SID: 480110301

<u>Question 1</u>

a. Let X_i be the amount of investment in fund A and B, $i \in A, B$.

Constraints:

$$X_A + X_B = 80000 \text{ (total investment)}$$

$$X_A \geq 30000 \text{ (fund A)}$$

$$X_B \geq 20000 \text{ (fund B)}$$

$$X_A, X_B \geq 0 \text{ (non} - \text{negative)}$$

Objective: ***maximize*** $\mathbf{0.14}\boldsymbol{X_A} + \mathbf{0.11}\boldsymbol{X_B}$

```
In [1]: import gurobi as grb

In [2]: fund = ["A","B"]
        return_rate = {"A":0.14,"B":0.11}
        risk = {"A":0.6,"B":0.4}

In [3]: A1 = grb.Model('Q1a')

        Using license file /Users/chelseayeung/gurobi.lic
        Academic license - for non-commercial use only

In [4]: X = A1.addVars(fund,obj = return_rate,name = "X")

In [5]: A1.addConstr(X["A"]+X["B"]==80000)
        A1.addConstr(X["A"]>=30000)
        A1.addConstr(X["B"]>=20000)

Out[5]: <gurobi.Constr *Awaiting Model Update*>

In [6]: A1.ModelSense = grb.GRB.MAXIMIZE
```

In [7]:
```
A1.optimize()
```

```
Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (mac64)
Optimize a model with 3 rows, 2 columns and 4 nonzeros
Model fingerprint: 0x6937dfc8
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e-01, 1e-01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+04, 8e+04]
Presolve removed 3 rows and 2 columns
Presolve time: 0.00s
Presolve: All rows and columns removed
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    1.0600000e+04   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.01 seconds
Optimal objective  1.060000000e+04
```

In [8]:
```
A1.write("Q1a.lp")
```

In [9]:
```
return_a = A1.ObjVal/80000
risk_a = (sum(X[i].x*risk[i] for i in X))/80000
```

In [10]:
```
print("The maximum return is {}.".format(return_a))
print("The risk is {}.".format(risk_a))
print("The investment in fund A is ${}.".format(X["A"].x))
print("The investment in fund B is ${}.".format(X["B"].x))
```

```
The maximum return is 0.1325.
The risk is 0.55.
The investment in fund A is $60000.0.
The investment in fund B is $20000.0.
```

The objective value represents that the maximum return rate is 13.25%. $X_A = 60000$, $X_B = 20000$ therefore the investment in fund A is $60000 while the investment in fund B is $20000. In this case the risk is 0.55.

b. Let $\mathrm{X_i}$ be the amount of investment in fund A and B, $\mathrm{i} \in \mathrm{A, B}$.

Constraints:

$$\mathrm{X_A + X_B = 80000\ (total\ investment)}$$

$$\mathrm{X_A \geq 30000\ (fund\ A)}$$

$$\mathrm{X_B \geq 20000\ (fund\ B)}$$

$$\mathrm{X_A, X_B} \geq 0\ (non-negative)$$

Objective: ***minimize*** $\boldsymbol{0.6X_A + 0.4X_B}$

```
In [11]: A2 = grb.Model('Q1b')
```

```
In [12]: X2 = A2.addVars(fund,obj = risk,name = "X")
```

```
In [13]: A2.addConstr(X2["A"]+X2["B"]==80000)
         A2.addConstr(X2["A"]>=30000)
         A2.addConstr(X2["B"]>=20000)

Out[13]: <gurobi.Constr *Awaiting Model Update*>
```

```
In [14]: A2.ModelSense = grb.GRB.MINIMIZE
```

```
In [15]: A2.write("Q1b.lp")
```

```
In [16]: A2.optimize()

         Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (mac64)
         Optimize a model with 3 rows, 2 columns and 4 nonzeros
         Model fingerprint: 0xe99ce532
         Coefficient statistics:
           Matrix range     [1e+00, 1e+00]
           Objective range  [4e-01, 6e-01]
           Bounds range     [0e+00, 0e+00]
           RHS range        [2e+04, 8e+04]
         Presolve removed 3 rows and 2 columns
         Presolve time: 0.01s
         Presolve: All rows and columns removed
         Iteration    Objective       Primal Inf.    Dual Inf.      Time
                0    3.8000000e+04   0.000000e+00   0.000000e+00      0s

         Solved in 0 iterations and 0.01 seconds
         Optimal objective  3.800000000e+04
```

```
In [17]: risk_b = A2.ObjVal/80000
         return_b = (sum(X2[i].x*return_rate[i] for i in X2))/80000
```

```
In [18]: print("The minimum risk is {}.".format(risk_b))
         print("The return is {}.".format(return_b))

         print("The investment in fund A is ${}.".format(X2["A"].x))
         print("The investment in fund B is ${}.".format(X2["B"].x))

         The minimum risk is 0.475.
         The return is 0.12125.
         The investment in fund A is $30000.0.
         The investment in fund B is $50000.0.
```

