SAYANTAN RAHA

Roll # : BAI09056

IIMB - BAI09 - Assignment 3

In [2]:
```
from IPython.display import HTML

HTML('''<script>
code_show=true;
function code_toggle() {
 if (code_show){
 $('div.input').hide();
 } else {
 $('div.input').show();
 }
 code_show = !code_show
}
$( document ).ready(code_toggle);
</script>
<form action="javascript:code_toggle()"><input type="submit" value="Toggle on/off Code"></form>''')
```

Out[2]: Toggle on/off Code

In [1]:
```
import warnings
warnings.filterwarnings('ignore')

%load_ext rpy2.ipython
```

In [2]:
```
import pandas as pd
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:
```
from pulp import *
import pyomo.environ as pe
```

Q1 - 1

The following problem has been obtained by solving the set of following LP formulation:

Maximize Objective: 50 x_dh + 45 x_fn + 10 x_kbh + 60 x_kh

Decision variables:

- x_fn: quantity of FN
- x_dh: quantity of DH
- x_kh: quantity of KH
- x_kbh: quantity of KBH

Subject To

- Demand_1: x_fn <= 250
- Demand_2: x_dh <= 120
- Demand_3: x_kh <= 90
- Demand_4: x_kbh <= 550
- Non_Zero_Constraint_1: x_fn >= 0
- Non_Zero_Constraint_2: x_dh >= 0
- Non_Zero_Constraint_3: x_kh >= 0
- Non_Zero_Constraint_4: x_kbh >= 0
- Supply_1: 0.2 x_dh + 0.3 x_fn + 0.3 x_kbh + 0.4 x_kh <= 500
- Supply_2: 0.5 x_dh + 0.4 x_fn + 0.8 x_kbh + 0.8 x_kh <= 450
- Supply_3: 0.3 x_dh + 0.4 x_fn + 0.1 x_kbh + 0.1 x_kh <= 75
- Supply_4: 0.3 x_dh + 0.2 x_fn + 0.5 x_kbh + 0.6 x_kh <= 300
- Supply_5: 0.5 x_dh + 0.5 x_fn + 0.8 x_kbh + 0.4 x_kh <= 200

- The optimal values for the Problem are as follows.

In [8]:
```
# initialize the model
prob = LpProblem("HVMix", LpMaximize)
#List of decision variables
vehicles = ['fn', 'dh', 'kh', 'kbh']#, 'rm_M', 'rm_S', 'rm_FN', 'rm_CM', 'rm_G']
# create a dictionary of pulp variables with keys from ingredients
# the default lower bound is -inf
x = pulp.LpVariable.dict('x_%s', vehicles, lowBound = 0)

# Objective function

profit = [45, 50, 60, 10]
cost = dict(zip(vehicles, profit))

prob += sum([cost[i] * x[i] for i in vehicles]), "Objective" #['fn', 'dh', 'kh', 'kbh']

# Constraints

prob += x['fn'] <= 250, "Demand 1"
prob += x['dh'] <= 120, "Demand 2"
prob += x['kh'] <= 90, "Demand 3"
prob += x['kbh'] <= 550, "Demand 4"

prob += x['fn'] >= 0, "Non Zero Constraint 1"
prob += x['dh'] >= 0, "Non Zero Constraint 2"
prob += x['kh'] >= 0, "Non Zero Constraint 3"
prob += x['kbh'] >= 0, "Non Zero Constraint 4"

prob +=  .3 * x['fn'] + .2 * x['dh'] + .4 * x['kh'] + .3 * x['kbh'] <= 500, "Supply 1"
prob +=  .4 * x['fn'] + .5 * x['dh'] + .8 * x['kh'] + .8 * x['kbh'] <= 450, "Supply 2"
prob +=  .4 * x['fn'] + .3 * x['dh'] + .1 * x['kh'] + .1 * x['kbh'] <= 75, "Supply 3"
prob +=  .2 * x['fn'] + .3 * x['dh'] + .6 * x['kh'] + .5 * x['kbh'] <= 300, "Supply 4"
prob +=  .5 * x['fn'] + .5 * x['dh'] + .4 * x['kh'] + .8 * x['kbh'] <= 200, "Supply 5"

#prob.writeLP("tomatoMix.lp")

status = prob.solve(GLPK(options=["--ranges","HNMix.sen"]))
#print(status)
#print the result
for vehicle in vehicles:
    print(' {} :: {} ::'.format(vehicle,
    x[vehicle].value()))

print("Objective", value(prob.objective))
prob.writeLP("HNMix.lp")
```

```
 fn :: 75.0 ::
 dh :: 120.0 ::
 kh :: 90.0 ::
 kbh :: 0.0 ::
Objective 14775.0
```

```
# %load HNMix.sen
GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                          Page   1

Problem:
Objective:  Objective = 14775 (MAXimum)

   No. Row name     St      Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 Demand_1     BS      75.00000     175.00000          -Inf       50.37037      -5.00000   14400.00000 x_kbh
                                             .         250.00000      165.00000      21.66667   16400.00000 Demand_2

     2 Demand_2     NU     120.00000         .              -Inf           .        -16.25000   12825.00000
Non_Zero_Constraint_2
                                          16.25000     120.00000      220.00000          +Inf   16400.00000

Non_Zero_Constraint_1

     3 Demand_3     NU      90.00000         .              -Inf           .        -48.75000   10387.50000
Non_Zero_Constraint_3
                                          48.75000      90.00000      331.81818          +Inf   26563.63636 Supply_5

     4 Demand_4     BS          .          550.00000          -Inf           .             -Inf   14775.00000
                                             .         550.00000       98.51852       1.25000   14775.00000 x_kbh

     5 Non_Zero_Constraint_1
                    BS      75.00000     -75.00000           .         50.37037      -5.00000   14400.00000 x_kbh
                                             .              +Inf      165.00000      21.66667   16400.00000 Demand_2

     6 Non_Zero_Constraint_2
                    BS     120.00000    -120.00000           .             .        -16.25000   12825.00000 Demand_2
                                             .              +Inf      120.00000          +Inf          +Inf

     7 Non_Zero_Constraint_3
                    BS      90.00000     -90.00000           .             .        -48.75000   10387.50000 Demand_3
                                             .              +Inf       90.00000          +Inf          +Inf

     8 Non_Zero_Constraint_4
                    BS          .             .              .             .             -Inf   14775.00000
                                             .              +Inf       98.51852       1.25000   14775.00000 x_kbh

     9 Supply_1     BS      82.50000     417.50000          -Inf       53.25000    -150.00000    2400.00000 Demand_3
                                             .         500.00000      104.66667       5.55556   15233.33333 x_kbh

    10 Supply_2     BS     162.00000     288.00000          -Inf       99.00000     -69.64286    3492.85714 Demand_3
                                             .         450.00000      230.96296       1.78571   15064.28571 x_kbh
```

```
GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                                         Page   2

Problem:    
Objective:  Objective = 14775 (MAXimum)

   No. Row name     St      Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
    11 Supply_3     NU      75.00000        .               -Inf       45.00000    -112.50000   11400.00000 
Non_Zero_Constraint_1
                                         112.50000      75.00000      128.20000          +Inf   20760.00000 Supply_5

    12 Supply_4     BS     105.00000     195.00000          -Inf       55.50000     -88.63636    5468.18182 Demand_3
                                            .          300.00000      149.33333       2.77778   15066.66667 x_kbh

    13 Supply_5     BS     133.50000      66.50000          -Inf       96.00000     -90.00000    2760.00000 Supply_3
                                            .          200.00000      336.00000       1.85185   15022.22222 x_kbh

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                                         Page   3

Problem:    
Objective:  Objective = 14775 (MAXimum)

   No. Column name  St      Activity      Obj coef   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 x_dh         BS     120.00000      50.00000          .              .         33.75000   12825.00000 Demand_2
                                            .               +Inf      120.00000          +Inf          +Inf 

     2 x_fn         BS      75.00000      45.00000          .          50.37037      40.00000   14400.00000 x_kbh
                                            .               +Inf      165.00000      66.66667   16400.00000 Demand_2

     3 x_kbh        NL        .           10.00000          .              .             -Inf   14775.00000 
Non_Zero_Constraint_4
                                          -1.25000          +Inf       98.51852      11.25000   14651.85185 Supply_5

     4 x_kh         BS      90.00000      60.00000          .              .         11.25000   10387.50000 Demand_3
                                            .               +Inf       90.00000          +Inf          +Inf 

End of report
```

Q1 - 2

- From the **SENSITIVITY ANALYSIS REPORT** report above we can clearly see that amount of KBH to be produced to maximize profit under current profit rates is **Zero (0)**

The **Reduced Cost / Marginal Cost for Objective Coefficient** for KBH is -1.25 implying the profit will reduce by 1.25 units if AH are to produce one unit of KBH. It is a Non-Basic variable with coefficient equal to Zero

Q1 - 3

- From the **SENSITIVITY ANALYSIS REPORT** report above we can clearly see that **Supply_1 Constraint (Availability of Maida = 500 Kg)** is not a binding constraint. There is already 417 Kg of extra (SLACK) Maida available with the supplier. Hence he should not be procuring the extra Maida from his friend.

We are assuming his friend will provide Maida at Market price and not free of cost

Q1 - 4

Assuming this question is for KH

- AH is producing 90Kg of KH @ Profit of 60/unit. Hence he can accept the order of 20Kg from the Halva Shop

Assuming this question is for additional 20 Kg of KH (above 90Kg)

- As per the sesitivity report we see that upper bound for KH production is 90, i.e. if they are to produce additional 20 Kg, then we need to change the Constraints and re-solve the LP. Current Optimal Solution will not remain same

Assuming this question is for KBH

- From the Sensitivity Report we can see that, Reduced Cost / Marginal Cost for Objective Coefficient for KBH is -1.25. In order for him to accept any orders for KBH the minimum value of Profit from KBH should be 11.25/unit. Hence he should increase the Profit on KBH by 1.25/unit, if he is to accept this order

Q1 - 5

- From the Sensitivity Report we can see that the Profit on DH can be reduced by max of 16.25/unit for the current solution to remain optimal. Hence providing a discount of 10 INR/unit of DH **does not change** the optimal production plan

Q1 - 6

- ASSUMPTIONS for the following solution
  - We are increasing the profit amounts by 20% implying the Profit of KBH will increase from 10 to 12

In [9]:
```
print("Increased Profit for DH = {}".format(1.2*45))
print("Increased Profit for FN = {}".format(1.2*50))
print("Increased Profit for KH = {}".format(1.2*60))
print("Increased Profit for KBH = {}".format(1.2*10))
```

