1.09674581e-09 1.52066636e-09
 3.49238902e-09 5.36478162e-09 2.12655640e-09 4.06072236e-09
 1.46119343e-09 1.16306882e-08 1.21143711e-08 2.46480073e-08
 1.30256307e-09 2.62589374e-09 4.02158708e-09 2.78288974e-09
 2.08677345e-09 5.28786986e-09 5.23887337e-09 3.47381983e-09
 1.49402972e-09 9.55443628e-10]
The Optimal Solution is:  29.000000705402222
```

5.

The following results are obtained when we solve the LPP's we made in the previous questions.

For Question 2:

Code:

```
reset;

var x{1..4} >= 0 binary;

maximize z: x[1]+x[2]+x[3]+x[4];

s.t. c1: x[1]<=1;
s.t. c2: x[1]+x[2]+x[4]<=1;
s.t. c3: x[3]+x[2]<=1;
s.t. c4: x[3]+x[4]<=1;

option solver cplex;

solve;

display x,z;
```

Output:

```
x [*] :=
1  1
2  0
3  1
4  0
;

z = 2
```

For Question 3:

Code:

```
reset;

var x{1..4} >= 0 binary;

minimize z: x[1]+x[2]+x[3]+x[4];

s.t. c1: x[1]+x[2]>=1;
s.t. c2: x[2]+x[3]>=1;
s.t. c3: x[2]+x[4]>=1;
s.t. c4: x[3]+x[4]>=1;

option solver cplex;

solve;

display x,z;
```

Output:

```
x [*] :=
1  0
2  1
3  1
4  0
;

z = 2
```

For Question 4:

Max-Flow:

Code:

```
reset;

var x{1..31} >= 0;

maximize z: x[1] + x[4] + x[6];

s.t. c1: x[1]+x[2]+x[3]-x[12]-x[14]=0;
s.t. c2: x[4]-x[2]-x[7]=0;
s.t. c3: x[6]+x[7]+x[8]-x[9]-x[15]-x[18]=0;
s.t. c4: x[9]+x[10]+x[11]-x[8]-x[16]-x[19]=0;
s.t. c5: x[12]-x[10]-x[22]=0;
s.t. c6: x[14]-x[3]-x[11]-x[27]=0;
s.t. c7: x[15]+x[16]+x[17]-x[20]-x[23]-x[24]-x[28]=0;
```

```
s.t. c8: x[18]+x[19]+x[20]+x[21]-x[17]-x[25]-x[29]=0;
s.t. c9: x[22]+x[23]-x[26]-x[30]=0;
s.t. c10: x[24]+x[25]+x[26]-x[21]-x[31]=0;
s.t. c11: x[1]<=11;
s.t. c12: x[2]<=3;
s.t. c13: x[3]<=12;
s.t. c14: x[4]<=15;
s.t. c15: x[5]<=8;
s.t. c16: x[6]<=10;
s.t. c17: x[7]<=5;
s.t. c18: x[8]<=4;
s.t. c19: x[9]<=6;
s.t. c20: x[10]<=3;
s.t. c21: x[11]<=4;
s.t. c22: x[12]<=18;
s.t. c23: x[13]<=16;
s.t. c24: x[14]<=4;
s.t. c25: x[15]<=3;
s.t. c26: x[16]<=17;
s.t. c27: x[17]<=4;
s.t. c28: x[18]<=11;
s.t. c29: x[19]<=6;
s.t. c30: x[20]<=4;
s.t. c31: x[21]<=2;
s.t. c32: x[22]<=13;
s.t. c33: x[23]<=9;
s.t. c34: x[24]<=4;
s.t. c35: x[25]<=5;
s.t. c36: x[26]<=7;
s.t. c37: x[27]<=21;
s.t. c38: x[28]<=3;
s.t. c39: x[29]<=4;
s.t. c40: x[30]<=9;
s.t. c41: x[31]<=15;

option solver cplex;

solve;

display x, z;
```

We formulated the LPP of flow network and have displayed the solution below.