The objective value represents that the minimum risk is 0.475. $X_A = 30000$, $X_B = 50000$ therefore the investment in fund in fund A is 30000 while the investment in fund B is 50000. In this case the overall return rate is 12.125%.

c. Let X_i be the amount of investment in fund A and B, $i \in A, B$.

Minimum risk: 0.475; Maximum return rate: 0.1325

Constraints:

$$X_A + X_B = 80000 \text{ (total investment)}$$

$$X_A \geq 30000 \text{ (fund A)}$$

$$X_B \geq 20000 \text{ (fund B)}$$

$$X_A, X_B \geq 0\ (non-negative)$$

$$\frac{0.1325 - \frac{0.14X_A + 0.11X_B}{80000}}{0.1325} \leq Q\ (return)$$

$$\frac{\frac{0.6X_A + 0.4X_B}{80000} - 0.475}{0.475} \leq Q\ (risk)$$

$$Q \geq 0$$

Objective: ***minimize Q***

In [19]:
```
A3 = grb.Model('Q1c')
```

In [20]:
```
X3 = A3.addVars(fund,name ="X")
Q = A3.addVar(obj=1,name="Q")
```

In [21]:
```
min_risk = A2.ObjVal
max_return = A1.ObjVal
```

In [22]:
```
A3.addConstr(X3["A"]+X3["B"]==80000)
A3.addConstr(X3["A"]>=30000)
A3.addConstr(X3["B"]>=20000)
A3.addConstr((0.1325-(0.14*X3["A"]+0.11*X3["B"])/80000)/0.1325<=Q)
A3.addConstr(((0.6*X3["A"]+0.4*X3["B"])/80000-0.475)/0.475<=Q)
```

Out[22]:
```
<gurobi.Constr *Awaiting Model Update*>
```

In [23]:
```
A3.ModelSense = grb.GRB.MINIMIZE
```

In [24]:
```
A3.write("Q1c.lp")
```

```
In [25]: A3.optimize()
```

```
Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (mac64)
Optimize a model with 5 rows, 3 columns and 10 nonzeros
Model fingerprint: 0x174147e4
Coefficient statistics:
  Matrix range     [1e-05, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 8e+04]
Presolve removed 3 rows and 1 columns
Presolve time: 0.01s
Presolved: 2 rows, 2 columns, 4 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   1.578947e-01   0.000000e+00      0s
       2    5.5214724e-02   0.000000e+00   0.000000e+00      0s

Solved in 2 iterations and 0.01 seconds
Optimal objective  5.521472393e-02
```

```
In [26]: Q = A3.ObjVal
risk_c = (sum(X3[i].x*risk[i] for i in X3))/80000
return_c = (sum(X3[i].x*return_rate[i] for i in X3))/80000
```