```
Increased Profit for DH = 54.0
Increased Profit for FN = 60.0
Increased Profit for KH = 72.0
Increased Profit for KBH = 12.0
```

- Since the simultaneous increase in the coefficients of the Non-Basic Variables (non Zero) are withing the permissible ranges (as seen from the Sensitivity Report) and the sum of percentage increase is less than 100%, hence as per the 100% Rule there is **no change in the optimal Solution**
- We are still assuming that KBH is not being produced

In [10]:
```
newprofit = [54, 60, 72, 12]
quantity = [75, 120, 90, 0]

print("Current Profit due to change in Profit Values = {}".format(sum([newprofit[i] * quantity[i] for i in [0,1,2,3]])))
```

```
Current Profit due to change in Profit Values = 17730
```

Q1 - 7

- As per the sensitivity Report the Constraint Supply_3 is binding and has the highest Marginal Cost (112.5). This constraint corresponds to the Supply Constraints for Fruits and Nuts.
- What this implies:
  - Increasing availability of Fruits and Nuts from 75 Kg by one unit increases profit by 112.5 INR
  - The above is valid only in the amount of Fruits and Nuts are increases from current available level of 75 till 128. Beyond this range if availability is increased then the current shadow price will not hold true

Q1 - 8

- From the Sensitivity Report we can see that the Profit on DH can be reduced to 33.75/unit for the current solution to remain optimal.
- From the Sensitivity Report we can see that the Profit on KH can be reduced to 11.25/unit for the current solution to remain optimal.

- As per the problem statement the reduction in DH is 8/Unit and reduction in KH is 24/Unit
- We will compute the % change and use the 100% Rules to check if the changes are below 100% or not. Since both are Non-Basic Variables hence if the allowed change is less than 100% hence from using the 100% Rule for change in Objective coefficients we know the Optimal Solution will remain unchanged

In [13]:
```
print("% change in DH in the allowed direction = {}".format(8 / 16.25))
print("% change in KH in the allowed direction = {}".format(24 / (60-11.25)))
print()
print("Sum of the % changes in the allowed directions is < 100%, hence there is no change in the Optimal Solution")
```

```
% change in DH in the allowed direction = 0.49230769230769234
% change in KH in the allowed direction = 0.49230769230769234

Sum of the % changes in the allowed directions is < 100%, hence there is no change in the Optimal Solution
```

Q2 - a

Decision Variables: 15 Binary Variables for each compartment and each Fule Type. This will indicate which type of Fuel should be carried in which container.

e.g. ys1 = 1, will indicate Fuel S is being carried in Container 1 We use Python (PULP and GLPK Solver to solve the solution). Hence the DV will be as follows:

compartments = ['1', '2', '3', '4', '5']

- ys = LpVariable.dicts("Ys_", compartments, 0, None, cat = LpBinary) # 5 Variables for Fuel S
- yr = LpVariable.dicts("Yr_", compartments, 0, None, cat = LpBinary) # 5 Variables for Fuel R
- yu = LpVariable.dicts("Yu_", compartments, 0, None, cat = LpBinary) # 5 Variables for Fuel U

15 Binary Variables for quantity of fuel carried in each compartment. If corresponding binary indicator is 1, this will have non zero value.

e.g. ys1 = 1, s1 = 2800

In Python we will define these variables as follows:

- S = pulp.LpVariable.dict('S_%s', compartments, lowBound = 0)
- R = pulp.LpVariable.dict('R_%s', compartments, lowBound = 0)
- U = pulp.LpVariable.dict('U_%s', compartments, lowBound = 0)

In [ ]:

```
# initialize the model
prob = LpProblem("fuelMin", LpMinimize)
#List of decision variables
compartments = ['1', '2', '3', '4', '5']
# create a dictionary of pulp variables with keys from ingredients
S = pulp.LpVariable.dict('S_%s', compartments, lowBound = 0)
R = pulp.LpVariable.dict('R_%s', compartments, lowBound = 0)
U = pulp.LpVariable.dict('U_%s', compartments, lowBound = 0)

ys = LpVariable.dicts("Ys_", compartments, 0, None, cat = LpBinary)
yr = LpVariable.dicts("Yr_", compartments, 0, None, cat = LpBinary)
yu = LpVariable.dicts("Yu_", compartments, 0, None, cat = LpBinary)

# Objective function

loss = [10, 8, 6]

prob += 10* (2900 - sum([S[i] for i in compartments])) +\
        8* (4000 - sum([R[i] for i in compartments])) +\
        6* (4900 - sum([U[i] for i in compartments])) , "Objective"

# Constraints
prob += ys['1'] + yr['1'] + yu['1'] <= 1, "Integer Constraint for One Type of Fuel in Container 1"
prob += ys['2'] + yr['2'] + yu['2'] <= 1, "Integer Constraint for One Type of Fuel in Container 2"
prob += ys['3'] + yr['3'] + yu['3'] <= 1, "Integer Constraint for One Type of Fuel in Container 3"
prob += ys['4'] + yr['4'] + yu['4'] <= 1, "Integer Constraint for One Type of Fuel in Container 4"
prob += ys['5'] + yr['5'] + yu['5'] <= 1, "Integer Constraint for One Type of Fuel in Container 5"

prob += S['1'] <= 2700 * ys['1'], "Maximum Capacity of S Fuel if Container 1 has S"
prob += R['1'] <= 2700 * yr['1'], "Maximum Capacity of R Fuel if Container 1 has R"
prob += U['1'] <= 2700 * yu['1'], "Maximum Capacity of U Fuel if Container 1 has U"

prob += S['2'] <= 2800 * ys['2'], "Maximum Capacity of S Fuel if Container 2 has S"
prob += R['2'] <= 2800 * yr['2'], "Maximum Capacity of R Fuel if Container 2 has R"
prob += U['2'] <= 2800 * yu['2'], "Maximum Capacity of U Fuel if Container 2 has U"

prob += S['3'] <= 1100 * ys['3'], "Maximum Capacity of S Fuel if Container 3 has S"
prob += R['3'] <= 1100 * yr['3'], "Maximum Capacity of R Fuel if Container 3 has R"
prob += U['3'] <= 1100 * yu['3'], "Maximum Capacity of U Fuel if Container 3 has U"

prob += S['4'] <= 1800 * ys['4'], "Maximum Capacity of S Fuel if Container 4 has S"
prob += R['4'] <= 1800 * yr['4'], "Maximum Capacity of R Fuel if Container 4 has R"
prob += U['4'] <= 1800 * yu['4'], "Maximum Capacity of U Fuel if Container 4 has U"

prob += S['5'] <= 3400 * ys['5'], "Maximum Capacity of S Fuel if Container 5 has S"
prob += R['5'] <= 3400 * yr['5'], "Maximum Capacity of R Fuel if Container 5 has R"
prob += U['5'] <= 3400 * yu['5'], "Maximum Capacity of U Fuel if Container 5 has U"


prob += sum([S[i] for i in compartments]) >= 2400, "Maximum Shortfall (500) Constraint for S Fuel"
prob += sum([R[i] for i in compartments]) >= 3500, "Maximum Shortfall (500) Constraint for R Fuel"
prob += sum([U[i] for i in compartments]) >= 4400, "Maximum Shortfall (500) Constraint for U Fuel"

prob += sum([S[i] for i in compartments]) <= 2900, "Demand Constraint for S Fuel"
prob += sum([R[i] for i in compartments]) <= 4000, "Demand Constraint for R Fuel"
prob += sum([U[i] for i in compartments]) <= 4900, "Demand Constraint for U Fuel"

#print(prob)

prob.writeLP("fuelMin.lp")

status = prob.solve(GLPK())
#print(status)
#print the result
for i in compartments:
    print(' Container {} :: Fuel S {} ::'.format(i, S[i].value()))
    print(' Container {} :: Fuel R {} ::'.format(i, R[i].value()))
    print(' Container {} :: Fuel U {} ::'.format(i, U[i].value()))

print("Objective", value(prob.objective))


'''
There is also an alternate optimal solution to this problem:

 Container 1 :: Fuel S 0.0 ::

 Container 1 :: Fuel R 2700.0 ::

 Container 1 :: Fuel U 0.0 ::

 Container 2 :: Fuel S 2800.0 ::

 Container 2 :: Fuel R 0.0 ::

 Container 2 :: Fuel U 0.0 ::

 Container 3 :: Fuel S 0.0 ::

 Container 3 :: Fuel R 1100.0 ::

 Container 3 :: Fuel U 0.0 ::

 Container 4 :: Fuel S 0.0 ::

 Container 4 :: Fuel R 0.0 ::

 Container 4 :: Fuel U 1800.0 ::
```

```
Container 5 :: Fuel S 0.0 ::

Container 5 :: Fuel R 0.0 ::

Container 5 :: Fuel U 3100.0 ::