```
x [*] :=
 1 11     5  0     9  4    13  0    17  2    21  0    25  5
 2  3     6 10    10  1    14  4    18 11    22  9    26  0
 3  0     7  5    11  0    15  0    19  0    23  0    27  4
 4  8     8  0    12 10    16  5    20  0    24  4    28  3
;

z = 29
```

Min-Cut:

Code:

```
reset;

var y{1..31} >= 0;
var yu{1..10};
```

```
minimize z:
11*y[1]+3*y[2]+12*y[3]+15*y[4]+8*y[5]+10*y[6]+5*y[7]+4*y[8]+6*y[9]+3*y[10]+
4*y[11]+18*y[12]+16*y[13]+4*y[14]+3*y[15]+17*y[16]+4*y[17]+11*y[18]+6*y[19]
+4*y[20]+2*y[21]+13*y[22]+9*y[23]+4*y[24]+5*y[25]+7*y[26]+21*y[27]+3*y[28]+
4*y[29]+9*y[30]+15*y[31];

s.t. c1: y[1]+yu[1]>=1;
s.t. c2: y[2]+yu[1]-yu[2]>=0;
s.t. c3: y[3]+yu[1]-yu[6]>=0;
s.t. c4: y[4]+yu[2]>=1;
s.t. c5: y[5]>=0;
s.t. c6: y[6]+yu[3]>=1;
s.t. c7: y[7]-yu[2]+yu[3]>=0;
s.t. c8: y[8]+yu[3]-yu[4]>=0;
s.t. c9: y[9]-yu[3]+yu[4]>=0;
s.t. c10: y[10]+yu[4]-yu[5]>=0;
s.t. c11: y[11]+yu[4]-yu[6]>=0;
s.t. c12: y[12]-yu[1]+yu[5]>=0;
s.t. c13: y[13]>=0;
s.t. c14: y[14]-yu[1]+yu[6]>=0;
s.t. c15: y[15]-yu[3]+yu[7]>=0;
s.t. c16: y[16]-yu[4]+yu[7]>=0;
s.t. c17: y[17]+yu[7]-yu[8]>=0;
s.t. c18: y[18]-yu[3]+yu[8]>=0;
s.t. c19: y[19]-yu[4]+yu[8]>=0;
s.t. c20: y[20]-yu[7]+yu[8]>=0;
s.t. c21: y[21]+yu[8]-yu[10]>=0;
s.t. c22: y[22]-yu[5]+yu[9]>=0;
s.t. c23: y[23]-yu[7]+yu[9]>=0;
s.t. c24: y[24]-yu[7]+yu[10]>=0;
s.t. c25: y[25]-yu[8]+yu[10]>=0;
s.t. c26: y[26]-yu[9]+yu[10]>=0;
s.t. c27: y[27]-yu[6]>=0;
s.t. c28: y[28]-yu[7]>=0;
s.t. c29: y[29]-yu[8]>=0;
s.t. c30: y[30]-yu[9]>=0;
s.t. c31: y[31]-yu[10]>=0;

option solver cplex;

solve;

display y, yu, z;
```

We formulated the dual of the LPP of the network and displayed the output below.

```
:     y  yu     :=
1     1   0
2     1   1
3     0   0
4     0   0
5     0   0
6     1   0
7     1   0
8     0   0
9     0   0
10    0   0
11    0   .
12    0   .
13    0   .
14    0   .
15    0   .
16    0   .
17    0   .
18    0   .
19    0   .
20    0   .
21    0   .
22    0   .
23    0   .
24    0   .
25    0   .
26    0   .
27    0   .
28    0   .
29    0   .
30    0   .
31    0   .
;
```

```
z = 29
```

As we can see, the optimal value in both the primal and dual form is the same. Therefore, we can see that for the given flow network min-cut = max-flow = 29