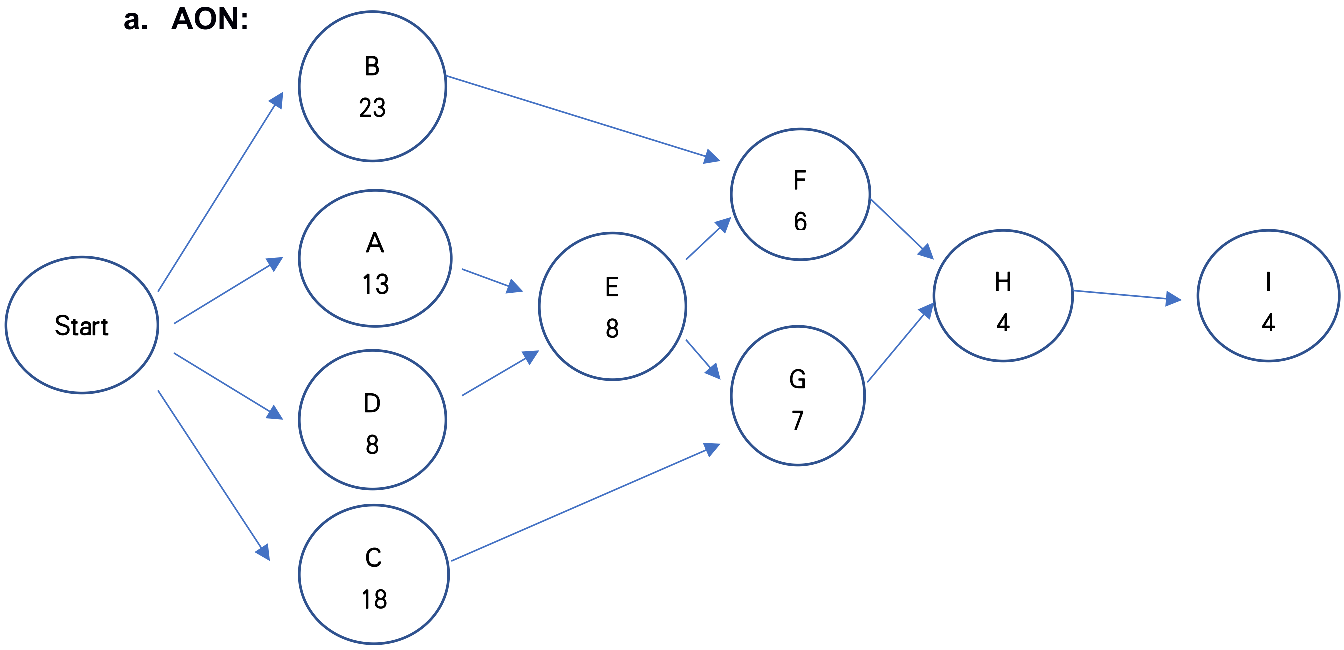
```
In [27]: print("The minimum Q is {}.".format(Q))
print("The return is {}.".format(return_c))
print("The risk is {}.".format(risk_c))
print("The investment in fund A is ${}.".format(X3["A"].x))
print("The investment in fund B is ${}.".format(X3["B"].x))
```

```
The minimum Q is 0.05521472392638O715.
The return is 0.1251840490797546.
The risk is 0.5012269938650308.
The investment in fund A is $40490.797546012305.
The investment in fund B is $39509.202453987695.
```

The objective value indicates that the minimum value of percentage deviation Q is 5.52%. $X_A = 40490.8,\ X_B = 39509.2$, therefore the optimal solution of investments is $\$40490.8$ for fund A and $\$39509.2$ for fund B. In this case, the overall return rate is 12.5% and the risk is 0.5.

Question 2

a. AON:



b.

Activity	Duration	EST	EFT	LST	LFT	Slack
A	13	0	13	1	14	1
B	23	0	23	0	23	0
C	18	0	18	4	22	4
D	8	0	8	6	14	6
E	8	13	21	14	22	1
F	6	23	29	23	29	0
G	7	21	28	22	29	1
H	4	29	33	29	33	0
I	4	33	37	33	37	0

Activity B, F, H and I are critical activities.

c. The duration of the project 37 days. The critical path is B-F-H-I.

d.

Activity	Normal		Crash			
	Time(days)	Cost (dollars)	Time (days)	Cost (dollars)	Cost per day	Allowable crash
A	13	11,000	10	15,000	1333.33	3
B	23	5,000	21	6,000	500	2
C	18	3,000	15	3,500	166.67	3
D	8	1,500	6	2,000	250	2
E	8	750	5	1,200	150	3
F	6	600	4	1,200	300	2
G	7	1,000	5	1,500	250	2
H	4	250	3	450	200	1
I	4	200	2	300	50	2

There are 6 paths: BFHI, AEFHI, AEGHI, DEFHI, DEGHI, CGHI

Path	Normal	2*I (2*50)	H(200)	2*F(2*300)	3*E(3*150)	2*B(2*500)
BFHI	37	35	34	32	32	30
AEFHI	35	33	32	30	27	27
AEGHI	36	34	33	33	30	30
DEFHI	30	28	27	25	22	22
DEGHI	31	29	28	28	25	25
CGHI	33	31	30	30	30	30

Crashing cost	100	200	600	450	1000

The earliest completion time is 30 days. It is impossible for the owner to complete the project within 20 days but it can at least meet the requirement of 32 days. The cost is $25650, which is calculated by $Total\ cost = Normal\ cost + Crashing\ cost = (11000 + 5000 + 3000 + 1500 + 750 + 600 + 1000 + 250 + 200) + (100 + 200 + 600 + 450 + 1000) = 25650.$

Question 3

a. Let X_i be the number of trucks i, where $i \in L, M, S$ representing large, medium and small.