Objective 2600.0


**Un fulfilled Demand**:
- Fuel S: 100
- Fuel R: 200
- Fuel U: 0
```

2 - b

Constraints

- Demand_Constraint_for_R_Fuel: R_1 + R_2 + R_3 + R_4 + R_5 <= 4000
- Demand_Constraint_for_S_Fuel: S_1 + S_2 + S_3 + S_4 + S_5 <= 2900
- Demand_Constraint_for_U_Fuel: U_1 + U_2 + U_3 + U_4 + U_5 <= 4900
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_1: Yr__1 + Ys__1 + Yu__1 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_2: Yr__2 + Ys__2 + Yu__2 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_3: Yr__3 + Ys__3 + Yu__3 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_4: Yr__4 + Ys__4 + Yu__4 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_5: Yr__5 + Ys__5 + Yu__5 <= 1
- Maximum_Capacity_of_R_Fuel_if_Container_1_has_R: R_1 - 2700 Yr__1 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_2_has_R: R_2 - 2800 Yr__2 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_3_has_R: R_3 - 1100 Yr__3 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_4_has_R: R_4 - 1800 Yr__4 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_5_has_R: R_5 - 3400 Yr__5 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_1_has_S: S_1 - 2700 Ys__1 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_2_has_S: S_2 - 2800 Ys__2 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_3_has_S: S_3 - 1100 Ys__3 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_4_has_S: S_4 - 1800 Ys__4 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_5_has_S: S_5 - 3400 Ys__5 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_1_has_U: U_1 - 2700 Yu__1 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_2_has_U: U_2 - 2800 Yu__2 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_3_has_U: U_3 - 1100 Yu__3 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_4_has_U: U_4 - 1800 Yu__4 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_5_has_U: U_5 - 3400 Yu__5 <= 0
- Maximum_Shortfall_(500)*Constraint*for_R_Fuel: R_1 + R_2 + R_3 + R_4 + R_5 >= 3500
- Maximum_Shortfall_(500)*Constraint*for_S_Fuel: S_1 + S_2 + S_3 + S_4 + S_5 >= 2400
- Maximum_Shortfall_(500)*Constraint*for_U_Fuel: U_1 + U_2 + U_3 + U_4 + U_5 >= 4400
- Binaries Yr__1 Yr__2 Yr__3 Yr__4 Yr__5 Ys__1 Ys__2 Ys__3 Ys__4 Ys__5 Yu__1 Yu__2 Yu__3 Yu__4 Yu__5
- Non Zero: S1, S2, S3, S4, S5, R1, R2, R3, R4, R5, U1,U2, U3, U4, U5 >= 0

2 - c

Objective

Minimize Objective: - 8 R_1 - 8 R_2 - 8 R_3 - 8 R_4 - 8 R_5 - 10 S_1 - 10 S_2 - 10 S_3 - 10 S_4 - 10 S_5 - 6 U_1 - 6 U_2 - 6 U_3 - 6 U_4 - 6 U_5

1. 8 = Penalty on not fulfilling R typer Fuel / Litre
2. 10 = Penalty on not fulfilling S typer Fuel / Litre
3. 6 = Penalty on not fulfilling U typer Fuel / Litre
4. Decision Variables : Fuel of Type S/R/U in Container 1/2/3/4/5: S1, S2, S3, S4, S5, R1, R2, R3, R4, R5, U1,U2, U3, U4, U5 >= 0

2 - d

In order to incorporate the new Penalty Structure, I will modify the objective and the constarints in the following manner:

New Decision Variables:

- D1_0 = Non Zero value means S1 deficit is more than 250 by D1_0 amount. D1_0 will zero if Total deficit is less 250. e.g. If D1_0 = 10, defict = 250+10 =260
- D1_1 = Non Zero value means S1 deficit is less than 250 by D1_1 amount. D1_1 will zero if Total deficit is more 250. e.g. If D1_1 = 10, defict = 250-10 =240
- D2_0 = Same logic for quantity > 250 for R
- D2_1 = Same logic for quantity < 250 for R
- D3_0 = Same logic for quantity > 250 for U
- D3_1 = Same logic for quantity < 250 for U

New Objective

- Minimize Objective: 10* (250 - d1[1]) + 11 * d1[0]+ 8* (250 - d2[1]) + 8.8 * d2[0] + 6* (250 - d3[1]) + 6.6 * d3[0]

Logic:

- If quantity > 250: # assume 260 for S1
- D1_1 = 10, D1_0 = 0, Penalty = 10 * (250-D1_0) + 11 * D1_1

Logic:

- If quantity < 250: # assume 240 for S1
- D1_1 = 0, D1_0 = 10, Penalty = 10 * (250-D1_0) + 11 * D1_1

Subject To

- New Constraints
  - *C1: S*1 + S_2 + S_3 + S_4 + S_5 + 250 + d1_0 - d1_1 <= 2900
  - *C2: R*1 + R_2 + R_3 + R_4 + R_5 + 250 + d2_0 - d2_1 <= 4000
  - *C3: U*1 + U_2 + U_3 + U_4 + U_5 + 250 + d3_0 - d3_1 <= 4900

Earlier Constraints

- Demand_Constraint_for_R_Fuel: R_1 + R_2 + R_3 + R_4 + R_5 == 4000
- Demand_Constraint_for_S_Fuel: S_1 + S_2 + S_3 + S_4 + S_5 == 2900
- Demand_Constraint_for_U_Fuel: U_1 + U_2 + U_3 + U_4 + U_5 == 4900
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_1: Yr__1 + Ys__1 + Yu__1 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_2: Yr__2 + Ys__2 + Yu__2 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_3: Yr__3 + Ys__3 + Yu__3 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_4: Yr__4 + Ys__4 + Yu__4 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_5: Yr__5 + Ys__5 + Yu__5 <= 1
- Maximum_Capacity_of_R_Fuel_if_Container_1_has_R: R_1 - 2700 Yr__1 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_2_has_R: R_2 - 2800 Yr__2 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_3_has_R: R_3 - 1100 Yr__3 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_4_has_R: R_4 - 1800 Yr__4 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_5_has_R: R_5 - 3400 Yr__5 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_1_has_S: S_1 - 2700 Ys__1 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_2_has_S: S_2 - 2800 Ys__2 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_3_has_S: S_3 - 1100 Ys__3 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_4_has_S: S_4 - 1800 Ys__4 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_5_has_S: S_5 - 3400 Ys__5 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_1_has_U: U_1 - 2700 Yu__1 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_2_has_U: U_2 - 2800 Yu__2 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_3_has_U: U_3 - 1100 Yu__3 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_4_has_U: U_4 - 1800 Yu__4 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_5_has_U: U_5 - 3400 Yu__5 <= 0
- Maximum_Shortfall_(500)*Constraint*for_R_Fuel: R_1 + R_2 + R_3 + R_4 + R_5 >= 3500
- Maximum_Shortfall_(500)*Constraint*for_S_Fuel: S_1 + S_2 + S_3 + S_4 + S_5 >= 2400
- Maximum_Shortfall_(500)*Constraint*for_U_Fuel: U_1 + U_2 + U_3 + U_4 + U_5 >= 4400
- Binaries: Yr__1 Yr__2 Yr__3 Yr__4 Yr__5 Ys__1 Ys__2 Ys__3 Ys__4 Ys__5 Yu__1 Yu__2 Yu__3 Yu__4 Yu__5 End

In [ ]:
```python
# initialize the model
prob = LpProblem("fuelMin", LpMinimize)
#List of decision variables
compartments = ['1', '2', '3', '4', '5']
# create a dictionary of pulp variables with keys from ingredients
S = pulp.LpVariable.dict('S_%s', compartments, lowBound = 0)
R = pulp.LpVariable.dict('R_%s', compartments, lowBound = 0)
U = pulp.LpVariable.dict('U_%s', compartments, lowBound = 0)

ys = LpVariable.dicts("Ys_", compartments, 0, None, cat = LpBinary)
yr = LpVariable.dicts("Yr_", compartments, 0, None, cat = LpBinary)
yu = LpVariable.dicts("Yu_", compartments, 0, None, cat = LpBinary)

d1 = LpVariable.dicts("d1", range(0, 2), 0, None)
d2 = LpVariable.dicts("d2", range(0, 2), 0, None)
d3 = LpVariable.dicts("d3", range(0, 2), 0, None)

# Objective function

# change
prob += 10* (250 - d1[1]) + 11 * d1[0]+\
        8* (250 - d2[1]) + 8.8 * d2[0] +\
        6*  (250 - d3[1]) + 6.6 * d3[0], "Objective"

# Constraints
prob += 250 + d1[0] - d1[1] == (2900 - sum([S[i] for i in compartments])) # change
prob += 250 + d2[0] - d2[1] == (4000 - sum([R[i] for i in compartments])) # change
prob += 250 + d3[0] - d3[1] == (4900 - sum([U[i] for i in compartments])) # change


prob += ys['1'] + yr['1'] + yu['1'] <= 1, "Integer Constraint for One Type of Fuel in Container 1"
prob += ys['2'] + yr['2'] + yu['2'] <= 1, "Integer Constraint for One Type of Fuel in Container 2"
prob += ys['3'] + yr['3'] + yu['3'] <= 1, "Integer Constraint for One Type of Fuel in Container 3"
prob += ys['4'] + yr['4'] + yu['4'] <= 1, "Integer Constraint for One Type of Fuel in Container 4"
prob += ys['5'] + yr['5'] + yu['5'] <= 1, "Integer Constraint for One Type of Fuel in Container 5"

prob += S['1'] <= 2700 * ys['1'], "Maximum Capacity of S Fuel if Container 1 has S"
prob += R['1'] <= 2700 * yr['1'], "Maximum Capacity of R Fuel if Container 1 has R"
prob += U['1'] <= 2700 * yu['1'], "Maximum Capacity of U Fuel if Container 1 has U"

prob += S['2'] <= 2800 * ys['2'], "Maximum Capacity of S Fuel if Container 2 has S"
prob += R['2'] <= 2800 * yr['2'], "Maximum Capacity of R Fuel if Container 2 has R"
prob += U['2'] <= 2800 * yu['2'], "Maximum Capacity of U Fuel if Container 2 has U"

prob += S['3'] <= 1100 * ys['3'], "Maximum Capacity of S Fuel if Container 3 has S"
prob += R['3'] <= 1100 * yr['3'], "Maximum Capacity of R Fuel if Container 3 has R"
prob += U['3'] <= 1100 * yu['3'], "Maximum Capacity of U Fuel if Container 3 has U"

prob += S['4'] <= 1800 * ys['4'], "Maximum Capacity of S Fuel if Container 4 has S"
prob += R['4'] <= 1800 * yr['4'], "Maximum Capacity of R Fuel if Container 4 has R"
prob += U['4'] <= 1800 * yu['4'], "Maximum Capacity of U Fuel if Container 4 has U"

prob += S['5'] <= 3400 * ys['5'], "Maximum Capacity of S Fuel if Container 5 has S"
prob += R['5'] <= 3400 * yr['5'], "Maximum Capacity of R Fuel if Container 5 has R"
prob += U['5'] <= 3400 * yu['5'], "Maximum Capacity of U Fuel if Container 5 has U"


prob += sum([S[i] for i in compartments]) >= 2400, "Maximum Shortfall (500) Constraint for S Fuel"
prob += sum([R[i] for i in compartments]) >= 3500, "Maximum Shortfall (500) Constraint for R Fuel"
prob += sum([U[i] for i in compartments]) >= 4400, "Maximum Shortfall (500) Constraint for U Fuel"

prob += sum([S[i] for i in compartments]) <= 2900, "Demand Constraint for S Fuel"
prob += sum([R[i] for i in compartments]) <= 4000, "Demand Constraint for R Fuel"
prob += sum([U[i] for i in compartments]) <= 4900, "Demand Constraint for U Fuel"

