

Nirbhay Sharma (B19CSE114)

optimization for machine learning

Que-1 & Que-4

Code

```
import numpy as np
from gekko import GEKKO
import sys

m = GEKKO(remote = False)

abc_matrix = {
    "1":
        {
            "A": np.array([[6, 3, 5, 2],[1, 0, 0, 1],[-1, 0, 1,
0],[0, -1, 0, 1]], dtype=np.float64),
            "b" : np.array([[10],[1],[0],
[0]],dtype=np.float64),
            "c" : np.array([[9],[5],[6],[4]],dtype=np.float64)
        },
    "4":
        {
            "A": np.array([[55, 45, 60, 50, 30],[40, 35, 25,
35, 30],[25, 20, 0, 30, 0],[0, 0, 1, 1, 0]], dtype=np.float64),
            "b" : np.array([[150],[110],[60],
[1]],dtype=np.float64),
            "c" : np.array([[120],[85],[105],[140],
[70]],dtype=np.float64)
        }
}

question_no = sys.argv[1]
A = abc_matrix[question_no]['A']
b = abc_matrix[question_no]['b']
c = abc_matrix[question_no]['c']

# print(A)

# print(b)

# print(c)

z = m.Array(m.Var, c.shape[0],integer=True,lb=0,ub=1)
m.qobj(c,x=z,otype='max')
m.axb(A,b,x=z,etype='<=')
m.options.SOLVER = 1
m.solve(dispatch=False)
```

```
print("Objective: ", m.options.OBJFCNVAL)
print(z)
```

```
"""
```

```
Ans of que-1
```

```
A
```

```
[[ 6.  3.  5.  2.]
 [ 1.  0.  0.  1.]
 [-1.  0.  1.  0.]
 [ 0. -1.  0.  1.]]
```

```
b
```

```
[[10.  1.  0.  0.]]
Objective: -14.0
[[1.0] [1.0] [0.0] [0.0]]
```

```
Ans of que-4
```

```
A
```

```
[[55. 45. 60. 50. 30.]
 [40. 35. 25. 35. 30.]
 [25. 20.  0. 30.  0.]
 [ 0.  0.  1.  1.  0.]]
```

```
b
```

```
[[150. 110.  60.  1.]]
Objective: -330.0
[[1.0] [0.0] [0.0] [1.0] [1.0]]
"""
```

Que-2

Code

```
from gekko import GEKKO
import numpy as np
import math

m = GEKKO(remote=False)
A = np.array([[5,7],[4,1],[3,-2],[-1, 0],[0, -1]], dtype=np.float64)
b = np.array([[27],[14],[9],[0],[0]], dtype=np.float64)
c = np.array([[7],[3]], dtype=np.float64)
z1=m.Var(1,integer=True,lb=0,ub=1)
z2=m.Var(1,integer=False,lb=0,ub=100)
z=[z1,z2]
print(z1,z2,z)
m.qobj(c,x=z,otype='max')
m.axb(A,b,x=z,etype='<=')
m.options.SOLVER = 1
m.solve()
print('Objective: ', m.options.OBJFCNVAL)
print(z)
print('x: ', z[0].value[0])
```

```
print('y: ', z[1].value[0])

"""
Objective: -16.428571429
[[1.0], [3.1428571429]]
x: 1.0
y: 3.1428571429
"""
```

Que-3

```
from gekko import GEKKO
import numpy as np
import math

m = GEKKO(remote=False)
A = np.array([[3,-2],[-8,10],[-1, 0]], dtype=np.float64)
b = np.array([[1],[10],[-0.3]],dtype=np.float64)
c = np.array([[1],[1]],dtype=np.float64)
z2=m.Var(1,integer=True,lb=0,ub=1)
z1=m.Var(1,integer=False,lb=0.3,ub=1000)
z=[z1,z2]
print(z1,z2,z)
m.qobj(c,x=z,otype='min')
m.axb(A,b,x=z,etype='<=')
m.options.SOLVER = 1
m.solve()
print('Objective: ', m.options.OBJFCNVAL)
print(z)
print('x: ', z[0].value[0])
print('y: ', z[1].value[0])

"""
Objective: 1.3
[[0.3], [1.0]]
x: 0.3
y: 1.0
"""
```