Constraints:

$$X_L + X_M + X_S = 70\ (total\ trucks)$$

$$X_L \geq 70 \times 0.2\ (20\%\ \ large\ trucks)$$

$$8X_L + 5X_M + 3X_S \geq 350\ (capacity)$$

$$X_L, X_M, X_S \geq 0\ (non-negative)$$

$$X_L, X_M, X_S\ are\ integers$$

Objective: $minimize\ 650X_L + 500X_M + 250X_S$

```
In [2]: A1 = grb.Model('Q3a')

        Using license file /Users/chelseayeung/gurobi.lic
        Academic license - for non-commercial use only

In [3]: cost  = {"L":650,"M":500,"S":250}
        pollute = {"L":8,"M":5,"S":7}
        safety = {"L":9,"M":8,"S":6}
        capacity = {"L":8,"M":5,"S":3}
        X1 = A1.addVars(cost, obj=cost,name ="X",vtype=grb.GRB.INTEGER)

In [4]: A1.addConstr(sum(X1[i] for i in X1)==70,name="Total_trucks")
        A1.addConstr(X1["L"]>=70*0.2,name="Minimum_large_trucks")
        A1.addConstr(sum(X1[i]*capacity[i] for i in X1)>=350,name="Total_capacity")

Out[4]: <gurobi.Constr *Awaiting Model Update*>

In [5]: A1.ModelSense = grb.GRB.MINIMIZE

In [6]: A1.write('Q3a.lp')
```

```
In [8]: print("The minimum cost is ${}".format(A1.ObjVal))
        for i in X1:
            print("X[{}]={}".format(i,int(X1[i].x)))
```

```
The minimum cost is $28700.0
X[L]=28
X[M]=0
X[S]=42
```

```
In [9]: mean_pollute = sum(X1[i].x*pollute[i] for i in X1)/70
        mean_safety = sum(X1[i].x*safety[i] for i in X1)/70
        print("The mean pollute rating is {}.".format(mean_pollute))
        print("The mean safety rating is {}.".format(mean_safety))
```

```
The mean pollute rating is 7.4.
The mean safety rating is 7.2.
```

$X_L = 28, X_M = 0, X_S = 42$, therefore 28 large trucks and 42 small trucks are used. The minimum cost is $28700. The mean pollution rating is 7.4 and the mean safety rating is 7.2 for this fleet.

b. Let X_i be the number of trucks i, where $i \in L, M, S$ representing large, medium and small.

Constraints:

$$X_L + X_M + X_S = 70\ (total\ trucks)$$

$$X_L \geq 70 \times 0.2\ (\#\ large\ trucks)$$

$$8X_L + 5X_M + 3X_S \geq 350\ (capacity)$$

$$X_L, X_M, X_S \geq 0\ (non-negative)$$

$$X_L, X_M, X_S\ are\ integers$$

Objective: $minimize\ 8X_L + 5X_M + 7X_S$

```
In [10]: A2 = grb.Model('Q3b')
```

```
In [11]: cost  = {"L":650,"M":500,"S":250}
         mean_pollute = {"L":8,"M":5,"S":7}
         safety = {"L":9,"M":8,"S":6}
         capacity = {"L":8,"M":5,"S":3}

         X2 = A2.addVars(mean_pollute,obj=mean_pollute, name="X",vtype=grb.GRB.INTEG
```

```
In [12]: A2.addConstr(sum(X2[i] for i in X2)==70,name="Total_trucks")
         A2.addConstr(X2["L"]>=70*0.2,name="Minimum_large_trucks")
         A2.addConstr(sum(X2[i]*capacity[i] for i in X2)>=350,name="Total_capacity")

Out[12]: <gurobi.Constr *Awaiting Model Update*>
```

```
In [13]: A2.ModelSense = grb.GRB.MINIMIZE
```

```
In [14]: A2.optimize()

  Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (mac64)
  Optimize a model with 3 rows, 3 columns and 7 nonzeros
  Model fingerprint: 0x5b4098dd
  Variable types: 0 continuous, 3 integer (0 binary)
  Coefficient statistics:
    Matrix range     [1e+00, 8e+00]
    Objective range  [5e+00, 8e+00]
    Bounds range     [0e+00, 0e+00]
    RHS range        [1e+01, 4e+02]
  Presolve removed 3 rows and 3 columns
  Presolve time: 0.00s
  Presolve: All rows and columns removed

  Explored 0 nodes (0 simplex iterations) in 0.02 seconds
  Thread count was 1 (of 8 available processors)

  Solution count 1: 392

  Optimal solution found (tolerance 1.00e-04)
  Best objective 3.920000000000e+02, best bound 3.920000000000e+02, gap 0.0
  000%
```

```
In [15]: A2.write('Q3b.lp')
```

```
In [16]: mean_safety_b = sum(X2[i].x*safety[i] for i in X2)/70
         cost_b = sum(X2[i].x*cost[i] for i in X2)
         for i in X2:
             print("X[{}]={}".format(i,int(X2[i].x)))

         print("The minimum mean pollute rating is {}.".format(A2.ObjVal/70))
         print("The mean safety rating is {}.".format(mean_safety_b))
         print("The cost is {}.".format(cost_b))

         X[L]=14
         X[M]=56
         X[S]=0
         The minimum mean pollute rating is 5.6.
         The mean safety rating is 8.2.
         The cost is 37100.0.
```