#print(prob)

prob.writeLP("./fuelMin.lp")

status = prob.solve(GLPK())
#print(status)
#print the result
for i in compartments:
    print(' {} :: {} ::'.format(i, S[i].value()))
    print(' {} :: {} ::'.format(i, R[i].value()))
    print(' {} :: {} ::'.format(i, U[i].value()))
for i in range(0,2):
    print(' {} :: {} ::'.format(i, d1[i].value()))
    print(' {} :: {} ::'.format(i, d2[i].value()))
    print(' {} :: {} ::'.format(i, d3[i].value()))

print("Objective", value(prob.objective))
```

Q2 - e

New Objective:

Minimize Objective: - 10* (250 - d1[1]) + 11 * (250 + d1[0]) - 10 * 250 * (1 - y1[0]) - 11 * 250 * (y1[0]) +\ 8* (250 - d2[1]) + 8.8 * (250 + d2[0]) - 8 * 250 * (1 - y1[1]) - 8.8 * 250 * (y1[1]) +\ 6* (250 - d3[1]) + 6.6 * (250 + d3[0]) - 6 * 250 * (1 - y1[2]) - 6.6 * 250 * (y1[2])

Logic:

- If S deficit = 240,
    - d1_1 = 10, Y[1] = 1, hence Loss = 10 *240 + 11 * 250 - 0 - 11 * 250 ...
- If S deficit = 260,

- d1_0 = 10, Y[1] = 0, hence Loss = 10 *250 + 11 * 260 - 10 * 250 - 8 ...

The constraints to factor the above in added below:

Subject To

New Constraints

- *C10: - 250 Y1__2 + d3*1 <= 0
- *C11: d3*0 + d3_1 <= 250
- *C12: 250 Y1__2 + d3*0 <= 250
- *C4: - 250 Y1__0 + d1*1 <= 0
- *C5: d1*0 + d1_1 <= 250
- *C6: 250 Y1__0 + d1*0 <= 250
- *C7: - 250 Y1__1 + d2*1 <= 0
- *C8: d2*0 + d2_1 <= 250
- *C9: 250 Y1__1 + d2*0 <= 250

Older COnstraints

- *C1: S*1 + S_2 + S_3 + S_4 + S_5 + 250 + d1_0 - d1_1 <= 2900
- *C2: R*1 + R_2 + R_3 + R_4 + R_5 + 250 + d2_0 - d2_1 <= 4000
- *C3: U*1 + U_2 + U_3 + U_4 + U_5 + 250 + d3_0 - d3_1 <= 4900
- Demand_Constraint_for_R_Fuel: R_1 + R_2 + R_3 + R_4 + R_5 <= 4000
- Demand_Constraint_for_S_Fuel: S_1 + S_2 + S_3 + S_4 + S_5 <= 2900
- Demand_Constraint_for_U_Fuel: U_1 + U_2 + U_3 + U_4 + U_5 <= 4900
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_1: Yr__1 + Ys__1 + Yu__1 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_2: Yr__2 + Ys__2 + Yu__2 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_3: Yr__3 + Ys__3 + Yu__3 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_4: Yr__4 + Ys__4 + Yu__4 <= 1
- Integer_Constraint_for_One_Type_of_Fuel_in_Container_5: Yr__5 + Ys__5 + Yu__5 <= 1
- Maximum_Capacity_of_R_Fuel_if_Container_1_has_R: R_1 - 2700 Yr__1 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_2_has_R: R_2 - 2800 Yr__2 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_3_has_R: R_3 - 1100 Yr__3 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_4_has_R: R_4 - 1800 Yr__4 <= 0
- Maximum_Capacity_of_R_Fuel_if_Container_5_has_R: R_5 - 3400 Yr__5 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_1_has_S: S_1 - 2700 Ys__1 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_2_has_S: S_2 - 2800 Ys__2 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_3_has_S: S_3 - 1100 Ys__3 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_4_has_S: S_4 - 1800 Ys__4 <= 0
- Maximum_Capacity_of_S_Fuel_if_Container_5_has_S: S_5 - 3400 Ys__5 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_1_has_U: U_1 - 2700 Yu__1 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_2_has_U: U_2 - 2800 Yu__2 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_3_has_U: U_3 - 1100 Yu__3 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_4_has_U: U_4 - 1800 Yu__4 <= 0
- Maximum_Capacity_of_U_Fuel_if_Container_5_has_U: U_5 - 3400 Yu__5 <= 0
- Maximum_Shortfall_(500)*Constraint*for_R_Fuel: R_1 + R_2 + R_3 + R_4 + R_5 >= 3500
- Maximum_Shortfall_(500)*Constraint*for_S_Fuel: S_1 + S_2 + S_3 + S_4 + S_5 >= 2400
- Maximum_Shortfall_(500)*Constraint*for_U_Fuel: U_1 + U_2 + U_3 + U_4 + U_5 >= 4400
- Binaries: Yr__1 Yr__2 Yr__3 Yr__4 Yr__5 Ys__1 Ys__2 Ys__3 Ys__4 Ys__5 Yu__1 Yu__2 Yu__3 Yu__4 Yu__5 y1_0 y1_1 y2_0 y2_1 y3_0 y3_1 End

In [ ]:
```python
# initialize the model
prob = LpProblem("fuelMin", LpMinimize)
#List of decision variables
compartments = ['1', '2', '3', '4', '5']
# create a dictionary of pulp variables with keys from ingredients
S = pulp.LpVariable.dict('S_%s', compartments, lowBound = 0)
R = pulp.LpVariable.dict('R_%s', compartments, lowBound = 0)
U = pulp.LpVariable.dict('U_%s', compartments, lowBound = 0)

ys = LpVariable.dicts("Ys_", compartments, 0, None, cat = LpBinary)
yr = LpVariable.dicts("Yr_", compartments, 0, None, cat = LpBinary)
yu = LpVariable.dicts("Yu_", compartments, 0, None, cat = LpBinary)

d1 = LpVariable.dicts("d1", range(0, 2), 0, None)
d2 = LpVariable.dicts("d2", range(0, 2), 0, None)
d3 = LpVariable.dicts("d3", range(0, 2), 0, None)

y1 = LpVariable.dicts("Y1_", range(0, 3), 0, None, cat = LpBinary)
#y2 = LpVariable.dicts("Y2_", range(0, 3), 0, None, cat = LpBinary)


# Objective function

# change
prob += 10* (250 - d1[1]) + 11 * (250 + d1[0]) - 10 * 250 * (1 - y1[0]) - 11 * 250 * (y1[0]) +\
        8* (250 - d2[1]) + 8.8 * (250 + d2[0]) - 8 * 250 * (1 - y1[1]) - 8.8 * 250 * (y1[1]) +\
        6*  (250 - d3[1]) + 6.6 * (250 + d3[0]) - 6 * 250 * (1 - y1[2]) - 6.6 * 250 * (y1[2]) , "Objective"

# Constraints
prob += 250 + d1[0] - d1[1] >= (2900 - sum([S[i] for i in compartments])) # change
prob += 250 + d2[0] - d2[1] >= (4000 - sum([R[i] for i in compartments])) # change
prob += 250 + d3[0] - d3[1] >= (4900 - sum([U[i] for i in compartments])) # change

prob +=  d1[1] <= 250 * y1[0]
prob +=  d1[1] + d1[0] <= 250
prob +=  d1[0] <= 250*(1-y1[0])

prob +=  d2[1] <= 250 * y1[1]
prob +=  d2[1] + d2[0] <= 250
prob +=  d2[0] <= 250*(1-y1[1])

prob +=  d3[1] <= 250 * y1[2]
prob +=  d3[1] + d3[0] <= 250
prob +=  d3[0] <= 250*(1-y1[2])


prob += ys['1'] + yr['1'] + yu['1'] <= 1, "Integer Constraint for One Type of Fuel in Container 1"
prob += ys['2'] + yr['2'] + yu['2'] <= 1, "Integer Constraint for One Type of Fuel in Container 2"
prob += ys['3'] + yr['3'] + yu['3'] <= 1, "Integer Constraint for One Type of Fuel in Container 3"
prob += ys['4'] + yr['4'] + yu['4'] <= 1, "Integer Constraint for One Type of Fuel in Container 4"
prob += ys['5'] + yr['5'] + yu['5'] <= 1, "Integer Constraint for One Type of Fuel in Container 5"

prob += S['1'] <= 2700 * ys['1'], "Maximum Capacity of S Fuel if Container 1 has S"
prob += R['1'] <= 2700 * yr['1'], "Maximum Capacity of R Fuel if Container 1 has R"
prob += U['1'] <= 2700 * yu['1'], "Maximum Capacity of U Fuel if Container 1 has U"

prob += S['2'] <= 2800 * ys['2'], "Maximum Capacity of S Fuel if Container 2 has S"
prob += R['2'] <= 2800 * yr['2'], "Maximum Capacity of R Fuel if Container 2 has R"
prob += U['2'] <= 2800 * yu['2'], "Maximum Capacity of U Fuel if Container 2 has U"

prob += S['3'] <= 1100 * ys['3'], "Maximum Capacity of S Fuel if Container 3 has S"
prob += R['3'] <= 1100 * yr['3'], "Maximum Capacity of R Fuel if Container 3 has R"
prob += U['3'] <= 1100 * yu['3'], "Maximum Capacity of U Fuel if Container 3 has U"

prob += S['4'] <= 1800 * ys['4'], "Maximum Capacity of S Fuel if Container 4 has S"
prob += R['4'] <= 1800 * yr['4'], "Maximum Capacity of R Fuel if Container 4 has R"
prob += U['4'] <= 1800 * yu['4'], "Maximum Capacity of U Fuel if Container 4 has U"

prob += S['5'] <= 3400 * ys['5'], "Maximum Capacity of S Fuel if Container 5 has S"
prob += R['5'] <= 3400 * yr['5'], "Maximum Capacity of R Fuel if Container 5 has R"
prob += U['5'] <= 3400 * yu['5'], "Maximum Capacity of U Fuel if Container 5 has U"


prob += sum([S[i] for i in compartments]) >= 2400, "Maximum Shortfall (500) Constraint for S Fuel"
prob += sum([R[i] for i in compartments]) >= 3500, "Maximum Shortfall (500) Constraint for R Fuel"
prob += sum([U[i] for i in compartments]) >= 4400, "Maximum Shortfall (500) Constraint for U Fuel"

prob += sum([S[i] for i in compartments]) <= 2900, "Demand Constraint for S Fuel"
prob += sum([R[i] for i in compartments]) <= 4000, "Demand Constraint for R Fuel"
prob += sum([U[i] for i in compartments]) <= 4900, "Demand Constraint for U Fuel"

