LINEAR PROGRAMMING

IE-535

PROJECT REPORT

Name: Gokulakrishnan Swaminathan

PUID: 0030940426

Model: 22 (Lumber company)

Model: 22 (Lumber company) LP Model:

The lumber company has three sources and five markets. Rail and ship are two alternative modes of transporting the wood, and we need to find the optimal shipping plan such that the overall cost is reduced, while ensuring all the demand is met and sticking to the investment budget set by the company.

The unit costs of transporting by rail are:

	1	**2**	**3**	**4**	**5**
1	61	72	45	55	66
2	69	78	60	49	56
3	59	66	63	61	47

For ships, there are two components to the cost:

Unit shipping cost:

	1	**2**	**3**	**4**	**5**
1	31	38	24		35
2	36	43	28	24	31
3		33	36	32	26

Unit investment cost:

	1	**2**	**3**	**4**	**5**
1	275	303	238		285
2	293	318	270	250	265
3		283	275	268	240

Equivalent uniform annual cost of the ships may be calculated using the formula:

*Unit shipping cost + 0.1 * Unit investment cost.*

The values obtained are:

	1	**2**	**3**	**4**	**5**
1	58.5	68.3	47.8		63.5
2	65.3	74.8	55	49	57.5
3		61.3	63.5	58.8	50

Now, in order to derive the cost vector, we need to examine each Source-market pair and identify whether rail or ship provides the cheaper mode of transport.

After doing so, we can arrive at the optimal cost vector which we will use in the code as:

	1	2	3	4	5
1	58.5	68.3	45	55	63.5
2	65.3	74.8	55	49	56
3	59	61.3	63	58.8	47

Now, the linear program may be written as follows:

$$\min \sum_{i=0}^{3} \sum_{j=0}^{5} C_{ij}.x_{ij}$$

where x_{ij} denotes the number of units of wood (in million board feet) transported from source i to market j, and c_{ij} represents the corresponding cost vector.

s.t

$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} \leq 10$

$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} \leq 20$

$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} \leq 15$

$x_{11} + x_{21} + x_{31} \leq 7$

$x_{12} + x_{22} + x_{32} \leq 11$

$x_{13} + x_{23} + x_{33} \leq 9$

$x_{14} + x_{24} + x_{34} \leq 10$

$x_{15} + x_{25} + x_{35} \leq 8$

$x_{ij} \geq 0$, for all i and j

Also, to keep within budget, we will have to include an additional constraint related to the investment costs of ships is $275x_{11} + 303\,x_{12} + 285\,x_{15} + 293\,x_{21} + 318\,x_{22} + 270\,x_{23} + 283\,x_{32} + 268\,x_{34} \leq 6750$

We can see that total supply = total demand. So we can use equalities for all the constraints except the last.

Also for convenience $x_{11}, x_{12,} .., x_{35}$ have been taken as 1,2,3….,15 in the code.

Using this formulation, a general LP solver has been coded in Python, which can be used to solve any program just by giving the coefficients and constraints. The code is attached at the end. The snaps of the output are provided below:

Python Output:

Phase 1:

```
start phase 1
Tableau for phase 1
      0    1    2    3    4    5    6    7    8    9  ...   19   20   21
22  \
z   1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  ... -1.0 -1.0 -1.0 -
1.0
15  1.0  1.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
0.0
16  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  1.0  1.0  ...  0.0  0.0  0.0
0.0
17  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
0.0
23  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
0.0
24  0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  ... -1.0  0.0  0.0
0.0
25  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  ...  0.0 -1.0  0.0
0.0
26  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  ...  0.0  0.0 -1.0
0.0
27  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  ...  0.0  0.0  0.0 -
1.0

     23   24   25   26   27     B
z   0.0  0.0  0.0  0.0  0.0  45.0
15  0.0  0.0  0.0  0.0  0.0  10.0
16  0.0  0.0  0.0  0.0  0.0  20.0
17  0.0  0.0  0.0  0.0  0.0  15.0
23  1.0  0.0  0.0  0.0  0.0   7.0
24  0.0  1.0  0.0  0.0  0.0  11.0
25  0.0  0.0  1.0  0.0  0.0   9.0
26  0.0  0.0  0.0  1.0  0.0  10.0
27  0.0  0.0  0.0  0.0  1.0   8.0

[9 rows x 29 columns]
The Pivoting operations done to achieve optimal value is:

Variable to basis: 0
Variable to non basis: 23
Variable to basis: 1
Variable to non basis: 15
Variable to basis: 5
Variable to non basis: 0
Variable to basis: 6
Variable to non basis: 24
Variable to basis: 2
Variable to non basis: 25
Variable to basis: 3
Variable to non basis: 1
Variable to basis: 7
Variable to non basis: 16
Variable to basis: 10
Variable to non basis: 2
Variable to basis: 8
Variable to non basis: 5
Variable to basis: 11
Variable to non basis: 26
Variable to basis: 4
Variable to non basis: 17
```

```
optimality is achieved proceed to phase 2

Pivoting operations done to remove the artificial variables

Variable to basis: 15
Variable to non basis: 27


Tableau after phase 1:
       0    1    2    3    4    5    6    7    8    9  ...   19   20   21
22  \
z   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
0.0
3   1.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0 -1.0  ... -1.0 -1.0 -1.0
0.0
7   0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  ...  0.0 -1.0  0.0
0.0
4   0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  ...  0.0  0.0  0.0 -
1.0
8  -1.0 -1.0 -1.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  ...  1.0  1.0  0.0
0.0
6   1.0  1.0  0.0  0.0  0.0  1.0  1.0  0.0  0.0  0.0  ... -1.0  0.0  0.0
0.0
10  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
0.0
11 -1.0  0.0  0.0  0.0  0.0 -1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
0.0
15 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0  ...  1.0  1.0  1.0
1.0

     23   24   25   26   27    B
z  -1.0 -1.0 -1.0 -1.0 -1.0  0.0
3   1.0  1.0  1.0  1.0  0.0  2.0
7   0.0  0.0  1.0  0.0  0.0  9.0
4   0.0  0.0  0.0  0.0  1.0  8.0
8  -1.0 -1.0 -1.0  0.0  0.0  8.0
6   1.0  1.0  0.0  0.0  0.0  3.0
10  1.0  0.0  0.0  0.0  0.0  7.0
11 -1.0  0.0  0.0  0.0  0.0  8.0
15 -1.0 -1.0 -1.0 -1.0 -1.0 -0.0

[9 rows x 29 columns]
the tableau for phase 2
        0     1     2    3    4    5    6    7    8    9  ...   14   15
16  \
z   20.0  12.5  16.0 -0.0 -0.0  7.2 -0.0 -0.0 -0.0  1.5  ... -3.0 -0.0 -6.
0
3    1.0   1.0   1.0  1.0  0.0  0.0  0.0  0.0  0.0 -1.0  ... -1.0  0.0 -1.
0
7    0.0   0.0   1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  ...  0.0  0.0  0.
0
4    0.0   0.0   0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  ...  1.0  0.0  0.
0
8   -1.0  -1.0  -1.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  ...  1.0  0.0  1.
0
6    1.0   1.0   0.0  0.0  0.0  1.0  1.0  0.0  0.0  0.0  ... -1.0  0.0  0.
0
10   1.0   0.0   0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.
0
11  -1.0   0.0   0.0  0.0  0.0 -1.0  0.0  0.0  0.0  0.0  ...  1.0  0.0  0.
0
```

```
15  -0.0  -0.0  -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0  ... -0.0  1.0  1.
0

      17    18    19    20    21    22       B
z  -19.5 -78.5 -80.8 -61.0 -55.0 -63.5  2632.8
3   -1.0  -1.0  -1.0  -1.0  -1.0   0.0     2.0
7    0.0   0.0   0.0  -1.0   0.0   0.0     9.0
4    0.0   0.0   0.0   0.0   0.0  -1.0     8.0
8    1.0   1.0   1.0   1.0   0.0   0.0     8.0
6   -1.0  -1.0  -1.0   0.0   0.0   0.0     3.0
10   0.0  -1.0   0.0   0.0   0.0   0.0     7.0
11   1.0   1.0   0.0   0.0   0.0   0.0     8.0
15   1.0   1.0   1.0   1.0   1.0   1.0    -0.0
```

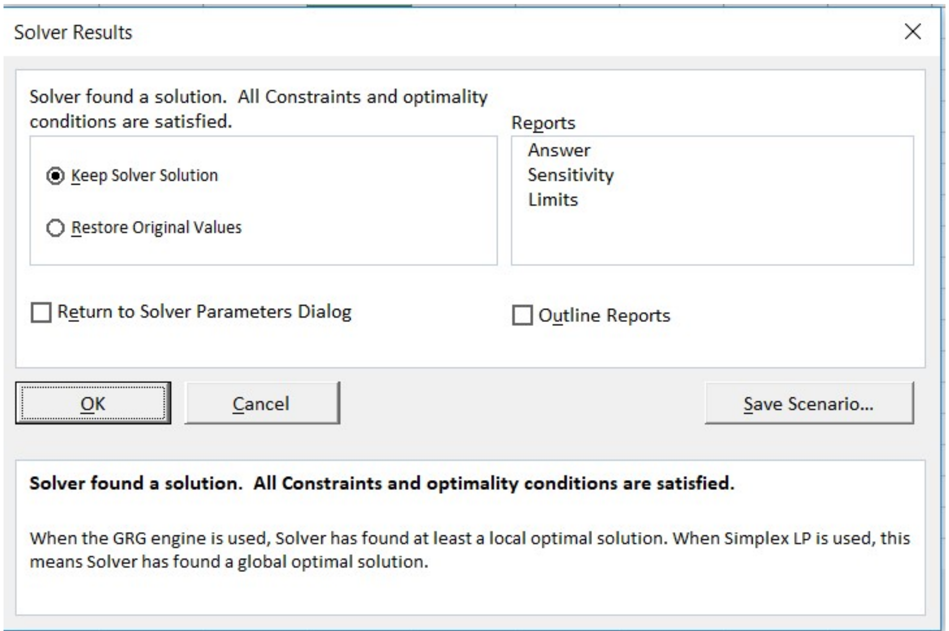
Phase 2:

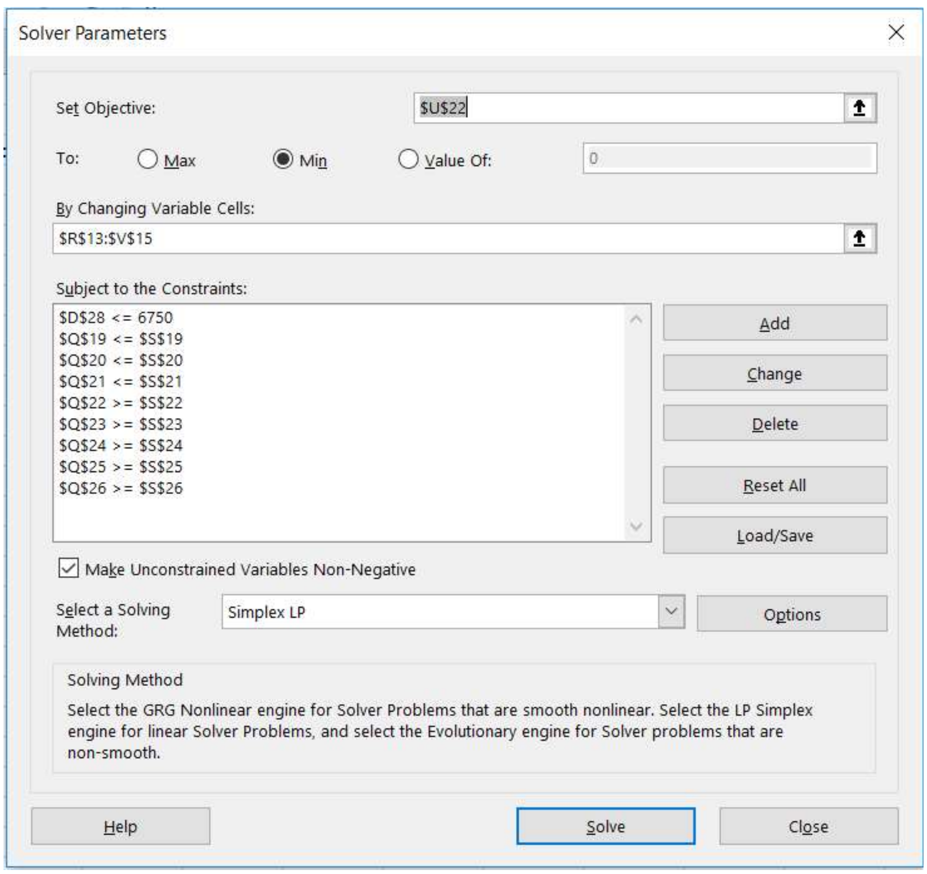
```
The Pivoting operations done to achieve optimal value is:

Variable to basis: 0
Variable to non basis: 3
Variable to basis: 9
Variable to non basis: 6
Variable to basis: 2
Variable to non basis: 4
Variable to basis: 16
Variable to non basis: 15
Variable to basis: 5
Variable to non basis: 7
Variable to basis: 14
Variable to non basis: 10


The Final tableau is:
       0    1    2     3     4    5    6    7    8    9 ...   14   15   1
6  \
z   0.0 -4.8  0.0 -12.8 -14.3  0.0 -4.5 -3.2 -0.0  0.0  ...  0.0 -6.8  0.0
0   1.0  1.0  0.0   1.0   1.0  0.0  0.0 -1.0  0.0  0.0  ...  0.0  1.0  0.0
5   0.0 -1.0  0.0  -1.0  -1.0  1.0  0.0  1.0  0.0  0.0  ...  0.0 -1.0  0.0
2   0.0  0.0  1.0   0.0   0.0  0.0  0.0  1.0  0.0  0.0  ...  0.0  0.0  0.0
8   0.0  0.0  0.0   1.0   0.0  0.0  0.0  0.0  1.0  0.0  ...  0.0  0.0  0.0
9   0.0  1.0  0.0   0.0   1.0  0.0  1.0  0.0  0.0  1.0  ...  0.0  0.0  0.0
14  0.0 -1.0  0.0   0.0   0.0  0.0 -1.0  0.0  0.0  0.0  ...  1.0  0.0  0.0
11  0.0  1.0  0.0   0.0   0.0  0.0  1.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
16  0.0  0.0  0.0   0.0   0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  1.0  1.0

     17    18    19    20    21    22       B
z  -9.0 -65.3 -70.3 -51.8 -49.0 -56.0  2431.6
0   0.0   0.0   0.0   1.0   0.0   0.0     1.0
5   0.0  -1.0   0.0  -1.0   0.0   0.0     6.0
2   0.0   0.0   0.0  -1.0   0.0   0.0     9.0
8   0.0   0.0   0.0   0.0  -1.0   0.0    10.0
9  -1.0   0.0  -1.0   0.0   0.0  -1.0     4.0
14  1.0   0.0   1.0   0.0   0.0   0.0     4.0
11  0.0   0.0  -1.0   0.0   0.0   0.0    11.0
16  1.0   1.0   1.0   1.0   1.0   1.0     0.0
```

```
The objective values are:

x0 : 1.000000

x1 : 0.000000

x2 : 9.000000

x3 : 0.000000

x4 : 0.000000

x5 : 6.000000

x6 : 0.000000

x7 : 0.000000

x8 : 10.000000

x9 : 4.000000

x10 : 0.000000

x11 : 11.000000

x12 : 0.000000

x13 : 0.000000

x14 : 4.000000

x15 : 0.000000

x16 : 0.000000

x17 : 0.000000

x18 : 0.000000

x19 : 0.000000

x20 : 0.000000

x21 : 0.000000

x22 : 0.000000


The objective function value is: 2431.600000
```

COMMERCIAL SOLVER:

The same LP has been solved using MS Excel Solver, the results of which are attached below:


Solver Results
Solver found a solution. All Constraints and optimality conditions are satisfied.
Reports
Answer
Sensitivity
Limits
Keep Solver Solution
Restore Original Values
Return to Solver Parameters Dialog
Outline Reports
OK
Cancel
Save Scenario...
Solver found a solution. All Constraints and optimality conditions are satisfied.
When the GRG engine is used, Solver has found at least a local optimal solution. When Simplex LP is used, this means Solver has found a global optimal solution.