$X_L = 14, X_M = 56, X_S = 0$, therefore 14 large trucks and 56 medium trucks are used. The minimum mean pollute rating is 5.6. In this case, the mean safety rating is 8.2 and the cost is $37100.

c. Let X_i be the number of trucks i, where $i \in L, M, S$ representing large, medium and small.

Constraints:

$$X_L + X_M + X_S = 70\ (total\ trucks)$$

$$X_L \geq 70 \times 0.2\ (\#\ large\ trucks)$$

$$8X_L + 5X_M + 3X_S \geq 350\ (capacity)$$

$$X_L, X_M, X_S \geq 0\ (non-negative)$$

$$X_L, X_M, X_S\ are\ integers$$

Objective: $minimize\ 9X_L + 8X_M + 6X_S$

In [17]:
```
A3 = grb.Model('Q3c')
```

In [18]:
```
cost  = {"L":650,"M":500,"S":250}
pollute = {"L":8,"M":5,"S":7}
mean_safety = {"L":9,"M":8,"S":6}
capacity = {"L":8,"M":5,"S":3}

X3 = A3.addVars(mean_safety,obj=mean_safety,name="X",vtype=grb.GRB.INTEGER)
```

In [19]:
```
A3.addConstr(sum(X3[i] for i in X3)==70,name="Total_trucks")
A3.addConstr(X3["L"]>=70*0.2,name="Minimum_large_trucks")
A3.addConstr(sum(X3[i]*capacity[i] for i in X3)>=350,name="Total_capacity")
```

Out[19]:
```
<gurobi.Constr *Awaiting Model Update*>
```

In [20]:
```
A3.ModelSense = grb.GRB.MAXIMIZE
```

In [21]:
```
A3.optimize()
```

```
Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (mac64)
Optimize a model with 3 rows, 3 columns and 7 nonzeros
Model fingerprint: 0x34a1bab1
Variable types: 0 continuous, 3 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 8e+00]
  Objective range  [6e+00, 9e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+01, 4e+02]
Presolve removed 3 rows and 3 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds
Thread count was 1 (of 8 available processors)

Solution count 1: 630

Optimal solution found (tolerance 1.00e-04)
Best objective 6.300000000000e+02, best bound 6.300000000000e+02, gap 0.0
000%
```

```
In [22]: mean_pollute_c = sum(X3[i].x*pollute[i] for i in X3)/70
cost_c = sum(X3[i].x*cost[i] for i in X3)
for i in X3:
    print("X[{}]={}".format(i,int(X3[i].x)))

print("The maximum mean safety rating is {}.".format(A3.ObjVal/70))
print("The mean pollute rating is {}.".format(mean_pollute_c))
print("The cost is {}.".format(cost_c))

X[L]=70
X[M]=0
X[S]=0
The maximum mean safety rating is 9.0.
The mean pollute rating is 8.0.
The cost is 45500.0.
```

$X_L = 70, X_M = 0, X_S = 0$, therefore 70 large trucks are used. The maximum mean safety rating is 9. In this case, the mean pollute rating is 8 and the cost is $45500.

d. Let X_i be the number of trucks i, where $i \in L, M, S$ representing large, medium and small.