#print(prob)

prob.writeLP("./fuelMin.lp")

status = prob.solve(GLPK())
#print(status)
#print the result
for i in compartments:
    print(' {} :: {} ::'.format(i, S[i].value()))
    print(' {} :: {} ::'.format(i, R[i].value()))
    print(' {} :: {} ::'.format(i, U[i].value()))

for i in range(0,2):
    print(' d1{} :: {} ::'.format(i, d1[i].value()))
    print(' d2{} :: {} ::'.format(i, d2[i].value()))
    print(' d3{} :: {} ::'.format(i, d3[i].value()))
```

```
for i in range(0,3):
    print(' {} :: {} ::'.format(i, y1[i].value()))

print("Objective", value(prob.objective))
```

Q3 - 1

- Optimal Production PLan
  - Alloy 1 - 0
  - Alloy 2 - 1500
  - Alloy 3 - 500
  - Alloy 4 - 1000

These can be derived from the Constraints Section of the Sensitivity Report

In [164]:
```
print("Objective Function Value = {}".format(1500*111 + 281* 500 + 188* 1000))
```

```
Objective Function Value = 495000
```

Q3 - 2 - Check / Object will definitely increase

As per the sesitivity Report Maximum Allowable increase in Metal 3 is 100 GM. Hence if 200 GM of metal is imported then this exceeds allowable increase for the Optimal Solution. This will change the current Optimal Solution, unfortunately the sensitivity report does not tell how it will change.

From Solving the dual solution we see that the Shadow Price for Y3 is 555 (Please see ans to 3-4, details are provided). Hence even if we procure 100 Gm of Metal 3, cost incurred is 20000, while for the 100 GM we know a additional revnue of 55500 can be made.

If we have to procude all 200GM they can procure even if they decide not to use the additional 100 GM as it will change the optimal solution.

Hence we can procure the amount.

Objective increases by 35000 to 530000

Q3 - 3

- From the sensitivity report we can see that the Demand for Alloy 3 can be increased by 750 units without changing the Optimal Solution. So if the demand increases by 200, it will not change the optimal solution.
- If we solve the Dual Formulation, the Dual Price for Y6 (Corresponding Variable for the Demand Constraint for Alloy 3) is 59 (See 3-4 for Details below). Hence they will add 5 * 200 = 11,800. Overall Objective = 506800
- However, we know the profit per unit of Alloy 3 is 281. Ideally as per my understanding the Dual Price should have been same as the Profit (Objective Value), may be the numbers have been changed from the actual formuation. I will consider the increase in objective to be equal to the increase due to profit from selling the additional 200 units of Alloy 3. Hence the Objective will increase by 281 * 200 = 56200,

Final Objective = 551200

Q 3 - 4

- The corresponding Dual formulations are:

0.2y1 + 0.4y2 + 0.4y3 + y4 + 0 + 0 + 0 >= 186

0.2y1 + 0.6y2 + 0.2y3 + 0 + y5 + 0 + 0 >= 111

0.3y1 + 0.3y2 + 0.4y3 + 0 + 0 + y6 + 0 >= 281

0.5y1 + 0.5y2 + 0 + 0 + 0 + 0 + y7 >= 188

- Constraints 3, 6, 7 are binding in Primal, hence the corresponding Dual Variables are non-Zero. Hence y3, y6 and y7 are non zero.

Solving above equations we get: y3 = 555 y6 = 59 y7 = 188

Substituting the value of Y3 in equation 1 we get the Surplus = 36. This is the Reduced Cost for x1. hence the profit needs to be increased by 36 (Profit = 222) for Sedon to accept the order

Q 3 - 5

- As we can clearly see that the increase in the profit values for all are well outside the range of the permissible values as per the sensitivity report. The Solution will no longer remain optimal with these revised objective values. We need to resolve to find new Optimal Solution with these Profits

Q 3 - 6

- From sensitivity report we see that maximum Allowed decrease of Alloy 2 is 500 (Actual Value = 1000), which does not alter the optimal solution.
- The government restriction which reduces sales of alloy 2 value to 1000, hence does not impact the optimal Solution

Q 3 - 7

- Maximum Permissible change (increase) for Alloy 3 = 500. Actual change = 500. Hence percentage increase is 100%
- Maximum Permissible change (decrease) for Alloy 2 = 500. Actual change = 500. Hence percentage increase is 100%

Since the simultatneous change is more than 100%, hence this violates the 100% Rule for change. The solution may no longer remain optimal. We need to resolve the problem before making any conclusions.

Q 3 - 8

- Alloy 5 is introduced

The new sets of Primal equations are:

- Objective:

186x1 + 111x2 + 281x3 + 188x4 + 220x5

- Constraints:

0.2x1 +0.2x2 + 0.3x3 + 0.5x4 + 0.5x5 <= 2000

0.4x1 +0.6x2 + 0.3x3 + 0.5x4 + 0.4x5 <= 3000

0.2x1 +0.2x2 + 0.4x3 + 0x4 + 0.1x5 <= 500

x1 <= 1000

x2 <= 2000

x3 <= 500

x4 <= 1000

x5 <= 1500

Let us assume that x5 = 0, then the formulation remains optimal if the following dual constraint is feasible and non-binding:

0.5y1 + 0.4y2 + 0.1y3 + 0 + 0 + 0 + 0 + 0 >= 220

We know : y1, y2 = 0

y3 = 555,

hence 55.5 >= 220, which is infeasible. Hence x5 != 0, and with new alloy the old solution is no longer optimal

Q4

Decision Variables

- hiredEmpl_i, i = 1,2,3,4. Employees to be hired for Months September, October, January, February (Integers)
- transfrEmpl_i, i= 1..6, Employees to be trasferred from other location all 6 months (Integers)
- Binary Variables: isReqHire_1 - (Hiring in March), isReqHire_2 - (Hiring in December)
- analyst_1 = 79, as of First Day of September

Minimize Objective: 36000 analyst_1 + 34800 hiredEmpl_1 + 29100 hiredEmpl_2 + 12000 hiredEmpl_3 + 6000 hiredEmpl_4 + 20000 isReqHire_1 + 20000 isReqHire_2 + 8000 transfrEmpl_1 + 8000 transfrEmpl_2 + 8000 transfrEmpl_3 + 8000 transfrEmpl_4 + 8000 transfrEmpl_5 + 8000 transfrEmpl_6

Subject To

- Demand_Dec: analyst_1 + 0.95 hiredEmpl_1 + 0.95 hiredEmpl_2 + 0.8 transfrEmpl_4 >= 65
- Demand_Feb: analyst_1 + 0.95 hiredEmpl_1 + 0.95 hiredEmpl_2 + hiredEmpl_3 + hiredEmpl_4 + 0.8 transfrEmpl_6 >= 90
- Demand_Jan: analyst_1 + 0.95 hiredEmpl_1 + 0.95 hiredEmpl_2 + hiredEmpl_3 + 0.8 transfrEmpl_5 >= 80
- Demand_Nov: analyst_1 + 0.95 hiredEmpl_1 + hiredEmpl_2 + 0.8 transfrEmpl_3 >= 90
- Demand_Oct: analyst_1 + hiredEmpl_1 + hiredEmpl_2 + 0.8 transfrEmpl_2 >= 105
- Demand_Sept: analyst_1 + hiredEmpl_1 + 0.8 transfrEmpl_1 >= 110
- Initial_Number_of_confirmed_Analyst_as_of_Sept01: analyst_1 = 79
- Transferred_Employee_20%*Constraint: 1.2 analyst*1 + 1.16 hiredEmpl_1 + 0.97 hiredEmpl_2 + 0.4 hiredEmpl_3 + 0.2 hiredEmpl_4 - transfrEmpl_1 - transfrEmpl_2 - transfrEmpl_3 - transfrEmpl_4 - transfrEmpl_5 - transfrEmpl_6 >= 0
- Hired Employees per month *C1: hiredEmpl*1 - 50 isReqHire_1 <= 0
- *C2: hiredEmpl*2 - 50 isReqHire_1 <= 0
- *C3: hiredEmpl*3 - 50 isReqHire_2 <= 0
- *C4: hiredEmpl*4 - 50 isReqHire_2 <= 0
- 0 <= hiredEmpl_1
- 0 <= hiredEmpl_2
- 0 <= hiredEmpl_3
- 0 <= hiredEmpl_4
- 0 <= transfrEmpl_1
- 0 <= transfrEmpl_2
- 0 <= transfrEmpl_3
- 0 <= transfrEmpl_4
- 0 <= transfrEmpl_5
- 0 <= transfrEmpl_6

End

In [143]:
```
# initialize the model
prob = LpProblem("hireMin", LpMinimize)
#List of decision variables
demand = {
    1: 110,
    2: 105,
    3: 90,
    4: 65.,
    5: 80,
    6: 90
}

T = len(demand)
hierCost = 20000
analystCost = 6000
transferAnalystCost = 8000

# create a dictionary of pulp variables with keys from ingredients
transfrEmpl = LpVariable.dicts("transfrEmpl", range(1, T+1), 0, None, cat = LpInteger)
isReqHire =  LpVariable.dicts("isReqHire", range(1, 3), 0, None, cat = LpBinary)
analyst =  LpVariable.dicts("analyst", range(1, 2), 0, None)
hiredEmpl = LpVariable.dicts("hiredEmpl", range(1, 5), 0, None, cat = LpInteger)