Solver Parameters
Set Objective:
$U$22
To:
Max
Min
Value Of:
0
By Changing Variable Cells:
$R$13:$V$15
Subject to the Constraints:
$D$28 <= 6750
$Q$19 <= $S$19
$Q$20 <= $S$20
$Q$21 <= $S$21
$Q$22 >= $S$22
$Q$23 >= $S$23
$Q$24 >= $S$24
$Q$25 >= $S$25
$Q$26 >= $S$26
Add
Change
Delete
Reset All
Load/Save
Make Unconstrained Variables Non-Negative
Select a Solving Method:
Simplex LP
Options
Solving Method
Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.
Help
Solve
Close

Unit cost by rail market

	1	2	3	4	5
1	61	72	45	55	66
2	69	78	60	49	56
3	59	66	63	61	47

Unit cost by ship market

	1	2	3	4	5
1	31	38	24		35
2	36	43	28	24	31
3		33	36	32	26

Investment for ship market

	1	2	3	4	5
1	275	303	238		285
2	293	318	270	250	265
3		283	275	268	240

Total cost for ship market

	1	2	3	4	5
1	58.5	68.3	47.8		63.5
2	65.3	74.8	55	49	57.5
3		61.3	63.5	58.8	50

Final cost matrix (c)

	1	2	3	4	5
1	58.5	68.3	45	55	63.5
2	65.3	74.8	55	49	56
3	59	61.3	63	58.8	47

Solution matrix

	1	2	3	4	5
1	1	0	9	0	0
2	6	0	0	10	4
3	0	11	0	0	4

Constraint matrix

															Sum	**b**	
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	10	10	
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	20	20	**z**
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	15	15	
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	7	7	2431.6
0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	11	11	
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	9	9	
0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	10	10	
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	8	8	

Total Investment 5146

Result: Solver found a solution. All Constraints and optimality conditions are satisfied.

Solver Engine

Engine: Simplex LP

Solution Time: 0.031 Seconds.

Iterations: 20 Subproblems: 0

Solver Options

Max Time Unlimited, Iterations Unlimited, Precision 0.000001, Use Automatic Scaling

Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume NonNegative

Objective Cell (Min)

Cell	Name	Original Value	Final Value
$U$22	z	2431.6	2431.6

Variable Cells

Cell	Name	Original Value	Final Value	Integer
$R$13		1	1	Contin
$S$13	Solution matrix	0	0	Contin
$T$13	Investment for ship market	9	9	Contin
$U$13		0	0	Contin
$V$13		0	0	Contin
$R$14		6	6	Contin
$S$14	Solution matrix	0	0	Contin
$T$14	Investment for ship market	0	0	Contin
$U$14		10	10	Contin
$V$14		4	4	Contin
$R$15		0	0	Contin
$S$15	Solution matrix	11	11	Contin
$T$15	Investment for ship market	0	0	Contin
$U$15		0	0	Contin
$V$15		4	4	Contin

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$D$28	Total Investment	5146	$D$28<=6750	Not Binding	1604
$Q$19	Sum	10	$Q$19<=$S$19	Binding	0
$Q$20	Sum	20	$Q$20<=$S$20	Binding	0
$Q$21	Sum	15	$Q$21<=$S$21	Binding	0
$Q$22	Sum	7	$Q$22>=$S$22	Binding	0
$Q$23	Sum	11	$Q$23>=$S$23	Binding	0
$Q$24	Sum	9	$Q$24>=$S$24	Binding	0
$Q$25	Sum	10	$Q$25>=$S$25	Binding	0
$Q$26	Sum	8	$Q$26>=$S$26	Binding	0

CONCLUSION:

As can be seen, the optimal objective function value obtained from both is the same:

$z^* = 2431.6$

$x_{11} = 1$

$x_{13} = 9$

$x_{21} = 6$

$x_{24} = 10$

$x_{25} = 4$

$x_{32} = 11$

$x_{35} = 4$

Remaining x_{ij} are zeros.

Python CODE:

```python
import pandas as pd
import numpy as np

def unboundness(rt): #Checks the unboundness
    proceed = False
    if rt.all() < 0:
        print('The problem is infeasible')
        proceed = True
    return proceed

def cycling(rt,vtlb): #Resolves cycling using Bland's rule
    dum = np.extract(rt == np.inf, rt)
    dum = rt[rt == np.min(dum)].index
    dum1 = [x for x in dum if x not in vtlb]
    return min(dum1)

def check_cycling(rt, dum, vtlb): # Checks for cycling
    count = 0
    for i in dum:
        if (i == np.inf):
            count += 1
    if count == len(dum): #if cycling exists
        return cycling(rt,vtlb)
    else:
        dum = rt[rt == np.min(dum)].index #Resolves the conflict by choosing
```

```
the varaible with minimum index
        dum1 = [x for x in dum if x not in vtlb]
        return min(dum1)

def row_operations(tab, ib, inb, v2b, v2nb, basis,vtlb): #Row operations on t
he tableau
    dum = tab.loc[v2nb,v2nb]/tab.loc[v2nb,v2b] #pivoting the variable to basi
s
    tab.loc[v2nb,:] = dum * tab.loc[v2nb,:]
    for i in ib + ['z']: # row operations for the pivoted row
        if i != v2nb:
            dum = -tab.loc[i,v2b]/tab.loc[v2nb,v2b]
            tab.loc[i,:] += (dum * tab.loc[v2nb,:])
    dum = basis.index(v2nb) #updating the basis,non basis indices
    basis[dum] = v2b
    inb.remove(v2b)
    inb.append(v2nb)
    vtlb.append(v2nb)
    tab.index = ['z'] + basis
    ib = basis
    return tab, ib, inb,vtlb

def simplex(tab, ib, inb,vtlb):
    basis = ib
    ib = sorted(ib)
    inb = sorted(inb)
    find_pivot = tab.loc['z',inb] #Find the variable to enter the basis
    dum = np.extract(find_pivot >= 0 , find_pivot)
    v2bi = find_pivot[find_pivot == np.max(dum)].index
    v2b = min(v2bi)
    inter = []
    for i in ib: #Find the variable to leave the basis
        if tab.loc[i,v2b]>0:
            inter.append(i)
    rt = tab.loc[inter, 'B'] / tab.loc[inter, v2b]
    if unboundness(rt):
        return tab, ib, inb
    else:
        dum = np.extract(rt >= 0, rt)
        v2nb = check_cycling(rt, dum,vtlb)
        tab, ib, inb,vtlb = row_operations(tab, ib, inb, v2b, v2nb,basis,vtlb
)
        print("Variable to basis:", v2b)
        print("Variable to non basis:",v2nb)
        return tab, ib, inb,vtlb

def artificial_pivot(tab,ib,inb,vtlb): # Special case: when artificial variab
les are present in basis after reaching optimal value in phase 1
    basis = ib
    ib = sorted(ib)
    inb = sorted(inb)
```

```
    for i in inb: #choosing the first non zero variable to enter the basis
        if tab.loc['z',i]!=0:
            v2b = i
            break
    rt = tab.loc[ib,'B']/tab.loc[ib,v2b]
    if unboundness(rt):
        return tab, ib, inb
    else:
        dum = np.extract(rt >= 0, rt)
        v2nb = check_cycling(rt, dum,vtlb) #non basis variable exits the basi
s
        tab, ib, inb,vtlb = row_operations(tab, ib, inb, v2b, v2nb,basis,vtlb
)
        print("Variable to basis:", v2b)
        print("Variable to non basis:",v2nb)
        return tab, ib, inb,vtlb

def phase1(tableau,basis_var,non_basis_var,artificial_var,C): #the phase 1
    #Initializing the tableau for phase 1
    tableau.loc['z', basis_var] = 0
    for i in non_basis_var+['B']:
        tableau.loc['z', i] = sum(tableau.loc[artificial_var, i])
    print('Tableau for phase 1 \n',tableau)
    # Condition check to perform the pivoting operations
    for i in non_basis_var:
        if tableau.loc['z',i]>0:
            condition1 = True
            break
    vtlb1 = [] # This variable keeps the record of the variable that entered
the basis -> left the basis -> re-enters the basis
    print('The Pivoting operations done to achieve optimal value is: \n')
    while (condition1):
        tableau, basis_var, non_basis_var,vtlb1 = simplex(tableau,basis_var,n
on_basis_var,vtlb1)
        for i in non_basis_var:
            if tableau.loc['z',i]>0:
                condition1 = True
                break
            else:
                condition1 = False
    print('\n')
    if tableau.loc['z','B']==0: #checking the optimality of phase 1
        print('optimality is achieved proceed to phase 2 \n')
        # If there any artificial variables that are present in the basis
        print('Pivoting operations done to remove the artificial variables \n
')
        while (len([x for x in artificial_var if x in basis_var ])>0):
            a=[x for x in artificial_var if x in basis_var ]
            b = [x for x in (tableau.columns[:-1]) if x not in artificial_var
]
```

```
            '''
            This is where we'll deal with redundancy in the problem. After th
e phase 1 optimality is achieved and if
            artificial variables are present in the basis with corresponding
artificial RHS = 0 and that the only artificial
            variable present in the basis, that row can be deleted.
            '''
            if (tableau.loc[basis_var,'B'].all()==0) and (tableau.loc[a,b].al
l()==0).all():
                for i in basis_var:
                    if tableau.loc[i,'B']==0:
                        tableau = tableau.drop([i])
                        basis_var.remove(i)
                        print('x%d is eliminated and this is where the reduda
ncy in the problem is dealt \n'%(i))
            else:
                tableau, basis_var, non_basis_var,vtlb1 = artificial_pivot(ta
bleau,basis_var,non_basis_var,vtlb1)#cycling to move the artificial variables
out of basis
        print('\n')
        print('Tableau after phase 1: \n',tableau)
        # Converting the tableau from phase 1 -> phase 2
        tableau = tableau.drop(columns=artificial_var)
        dum = np.zeros((1,len(slack_var)+1))
        dum = np.append(C,dum,axis=1)
        tableau.loc['z',:] = dum
        for i in basis_var:
            if tableau.loc['z',i] != 0:
                dum = tableau.loc['z',i]
                tableau.loc['z',:] -= dum*tableau.loc[i,:]
        tableau.loc['z',:] = -1*tableau.loc['z',:]
        print('the tableau for phase 2 \n',tableau)
        return tableau,basis_var,non_basis_var
    else:
        print('The LP is infeasible \n')
        return tableau,basis_var,non_basis_var


def phase2(tableau,basis_var,non_basis_var,artificial_var=[]): #the phase 2
    #condition to check for positive reduced cost
    for i in non_basis_var:
        if tableau.loc['z',i]>0:
            condition = True
            break
    non_basis_var = [x for x in non_basis_var if x not in artificial_var]
    vtlb2 = []
    print('The Pivoting operations done to achieve optimal value is: \n')
    while (condition):
        tableau, basis_var, non_basis_var,vtlb2 = simplex(tableau,basis_var,n
on_basis_var,vtlb2)
```

```
        for i in non_basis_var:
            if tableau.loc['z',i]>0:
                condition = True
                break
            else:
                condition = False
    return tableau, basis_var, non_basis_var

# input the problem in minimization format
typeproblem=1 #if maximization problem change this to -1
A = np.array([[1,1,1,1,1,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,1,1,1,1,1,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,1,1,1,1,1],
              [1,0,0,0,0,1,0,0,0,0,1,0,0,0,0],
              [0,1,0,0,0,0,1,0,0,0,0,1,0,0,0],
              [0,0,1,0,0,0,0,1,0,0,0,0,1,0,0],
              [0,0,0,1,0,0,0,0,1,0,0,0,0,1,0],
              [0,0,0,0,1,0,0,0,0,1,0,0,0,0,1]])
b = np.array([[10],[20],[15],[7],[11],[9],[10],[8]])
C = np.array([[58.5,68.3,45,55,63.5,65.3,74.8,55,49,56,59,61.3,63,58.8,47]])
C = typeproblem*C
'''
For constraint type:
 if <= or < write 1
 if >= or > write -1
 if = write 0
'''
ConsType = np.array([1,1,1,-1,-1,-1,-1,-1])
# ConsType = np.array([0,0,0,0,0,0,0,0])

n_const,n_var = A.shape #The parameters

#getting the indices of artificial, basis and slack variables
slack_var = []
Iden_loc = [0]*n_const
artificial_var = []

# If there are negative b values then change the sign of the values in corres
ponding row
for i in range(n_const):
    if (b[i]<0):
        A[i,:] = -1*A[i,:]
        b[i] = -1*b[i]
        ConsType[i] = -1*ConsType[i]

# Adding the slack and Identity variables into the list and adding the new va
riables into A
for i in range(n_const):
    if (ConsType[i]>0):
        n_var = n_var + 1
        dum = np.zeros((n_const,1))
```

```
        A = np.append(A,dum,axis=1)
        slack_var.append(n_var-1)
        Iden_loc[i]=(n_var-1)
        A[i,n_var-1] = 1
    elif (ConsType[i]<0):
        n_var = n_var + 1
        dum = np.zeros((n_const,1))
        A = np.append(A,dum,axis=1)
        slack_var.append(n_var-1)
        A[i,n_var-1] = -1

# Adding the artificial and Identity variables into the list
for i in range(n_const):
    if (ConsType[i]<0 or Iden_loc[i]==0):
        n_var = n_var + 1
        dum = np.zeros((n_const,1))
        A = np.append(A,dum,axis=1)
        artificial_var.append(n_var-1)
        Iden_loc[i]=(n_var-1)
        A[i,n_var-1] = 1

print('the slack variables are:',slack_var)
print('the artificial variables are:',artificial_var)
print('the initial basis variables are:',Iden_loc)

the slack variables are: [15, 16, 17, 18, 19, 20, 21, 22]
the artificial variables are: [23, 24, 25, 26, 27]
the initial basis variables are: [15, 16, 17, 23, 24, 25, 26, 27]

basis_var = Iden_loc
non_basis_var = [x for x in range(n_var) if x not in basis_var] #Getting the
non basis variable
print('The non basis variable are:', non_basis_var)

The non basis variable are: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
, 18, 19, 20, 21, 22]

#converting the obatined the matrices into a tableau format
tableau = pd.DataFrame(A)
dum = np.zeros((1, n_var - C.shape[1]))
dum = np.append(C, dum, axis = 1)
dum = pd.DataFrame(dum)
tableau = pd.concat([dum, tableau])
tableau.index = range(n_const+1)
dum = np.append(np.zeros((1, 1)), b, axis  = 0)
dum = pd.DataFrame(dum)
dum
tableau['B'] = dum
tableau.index = ['z'] + basis_var
print('The converted matrices into Tableau format is: \n',tableau)
```

```
The converted matrices into Tableau format is:
        0     1     2     3     4     5     6     7     8     9  ...   19   2
0  \
z   58.5  68.3  45.0  55.0  63.5  65.3  74.8  55.0  49.0  56.0  ...  0.0  0.0
15   1.0   1.0   1.0   1.0   1.0   0.0   0.0   0.0   0.0   0.0  ...  0.0  0.0
16   0.0   0.0   0.0   0.0   0.0   1.0   1.0   1.0   1.0   1.0  ...  0.0  0.0
17   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...  0.0  0.0
23   1.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0  ...  0.0  0.0
24   0.0   1.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0  ... -1.0  0.0
25   0.0   0.0   1.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0  ...  0.0 -1.0
26   0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0   1.0   0.0  ...  0.0  0.0
27   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0   1.0  ...  0.0  0.0

     21   22   23   24   25   26   27     B
z   0.0  0.0  0.0  0.0  0.0  0.0  0.0   0.0
15  0.0  0.0  0.0  0.0  0.0  0.0  0.0  10.0
16  0.0  0.0  0.0  0.0  0.0  0.0  0.0  20.0
17  0.0  0.0  0.0  0.0  0.0  0.0  0.0  15.0
23  0.0  0.0  1.0  0.0  0.0  0.0  0.0   7.0
24  0.0  0.0  0.0  1.0  0.0  0.0  0.0  11.0
25  0.0  0.0  0.0  0.0  1.0  0.0  0.0   9.0
26 -1.0  0.0  0.0  0.0  0.0  1.0  0.0  10.0
27  0.0 -1.0  0.0  0.0  0.0  0.0  1.0   8.0

[9 rows x 29 columns]
```

```python
# Checking for the presence of artificial variables
if (len(artificial_var)==0):
    print('Proceed to phase two')
    tableau,basis_var,non_basis_var=phase2(tableau,basis_var,non_basis_var,ar
tificial_var)
    ph2=0
else:
    print('start phase 1')
    tableau,basis_var,non_basis_var=phase1(tableau,basis_var,non_basis_var,ar
tificial_var,C)
    ph2=1
```

```
start phase 1
Tableau for phase 1
       0    1    2    3    4    5    6    7    8    9  ...   19   20   21   2
2  \
z   1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  ... -1.0 -1.0 -1.0 -1.0
15  1.0  1.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0  0.0
16  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  1.0  1.0  ...  0.0  0.0  0.0  0.0
17  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0  0.0
23  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0  0.0
24  0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  ... -1.0  0.0  0.0  0.0
25  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  ...  0.0 -1.0  0.0  0.0
26  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  ...  0.0  0.0 -1.0  0.0
```

```
27  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  ...  0.0  0.0  0.0 -1.0

     23   24   25   26   27     B
z   0.0  0.0  0.0  0.0  0.0  45.0
15  0.0  0.0  0.0  0.0  0.0  10.0
16  0.0  0.0  0.0  0.0  0.0  20.0
17  0.0  0.0  0.0  0.0  0.0  15.0
23  1.0  0.0  0.0  0.0  0.0   7.0
24  0.0  1.0  0.0  0.0  0.0  11.0
25  0.0  0.0  1.0  0.0  0.0   9.0
26  0.0  0.0  0.0  1.0  0.0  10.0
27  0.0  0.0  0.0  0.0  1.0   8.0

[9 rows x 29 columns]
The Pivoting operations done to achieve optimal value is:

Variable to basis: 0
Variable to non basis: 23
Variable to basis: 1
Variable to non basis: 15
Variable to basis: 5
Variable to non basis: 0
Variable to basis: 6
Variable to non basis: 24
Variable to basis: 2
Variable to non basis: 25
Variable to basis: 3
Variable to non basis: 1
Variable to basis: 7
Variable to non basis: 16
Variable to basis: 10
Variable to non basis: 2
Variable to basis: 8
Variable to non basis: 5
Variable to basis: 11
Variable to non basis: 26
Variable to basis: 4
Variable to non basis: 17


optimality is achieved proceed to phase 2

Pivoting operations done to remove the artificial variables

Variable to basis: 15
Variable to non basis: 27


Tableau after phase 1:
       0    1    2    3    4    5    6    7    8    9  ...   19   20   21   2
```

```
2  \
z   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0  0.0
3   1.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0 -1.0  ... -1.0 -1.0 -1.0  0.0
7   0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  ...  0.0 -1.0  0.0  0.0
4   0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  ...  0.0  0.0  0.0 -1.0
8  -1.0 -1.0 -1.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  ...  1.0  1.0  0.0  0.0
6   1.0  1.0  0.0  0.0  0.0  1.0  1.0  0.0  0.0  0.0  ... -1.0  0.0  0.0  0.0
10  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0  0.0
11 -1.0  0.0  0.0  0.0  0.0 -1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0  0.0
15 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0  ...  1.0  1.0  1.0  1.0

     23   24   25   26   27    B
z  -1.0 -1.0 -1.0 -1.0 -1.0  0.0
3   1.0  1.0  1.0  1.0  0.0  2.0
7   0.0  0.0  1.0  0.0  0.0  9.0
4   0.0  0.0  0.0  0.0  1.0  8.0
8  -1.0 -1.0 -1.0  0.0  0.0  8.0
6   1.0  1.0  0.0  0.0  0.0  3.0
10  1.0  0.0  0.0  0.0  0.0  7.0
11 -1.0  0.0  0.0  0.0  0.0  8.0
15 -1.0 -1.0 -1.0 -1.0 -1.0 -0.0

[9 rows x 29 columns]
the tableau for phase 2
        0     1     2    3    4    5    6    7    8    9  ...   14   15   16  \
z   20.0  12.5  16.0 -0.0 -0.0  7.2 -0.0 -0.0 -0.0  1.5  ... -3.0 -0.0 -6.0
3    1.0   1.0   1.0  1.0  0.0  0.0  0.0  0.0  0.0 -1.0  ... -1.0  0.0 -1.0
7    0.0   0.0   1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  ...  0.0  0.0  0.0
4    0.0   0.0   0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  ...  1.0  0.0  0.0
8   -1.0  -1.0  -1.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  ...  1.0  0.0  1.0
6    1.0   1.0   0.0  0.0  0.0  1.0  1.0  0.0  0.0  0.0  ... -1.0  0.0  0.0
10   1.0   0.0   0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
11  -1.0   0.0   0.0  0.0  0.0 -1.0  0.0  0.0  0.0  0.0  ...  1.0  0.0  0.0
15  -0.0  -0.0  -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0 -0.0  ... -0.0  1.0  1.0

      17    18    19    20    21    22       B
z  -19.5 -78.5 -80.8 -61.0 -55.0 -63.5  2632.8
3   -1.0  -1.0  -1.0  -1.0  -1.0   0.0     2.0
7    0.0   0.0   0.0  -1.0   0.0   0.0     9.0
4    0.0   0.0   0.0   0.0   0.0  -1.0     8.0
8    1.0   1.0   1.0   1.0   0.0   0.0     8.0
6   -1.0  -1.0  -1.0   0.0   0.0   0.0     3.0
10   0.0  -1.0   0.0   0.0   0.0   0.0     7.0
11   1.0   1.0   0.0   0.0   0.0   0.0     8.0
15   1.0   1.0   1.0   1.0   1.0   1.0    -0.0

[9 rows x 24 columns]
```

```
if ph2==1:
    tableau,basis_var,non_basis_var=phase2(tableau,basis_var,non_basis_var,ar
tificial_var)
```

```
The Pivoting operations done to achieve optimal value is:

Variable to basis: 0
Variable to non basis: 3
Variable to basis: 9
Variable to non basis: 6
Variable to basis: 2
Variable to non basis: 4
Variable to basis: 16
Variable to non basis: 15
Variable to basis: 5
Variable to non basis: 7
Variable to basis: 14
Variable to non basis: 10
```

```
print('The Final tableau is: \n', tableau)
```

```
The Final tableau is: 
       0    1    2     3     4    5    6    7    8    9  ...   14   15   16  \
z   0.0 -4.8  0.0 -12.8 -14.3  0.0 -4.5 -3.2 -0.0  0.0  ...  0.0 -6.8  0.0   
0   1.0  1.0  0.0   1.0   1.0  0.0  0.0 -1.0  0.0  0.0  ...  0.0  1.0  0.0   
5   0.0 -1.0  0.0  -1.0  -1.0  1.0  0.0  1.0  0.0  0.0  ...  0.0 -1.0  0.0   
2   0.0  0.0  1.0   0.0   0.0  0.0  0.0  1.0  0.0  0.0  ...  0.0  0.0  0.0   
8   0.0  0.0  0.0   1.0   0.0  0.0  0.0  0.0  1.0  0.0  ...  0.0  0.0  0.0   
9   0.0  1.0  0.0   0.0   1.0  0.0  1.0  0.0  0.0  1.0  ...  0.0  0.0  0.0   
14  0.0 -1.0  0.0   0.0   0.0  0.0 -1.0  0.0  0.0  0.0  ...  1.0  0.0  0.0   
11  0.0  1.0  0.0   0.0   0.0  0.0  1.0  0.0  0.0  0.0  ...  0.0  0.0  0.0   
16  0.0  0.0  0.0   0.0   0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  1.0  1.0   

     17    18    19    20    21    22       B  
z  -9.0 -65.3 -70.3 -51.8 -49.0 -56.0  2431.6  
0   0.0   0.0   0.0   1.0   0.0   0.0     1.0  
5   0.0  -1.0   0.0  -1.0   0.0   0.0     6.0  
2   0.0   0.0   0.0  -1.0   0.0   0.0     9.0  
8   0.0   0.0   0.0   0.0  -1.0   0.0    10.0  
9  -1.0   0.0  -1.0   0.0   0.0  -1.0     4.0  
14  1.0   0.0   1.0   0.0   0.0   0.0     4.0  
11  0.0   0.0  -1.0   0.0   0.0   0.0    11.0  
16  1.0   1.0   1.0   1.0   1.0   1.0     0.0  

[9 rows x 24 columns]
```

```
print('The objective values are: \n')
for i in tableau.columns:
    if i in basis_var:
        print('x%d : %f \n'%(i,tableau.loc[i,'B']))
```

```python
    elif i in non_basis_var:
        print('x%d : %f \n'%(i,0))
    else:
        print('\n')
        print('The objective function value is: %f'%(typeproblem*tableau.loc[
'z',i]))
```

```
The objective values are:

x0 : 1.000000

x1 : 0.000000

x2 : 9.000000

x3 : 0.000000

x4 : 0.000000

x5 : 6.000000

x6 : 0.000000

x7 : 0.000000

x8 : 10.000000

x9 : 4.000000

x10 : 0.000000

x11 : 11.000000

x12 : 0.000000

x13 : 0.000000

x14 : 4.000000

x15 : 0.000000

x16 : 0.000000

x17 : 0.000000

x18 : 0.000000

x19 : 0.000000

x20 : 0.000000
```

```
x21 : 0.000000

x22 : 0.000000


The objective function value is: 2431.600000
```