Minimum mean pollute rating: 5.6

Maximum safety rating: 9

Minimum cost: 28700

Constraints:

$$X_L + X_M + X_S = 70 \ (total\ trucks)$$

$$X_L \geq 70 \times 0.2 \ (\#\ large\ trucks)$$

$$8X_L + 5X_M + 3X_S \geq 350 \ (capacity)$$

$$2 \times \frac{9 - \frac{9X_L + 8X_M + 6X_S}{70}}{9} \leq Q \ (safety\ rating)$$

$$\frac{\frac{8X_L + 5X_M + 7X_S}{70} - 5.6}{5.6} \leq Q \ (pollute\ rating)$$

$$\frac{(650X_L + 500X_M + 250X_S) - 28700}{28700} \leq Q \ (cost)$$

$$Q \geq 0$$

$$X_L, X_M, X_S \geq 0 \ (non-negative)$$

$$X_L, X_M, X_S \text{ are integers}$$

Objective: $minimize\ Q$

In [23]:
```
A4 = grb.Model('Q3d')
```

In [24]:
```
cost  = {"L":650,"M":500,"S":250}
pollute = {"L":8,"M":5,"S":7}
safety = {"L":9,"M":8,"S":6}
capacity = {"L":8,"M":5,"S":3}
X4 = A4.addVars(cost, obj=0,name ="X",vtype=grb.GRB.INTEGER)
Q = A4.addVar(obj=1,name ="Q")
```

In [25]:
```
A4.addConstr(sum(X4[i] for i in X4)==70,name="Total_trucks")
A4.addConstr(X4["L"]>=70*0.2,name="Minimum_large_trucks")
A4.addConstr(sum(X4[i]*capacity[i] for i in X4)>=350,name="Total_capacity")

A4.addConstr(2*((9-(sum(X4[i]*safety[i] for i in X4)/70))/9)<=Q, name = "sa
A4.addConstr((sum(X4[i]*pollute[i] for i in X4)/70-5.6)/5.6<=Q,name = "poll
A4.addConstr((sum(X4[i]*cost[i] for i in X4)-28700)/28700<=Q,name="cost")
```

Out[25]:
```
<gurobi.Constr *Awaiting Model Update*>
```

In [26]:
```
A4.write('Q3d.lp')
```

In [27]:
```
A4.ModelSense = grb.GRB.MINIMIZE
```

In [28]:
```
A4.optimize()
```

```
Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (mac64)
Optimize a model with 6 rows, 4 columns and 19 nonzeros
Model fingerprint: 0xcdccce48
Variable types: 1 continuous, 3 integer (0 binary)
Coefficient statistics:
  Matrix range     [9e-03, 8e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 4e+02]
Presolve removed 1 rows and 0 columns
Presolve time: 0.00s
Presolved: 5 rows, 4 columns, 14 nonzeros
Variable types: 1 continuous, 3 integer (0 binary)
Found heuristic solution: objective 0.3111111

Root relaxation: objective 2.262211e-01, 3 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node T
ime


     0     0    0.22622    0    2    0.31111    0.22622  27.3%     -    0
s
H    0     0                       0.2285714    0.22622  1.03%     -    0
s
     0     0    0.22646    0    2    0.22857    0.22646  0.92%     -    0
s
H    0     0                       0.2282230    0.22646  0.77%     -    0
s
     0     0    0.22646    0    2    0.22822    0.22646  0.77%     -    0
s
```

```
Explored 1 nodes (5 simplex iterations) in 0.02 seconds
Thread count was 8 (of 8 available processors)

Solution count 3: 0.228223 0.228571 0.311111

Optimal solution found (tolerance 1.00e-04)
Best objective 2.282229965157e-01, best bound 2.282229965157e-01, gap 0.0
000%
```

In [29]:
```
safety_d = sum(X4[i].x*safety[i] for i in X4)/70
pollute_d = sum(X4[i].x*pollute[i] for i in X4)/70
cost_d = sum(X4[i].x*cost[i] for i in X4)
```

In [30]:
```
for i in X4:
    print("X[{}]={}".format(i,int(X4[i].x)))

print("The minimum value of Q is {}.".format(A4.ObjVal))
print("The mean safety rating is {}.".format(safety_d))
print("The mean pollute rating is {}.".format(pollute_d))
print("The cost is {}.".format(cost_d))
```

```
X[L]=15
X[M]=47
X[S]=8
The minimum value of Q is 0.22822299651567865.
The mean safety rating is 7.985714285714286.
The mean pollute rating is 5.871428571428571.
The cost is 35250.0.
```

$X_L = 15,\ X_M = 47,\ X_S = 8$, therefore 15 large trucks, 47 medium trucks and 8 large trucks are used. The minimum value of percentage deviation Q is 22.82%. In this case the mean safety rating is 7.99 the mean pollute rating is 5.87 and the cost is $35250.