# Objective function
prob += analyst[1] * 6000 + hiredEmpl[1] * 6000 + transfrEmpl[1] * 8000 +\
        analyst[1] * 6000 + hiredEmpl[1] * 6000 + hiredEmpl[2] * 6000 + transfrEmpl[2] * 8000 +\
        analyst[1] * 6000 +  0.95 * hiredEmpl[1] * 6000 + hiredEmpl[2] * 6000 + transfrEmpl[3] * 8000 +\
        analyst[1] * 6000 +  0.95 * hiredEmpl[1] * 6000 + 0.95 * hiredEmpl[2] * 6000 + transfrEmpl[4] * 8000 +\
        analyst[1] * 6000 +  0.95 * hiredEmpl[1] * 6000 + 0.95 * hiredEmpl[2] * 6000 + hiredEmpl[3] * 6000 + transfrEmpl[5]
        analyst[1] * 6000 +  0.95 * hiredEmpl[1] * 6000 + 0.95 * hiredEmpl[2] * 6000 + hiredEmpl[3] * 6000 + hiredEmpl[4] *
        isReqHire[1] * 20000 + isReqHire[2] * 20000, "Objective"


# Constraints
prob += hiredEmpl[1] <= 50 * isReqHire[1]
prob += hiredEmpl[2] <= 50 * isReqHire[1]
prob += hiredEmpl[3] <= 50 * isReqHire[2]
prob += hiredEmpl[4] <= 50 * isReqHire[2]

prob += analyst[1] == 79, "Initial Number of confirmed Analyst as of Sept01"
prob += analyst[1] + hiredEmpl[1] + transfrEmpl[1] * 0.8 >= demand[1], "Demand Sept"
prob += analyst[1] +  hiredEmpl[1] + hiredEmpl[2] + transfrEmpl[2] * 0.8 >= demand[2], "Demand Oct"
prob += analyst[1] +  0.95 * hiredEmpl[1] + hiredEmpl[2] + transfrEmpl[3] * 0.8  >= demand[3], "Demand Nov"
prob += analyst[1] +  0.95 * hiredEmpl[1] + 0.95 * hiredEmpl[2] + transfrEmpl[4] * 0.8 >= demand[4], "Demand Dec"
prob += analyst[1] +  0.95 * hiredEmpl[1] + 0.95 * hiredEmpl[2] + hiredEmpl[3] + transfrEmpl[5] * 0.8 >= demand[5], "Demand
prob += analyst[1] +  0.95 * hiredEmpl[1] + 0.95 * hiredEmpl[2] + hiredEmpl[3] + hiredEmpl[4] + transfrEmpl[6] * 0.8 >= dem

prob += 0.2 * (analyst[1] * 6 + hiredEmpl[1] + hiredEmpl[1] + hiredEmpl[2] +\
        hiredEmpl[1] * 0.95 + hiredEmpl[2] +\
        hiredEmpl[1] * 0.95 + hiredEmpl[2] * 0.95+\
        hiredEmpl[1] * 0.95 + hiredEmpl[2] * 0.95 + hiredEmpl[3] +\
        hiredEmpl[1] * 0.95 + hiredEmpl[2] * 0.95 + hiredEmpl[3] + hiredEmpl[4]) >= sum([transfrEmpl[i] for i in range(1,T+


#print(prob)

prob.writeLP("hireMin.lp")

status = prob.solve(GLPK())
#print(status)
#print the result
for i in range(1,7):
    print('Transferred Employee by Month {} :: {} ::'.format(i, transfrEmpl[i].value()))

for i in range(1,3):
    print('Is Hiring required Period {} :: {} ::'.format(i, isReqHire[i].value()))

for i in range(1,5):
    print('Nu mber of Hired Employees by Month{} :: {} ::'.format(i, hiredEmpl[i].value()))

print('Number of Analyst on Month 1 (September) {} :: {} ::'.format(1, analyst[1].value()))


print("Objective", value(prob.objective))
```

```
Transferred Employee by Month 1 :: 25 ::
Transferred Employee by Month 2 :: 19 ::
Transferred Employee by Month 3 :: 1 ::
Transferred Employee by Month 4 :: 0 ::
Transferred Employee by Month 5 :: 0 ::
Transferred Employee by Month 6 :: 1 ::
Is Hiring required Period 1 :: 1 ::
Is Hiring required Period 2 :: 0 ::
Nu mber of Hired Employees by Month1 :: 11 ::
Nu mber of Hired Employees by Month2 :: 0 ::
Nu mber of Hired Employees by Month3 :: 0 ::
Nu mber of Hired Employees by Month4 :: 0 ::
Number of Analyst on Month 1 (September) 1 :: 79.0 ::
Objective 3614800.0
```

Q5

- Problem is not solving if we have to match monthly demand. For Month 5 Demand == 2000, this cannot be achieved (Max that can be achived is 1600 for month 5)
- LP will solve if monthly constraints does not have to be met. Hence we will design a solution to meet overall demand of 7000 Stones

Decision variables (DV)

- Q1_Stone_i, i = 1..6
- Q2_Stone_i, i = 1..6
- carryFwd_i, i = 1..6 Carry for each period (Extra Stones)
- unMetDemand_i, i = 1..6 Unmet Demand for each period (Less stones Stones to be met in subsequent years)

Minimize Objective: 1000 carryFwd_0 + 1000 carryFwd_1 + 1000 carryFwd_2 + 1000 carryFwd_3 + 1000 carryFwd_4 + 1000 carryFwd_5 + 1000 carryFwd_6 + 215000 q1Stones_1 + 215000 q1Stones_2 + 215000 q1Stones_3 + 207500 q1Stones_4 + 207500 q1Stones_5 + 207500 q1Stones_6 + 240000 q2Stones_1 + 240000 q2Stones_2 + 240000 q2Stones_3 + 232500 q2Stones_4 + 232500 q2Stones_5 + 232500 q2Stones_6

Subject To

- All DV >0
- Stopped_Production_1: q2Stones_4 = 0
- Stopped_Production_2: q2Stones_5 = 0
- Overall Demand_Constraints_6: q1Stones_1 + q1Stones_2 + q1Stones_3 + q1Stones_4 + q1Stones_5 + q1Stones_6 + q2Stones_1 + q2Stones_2 + q2Stones_3 + q2Stones_4 + q2Stones_5 + q2Stones_6 >= 7000
- Production_Constraints_Q1_1: q1Stones_1 <= 800
- Production_Constraints_Q1_2: q1Stones_2 <= 800
- Production_Constraints_Q1_3: q1Stones_3 <= 800
- Production_Constraints_Q1_4: q1Stones_4 <= 800
- Production_Constraints_Q1_5: q1Stones_5 <= 800
- Production_Constraints_Q1_6: q1Stones_6 <= 800
- Production_Constraints_Q2_1: q2Stones_1 <= 1400
- Production_Constraints_Q2_2: q2Stones_2 <= 1400
- Production_Constraints_Q2_3: q2Stones_3 <= 1400
- Production_Constraints_Q2_4: q2Stones_4 <= 1400
- Production_Constraints_Q2_5: q2Stones_5 <= 1400
- Production_Constraints_Q2_6: q2Stones_6 <= 1400
- Stone_Storage_Limit_1: carryFwd_1 <= 1200
- Stone_Storage_Limit_2: carryFwd_2 <= 1200
- Stone_Storage_Limit_3: carryFwd_3 <= 1200
- Stone_Storage_Limit_4: carryFwd_4 <= 1200
- Stone_Storage_Limit_5: carryFwd_5 <= 1200
- Stone_Storage_Limit_6: carryFwd_6 <= 1200
- Demand_C1: - carryFwd_1 + q1Stones_1 + q2Stones_1 + unMetDemand_1 <= 700
- Demand_C2: carryFwd_1 - carryFwd_2 + q1Stones_2 + q2Stones_2 - unMetDemand_1 + unMetDemand_2 <= 700
- Demand_C3: carryFwd_2 - carryFwd_3 + q1Stones_3 + q2Stones_3 - unMetDemand_2 + unMetDemand_3 <= 1000
- Demand_C4: carryFwd_3 - carryFwd_4 + q1Stones_4 + q2Stones_4 - unMetDemand_3 + unMetDemand_4 <= 1200
- Demand_C5: carryFwd_4 - carryFwd_5 + q1Stones_5 + q2Stones_5 - unMetDemand_4 + unMetDemand_5 <= 2000
- Demand_C6: carryFwd_5 - carryFwd_6 + q1Stones_6 + q2Stones_6 - unMetDemand_5 + unMetDemand_6 <= 1400

End

In [149]:

```python
# initialize the model
prob = LpProblem("quarryMin", LpMinimize)
#List of decision variables
demand = {
    1: 700,
    2: 700,
    3: 1000,
    4: 1200,
    5: 2000,#2000
    6: 1400
}

T = len(demand)
q1Cost = 200000
q2Cost = 225000
milCost = 15000
farmCost = 7500
storageCost = 1000


# create a dictionary of pulp variables with keys from ingredients
q1Stones = LpVariable.dicts("q1Stones", range(1, T+1), 0, None)#, cat = LpInteger
q2Stones =  LpVariable.dicts("q2Stones", range(1, T+1), 0, None)#, cat = LpBinary
carryFwd =  LpVariable.dicts("carryFwd", range(0, T+1), 0, None)
unMetDemand =  LpVariable.dicts("unMetDemand", range(0, T+1), 0, None)

# Objective function
prob += sum([storageCost * carryFwd[i] for i in range(0, T+1)]) +\
        sum([q1Cost * q1Stones[i] for i in range(1, T+1)]) +\
        sum([q2Cost * q2Stones[i] for i in range(1, T+1)]) +\
        sum([milCost * (q1Stones[i] + q2Stones[i]) for i in range(1, 4)]) +\
        sum([farmCost * (q1Stones[i] + q2Stones[i]) for i in range(4, 7)]), "Objective"


# Constraints
prob += q2Stones[4] == 0, "Stopped Production 1"
prob += q2Stones[5] == 0, "Stopped Production 2"
prob += unMetDemand[0] == 0, "Un Met Demand"

for i in range(1, T+1):
    prob += carryFwd[i] <= 1200, "Stone Storage Limit " + str(i)
    prob += q1Stones[i] <= 800, "Production Constraints Q1 " + str(i)
    prob += q2Stones[i] <= 1400, "Production Constraints Q2 " + str(i)
prob += carryFwd[6] == 0, "No Carry at last period"
prob += carryFwd[0] == 0, "No Carry at first period"

for i in range(1, T+1):
    #prob += carryFwd[i] + demand[i] - unMetDemand[i]  + unMetDemand[i-1] <=  q1Stones[i] + q2Stones[i] #+ carryFwd[i - 1]
    #prob += carryFwd[i] >= carryFwd[i-1] + q1Stones[i] + q2Stones[i] - demand[i]
    prob += q1Stones[i] + q2Stones[i] + carryFwd[i-1] <= demand[i] + carryFwd[i] -unMetDemand[i] + unMetDemand[i-1]

#prob += q1Stones[1] + q2Stones[1] + carryFwd[0] <= demand[1] + carryFwd[1] -unMetDemand[1]
#prob += q1Stones[2] + q2Stones[2] + carryFwd[1] <= demand[2] + carryFwd[2] -unMetDemand[2] + unMetDemand[1]
#prob += q1Stones[3] + q2Stones[3] + carryFwd[2] <= demand[3] + carryFwd[3] -unMetDemand[3] + unMetDemand[2]
#prob += q1Stones[4] + q2Stones[4] + carryFwd[3] <= demand[4] + carryFwd[4] -unMetDemand[4] + unMetDemand[3]
#prob += q1Stones[5] + q2Stones[5] + carryFwd[4] <= demand[5] + carryFwd[5] -unMetDemand[5] + unMetDemand[4]
#prob += q1Stones[6] + q2Stones[6] + carryFwd[5] <= demand[6] + carryFwd[6] -unMetDemand[6] + unMetDemand[5]

prob += sum([q1Stones[i] for i in range(1,7)] + [q2Stones[i] for i in range(1,7)]) >= sum([demand[i] for i in range(1,7)]),
#print(prob)


prob.writeLP("quarryMin.lp")

status = prob.solve(GLPK(options=["--ranges","quarryMin.sen"]))

print(status)
#print the result
for i in range(0,T+1):
    print('Carry Forward :: Period ::  {} :: {} ::'.format(i, carryFwd[i].value()))

for i in range(1,T+1):
    print('Quarry1 :: Period :: {} :: {} ::'.format(i, q1Stones[i].value()))

for i in range(1,T+1):
    print('Quarry2 :: Period ::  {} :: {} ::'.format(i, q2Stones[i].value()))

for i in range(1,T+1):
    print('Unmet Demand :: Period ::  {} :: {} ::'.format(i, unMetDemand[i].value()))

print("Objective", value(prob.objective))
```

```
1
Carry Forward :: Period ::  0 :: 0.0 ::
Carry Forward :: Period ::  1 :: 100.0 ::
Carry Forward :: Period ::  2 :: 200.0 ::
Carry Forward :: Period ::  3 :: 800.0 ::
Carry Forward :: Period ::  4 :: 400.0 ::
Carry Forward :: Period ::  5 :: 0.0 ::
Carry Forward :: Period ::  6 :: 0.0 ::
Quarry1 :: Period :: 1 :: 800.0 ::
Quarry1 :: Period :: 2 :: 800.0 ::
Quarry1 :: Period :: 3 :: 800.0 ::
Quarry1 :: Period :: 4 :: 800.0 ::
Quarry1 :: Period :: 5 :: 800.0 ::
Quarry1 :: Period :: 6 :: 800.0 ::
Quarry2 :: Period ::  1 :: 0.0 ::
```

```
Quarry2 :: Period ::  2 :: 0.0 ::
Quarry2 :: Period ::  3 :: 800.0 ::
Quarry2 :: Period ::  4 :: 0.0 ::
Quarry2 :: Period ::  5 :: 0.0 ::
Quarry2 :: Period ::  6 :: 1400.0 ::
Unmet Demand :: Period ::  1 :: 0.0 ::
Unmet Demand :: Period ::  2 :: 0.0 ::
Unmet Demand :: Period ::  3 :: 0.0 ::
Unmet Demand :: Period ::  4 :: 0.0 ::
Unmet Demand :: Period ::  5 :: 800.0 ::
Unmet Demand :: Period ::  6 :: 0.0 ::
Objective 1533000000.0
```

In [ ]:

```
# %load quarryMin.sen
GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                         Page   1

Problem:
Objective:  Objective = 1533000000 (MINimum)

   No. Row name     St      Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 Demand_Constraints_6
                    NL    7000.00000          .       7000.00000     6600.00000 -242000.00000    1.4362e+09 carryFwd_4
                                      242000.00000          +Inf     7000.00000          +Inf     1.533e+09 _C5

     2 No_Carry_at_first_period
                    NS          .             .             .              .             -Inf     1.533e+09 carryFwd_0
                                        5000.00000          .              .             +Inf     1.533e+09 _C5

     3 No_Carry_at_last_period
                    NS          .             .             .              .             -Inf     1.533e+09 _C5
                                        1000.00000          .         800.00000          +Inf    1.5338e+09 unMetDemand_5

     4 Production_Constraints_Q1_1
                    NU     800.00000          .             -Inf      700.00000          -Inf    1.5353e+09 carryFwd_1
                                      -23000.00000     800.00000     1600.00000   23000.00000    1.5146e+09 q2Stones_3

     5 Production_Constraints_Q1_2
                    NU     800.00000          .             -Inf      600.00000          -Inf    1.5378e+09 carryFwd_2
                                      -24000.00000     800.00000     1600.00000   24000.00000    1.5138e+09 q2Stones_3

     6 Production_Constraints_Q1_3
                    NU     800.00000          .             -Inf      200.00000          -Inf     1.548e+09 Production_Cons
                                      -25000.00000     800.00000     1600.00000   25000.00000     1.513e+09 q2Stones_3

     7 Production_Constraints_Q1_4
                    NU     800.00000          .             -Inf      400.00000          -Inf    1.5464e+09 Stone_Storage_L
                                      -33500.00000     800.00000     1600.00000   33500.00000    1.5062e+09 carryFwd_3

     8 Production_Constraints_Q1_5
                    NU     800.00000          .             -Inf      400.00000          -Inf    1.5468e+09 Stone_Storage_L
                                      -34500.00000     800.00000     1200.00000   34500.00000    1.5192e+09 carryFwd_4

     9 Production_Constraints_Q1_6
                    NU     800.00000          .             -Inf      400.00000          -Inf    1.5468e+09 Stone_Storage_L
                                      -34500.00000     800.00000     1200.00000   34500.00000    1.5192e+09 carryFwd_4

    10 Production_Constraints_Q2_1
                    BS          .       1400.00000          -Inf      800.00000   -2000.00000     1.533e+09 q2Stones_1
                                              .       1400.00000           .             +Inf     1.533e+09

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                         Page   2

Problem:
Objective:  Objective = 1533000000 (MINimum)

   No. Row name     St      Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
    11 Production_Constraints_Q2_2
                    BS          .       1400.00000          -Inf      800.00000   -1000.00000     1.533e+09 q2Stones_2
                                              .       1400.00000           .             +Inf     1.533e+09

    12 Production_Constraints_Q2_3
                    BS     800.00000     600.00000          -Inf     1200.00000   -9500.00000    1.5254e+09 Production_Cons
                                              .       1400.00000           .       1000.00000    1.5338e+09 q2Stones_2

    13 Production_Constraints_Q2_4
                    BS          .       1400.00000          -Inf           .             -Inf     1.533e+09
                                              .       1400.00000           .             +Inf     1.533e+09

    14 Production_Constraints_Q2_5
                    BS          .       1400.00000          -Inf           .             -Inf     1.533e+09
                                              .       1400.00000           .             +Inf     1.533e+09

    15 Production_Constraints_Q2_6
                    NU    1400.00000          .             -Inf     1000.00000          -Inf    1.5368e+09 Stone_Storage_L
                                       -9500.00000    1400.00000     1800.00000    9500.00000    1.5292e+09 carryFwd_4

    16 Stone_Storage_Limit_1
                    BS     100.00000    1100.00000          -Inf          +Inf    -1000.00000    1.5329e+09 unMetDemand_1
                                              .       1200.00000           .      23000.00000    1.5353e+09 Production_Cons

    17 Stone_Storage_Limit_2
                    BS     200.00000    1000.00000          -Inf          +Inf    -1000.00000    1.5328e+09 unMetDemand_2
                                              .       1200.00000      100.00000   23000.00000    1.5376e+09 Production_Cons

    18 Stone_Storage_Limit_3
                    BS     800.00000     400.00000          -Inf          +Inf    -1000.00000    1.5322e+09 unMetDemand_3
                                              .       1200.00000      800.00000          +Inf          +Inf

    19 Stone_Storage_Limit_4
                    BS     400.00000     800.00000          -Inf      400.00000   -1000.00000    1.5326e+09 _C4
                                              .       1200.00000      400.00000          +Inf          +Inf

    20 Stone_Storage_Limit_5
                    BS          .       1200.00000          -Inf          +Inf    -1000.00000     1.533e+09 carryFwd_5
                                              .       1200.00000           .             +Inf     1.533e+09

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                         Page   3
```

```
Problem:
Objective:  Objective = 1533000000 (MINimum)

   No. Row name     St       Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
                                           Marginal   Upper bound          range         range  break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- -----------
    21 Stone_Storage_Limit_6
                    BS          .       1200.00000          -Inf          .             -Inf     1.533e+09
                                           .            1200.00000          .             +Inf     1.533e+09

    22 Stopped_Production_1
                    NS          .             .             .             .             -Inf     1.533e+09 q2Stones_4
                                        -8500.00000         .           800.00000        +Inf    1.5262e+09 carryFwd_3

    23 Stopped_Production_2
                    NS          .             .             .             .             -Inf     1.533e+09 q2Stones_5
                                        -9500.00000         .           400.00000        +Inf    1.5292e+09 carryFwd_4

    24 _C1          NU      700.00000         .             -Inf        700.00000        -Inf     1.533e+09 _C5
                                        -4000.00000     700.00000      800.00000    4000.00000   1.5326e+09 carryFwd_1

    25 _C2          NU      700.00000         .             -Inf        700.00000        -Inf     1.533e+09 _C5
                                        -3000.00000     700.00000      900.00000    3000.00000   1.5324e+09 carryFwd_2

    26 _C3          NU     1000.00000         .             -Inf       1000.00000        -Inf     1.533e+09 _C5
                                        -2000.00000    1000.00000     1400.00000    2000.00000   1.5322e+09 carryFwd_4

    27 _C4          NU     1200.00000         .             -Inf       1200.00000        -Inf     1.533e+09 _C5
                                        -1000.00000    1200.00000     1600.00000    1000.00000   1.5326e+09 carryFwd_4

    28 _C5          BS     2000.00000         .             -Inf           +Inf           .        1.533e+09 _C6
                                              .        2000.00000     2000.00000        +Inf          +Inf

    29 _C6          NU     1400.00000         .             -Inf       1400.00000        -Inf     1.533e+09 _C5
                                              .        1400.00000     2200.00000          .       1.533e+09 unMetDemand_5

    30 x            NS          .             .             .             .             -Inf     1.533e+09 _C5
                                        -4000.00000         .           100.00000        +Inf    1.5326e+09 carryFwd_1

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                                         Page   4

Problem:
Objective:  Objective = 1533000000 (MINimum)

   No. Column name  St       Activity      Obj coef   Lower bound       Activity      Obj coef  Obj value at Limiting
                                           Marginal   Upper bound          range         range  break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- -----------
     1 carryFwd_0   BS          .       1000.00000          .             .             -Inf     1.533e+09
                                              .             +Inf          .             +Inf     1.533e+09

     2 carryFwd_1   BS      100.00000   1000.00000          .          1200.00000          .      1.5329e+09 unMetDemand_1
                                              .             +Inf       -100.00000   24000.00000   1.5353e+09 Production_Cons

     3 carryFwd_2   BS      200.00000   1000.00000          .          1200.00000          .      1.5328e+09 unMetDemand_2
                                              .             +Inf        100.00000   24000.00000   1.5376e+09 Production_Cons

     4 carryFwd_3   BS      800.00000   1000.00000          .          1200.00000          .      1.5322e+09 unMetDemand_3
                                              .             +Inf        800.00000        +Inf          +Inf

     5 carryFwd_4   BS      400.00000   1000.00000          .           400.00000          .      1.5326e+09 _C4
                                              .             +Inf        400.00000        +Inf          +Inf

     6 carryFwd_5   NL          .       1000.00000          .          -800.00000          .      1.5322e+09 unMetDemand_5
                                        1000.00000          +Inf       1200.00000        +Inf     1.5342e+09 Stone_Storage_L

     7 carryFwd_6   BS          .       1000.00000          .             .             -Inf     1.533e+09
                                              .             +Inf          .             +Inf     1.533e+09

     8 q1Stones_1   BS      800.00000 215000.00000          .           800.00000        -Inf          -Inf
                                              .             +Inf        700.00000  238000.00000   1.5514e+09 Production_Cons

     9 q1Stones_2   BS      800.00000 215000.00000          .           800.00000        -Inf          -Inf
                                              .             +Inf        600.00000  239000.00000   1.5522e+09 Production_Cons

    10 q1Stones_3   BS      800.00000 215000.00000          .           800.00000        -Inf          -Inf
                                              .             +Inf        200.00000  240000.00000    1.553e+09 Production_Cons

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                                         Page   5

Problem:
Objective:  Objective = 1533000000 (MINimum)

   No. Column name  St       Activity      Obj coef   Lower bound       Activity      Obj coef  Obj value at Limiting
                                           Marginal   Upper bound          range         range  break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- -----------
    11 q1Stones_4   BS      800.00000 207500.00000          .           800.00000        -Inf          -Inf
                                              .             +Inf        400.00000  241000.00000   1.5598e+09 Production_Cons

    12 q1Stones_5   BS      800.00000 207500.00000          .           800.00000        -Inf          -Inf
                                              .             +Inf        400.00000  242000.00000   1.5606e+09 Production_Cons

    13 q1Stones_6   BS      800.00000 207500.00000          .           800.00000        -Inf          -Inf
                                              .             +Inf        400.00000  242000.00000   1.5606e+09 Production_Cons

    14 q2Stones_1   NL          .     240000.00000          .          -100.00000  238000.00000   1.5328e+09 carryFwd_1
                                        2000.00000          +Inf        800.00000        +Inf     1.5346e+09 q2Stones_3

    15 q2Stones_2   NL          .     240000.00000          .          -200.00000  239000.00000   1.5328e+09 carryFwd_2
                                        1000.00000          +Inf        800.00000        +Inf     1.5338e+09 q2Stones_3
```

```
    16 q2Stones_3   BS     800.00000  240000.00000         .         1200.00000  230500.00000   1.5254e+09 Production_Cons
                                             .            +Inf    -200.00000  241000.00000   1.5338e+09 q2Stones_2

    17 q2Stones_4   BS          .      232500.00000         .              .           -Inf     1.533e+09
                                             .            +Inf          .            +Inf     1.533e+09

    18 q2Stones_5   BS          .      232500.00000         .              .           -Inf     1.533e+09
                                             .            +Inf          .            +Inf     1.533e+09

    19 q2Stones_6   BS    1400.00000  232500.00000         .         1400.00000        -Inf         -Inf
                                             .            +Inf    1000.00000  242000.00000   1.5463e+09 Production_Cons

    20 unMetDemand_0
                    BS          .             .              .              .           -Inf     1.533e+09
                                             .            +Inf          .            +Inf     1.533e+09

GLPK 4.65 - SENSITIVITY ANALYSIS REPORT                                                         Page   6

Problem:
Objective:  Objective = 1533000000 (MINimum)

   No. Column name  St      Activity      Obj coef   Lower bound      Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound         range         range break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
    21 unMetDemand_1
                    NL          .             .              .       -100.00000   -1000.00000   1.5329e+09 carryFwd_1
                                        1000.00000        +Inf      1100.00000        +Inf     1.5341e+09 Stone_Storage_L

    22 unMetDemand_2
                    NL          .             .              .       -200.00000   -1000.00000   1.5328e+09 carryFwd_2
                                        1000.00000        +Inf      1000.00000        +Inf      1.534e+09 Stone_Storage_L

    23 unMetDemand_3
                    NL          .             .              .       -800.00000   -1000.00000   1.5322e+09 carryFwd_3
                                        1000.00000        +Inf       400.00000        +Inf     1.5334e+09 Stone_Storage_L

    24 unMetDemand_4
                    NL          .             .              .       -400.00000   -1000.00000   1.5326e+09 carryFwd_4
                                        1000.00000        +Inf       800.00000        +Inf     1.5338e+09 Stone_Storage_L

    25 unMetDemand_5
                    BS     800.00000          .              .        800.00000          .      1.533e+09 _C6
                                             .            +Inf       400.00000    9500.00000   1.5406e+09 Production_Cons

    26 unMetDemand_6
                    NL          .             .              .       -800.00000          .      1.533e+09 unMetDemand_5
                                             .            +Inf          .            +Inf      1.533e+09 _C5

End of report
```

Q - 6

Decision Variables:

- xi, i=1..5. Mins for each type of advertisement
- d1_0: if this takes positive value, then GRP Overall (Goal 1) >=100
- d1_1: if this takes positive value, then GRP Overall (Goal 1)<=100
- d2_0: if this takes positive value, then GRP Sport (Goal 2)>=20
- d2_1: if this takes positive value, then GRP Sport (Goal 2)<=20
- d3_0: if this takes positive value, then GRP Eng Chnl (Goal 2)>=5
- d3_1: if this takes positive value, then GRP Eng Chnl (Goal 2)<=5

Minimize Objective: d1_1 + d2_1 + d3_0

Subject To

- Goal1_C1: - d1_0 + d1_1 + 4.2 x1 + 3.5 x2 + 2.8 x3 + 2.5 x4 + 0.2 x5 >= 100
- Goal2_C2: - d2_0 + d2_1 + 4.2 x1 + 3.5 x2 >= 20
- Goal3_C3: d3_0 - d3_1 - 0.2 x5 >= -5
- Budget_C4: 120000 x1 + 85000 x2 + 70000 x3 + 60000 x4 + 25000 x5 <= 2000000
- All Variables >= 0

End

In [150]:

```python
from pulp import *
# initialize the model
prob = LpProblem("gpPortfolioBlend", LpMinimize)

# ---------------------
# VARIABLES
# ---------------------

d1 = LpVariable.dicts("d1", range(0, 2), 0, None)
d2 = LpVariable.dicts("d2", range(0, 2), 0, None)
d3 = LpVariable.dicts("d3", range(0, 2), 0, None)

x1=LpVariable("x1",0, None, cat = LpInteger)
x2=LpVariable("x2",0, None, cat = LpInteger)
x3=LpVariable("x3",0, None, cat = LpInteger)
x4=LpVariable("x4",0, None, cat = LpInteger)
x5=LpVariable("x5",0, None, cat = LpInteger)


# Constraints
prob += 4.2 * x1 + 3.5 * x2 + 2.8 * x3 + 2.5 * x4 + 0.2 * x5 >= 100 + d1[0] - d1[1]
prob += 4.2 * x1 + 3.5 * x2 >= 20 + d2[0] - d2[1]
prob += 5 + d3[0] - d3[1] >= 0.2 * x5

prob += 120000* x1 + 85000 * x2 + 70000 * x3 + 60000 * x4 + 25000 * x5 <= 2000000

# Objective function
prob += d1[1] + d2[1] + d3[0], "Objective"

prob.writeLP("gpPortfolioBlend.lp")

status = prob.solve(GLPK(options=["--ranges","gpPortfolioBlend.sen"]))
#print(status)

#print the result
print("Cricket :: {} ::".format(x1.value()))
print("Oth Sport :: {} ::".format(x2.value()))
print("Hindi Serial :: {} ::".format(x3.value()))
print("Hindi Movie :: {} ::".format(x4.value()))
print("English News :: {} ::".format(x5.value()))

for i in range(0, 2):
    print("D1 {} :: {}".format(i, d1[i].value()))


for i in range(0, 2):
    print("D2 {} :: {}".format(i, d2[i].value()))

for i in range(0, 2):
    print("D3 {} :: {}".format(i, d3[i].value()))

print("Objective {}::".format(value(prob.objective)))
```

```
Cricket :: 0 ::
Oth Sport :: 8 ::
Hindi Serial :: 0 ::
Hindi Movie :: 22 ::
English News :: 0 ::
D1 0 :: 0.0
D1 1 :: 17.0
D2 0 :: 0.0
D2 1 :: 0.0
D3 0 :: 0.0
D3 1 :: 0.0
Objective 17.0::
```

In [ ]: