Introduction

Name : Kartik Srinivas

Roll : ES20BTECH11015

Welcome to the analysis of the Robust LP Problem !

1. All the equations have been coded in LaTeX by me
2. The necessary explanations have been provided

```
import numpy as np
import cvxpy as cp
```

(a) Convexity

$$f(x) = \sup_{c \in C} c^T x = \sup_{c \in C} \quad \text{(Affine)}$$
$$Ax \geq b$$

This function is a maximum of affine functions = maximum of convex functions = convex

Constraints are polyhedra, so they are convex

```
np.random.seed(10)
(m,n) = (30,10)
A = np.random.rand(m,n)
A = np.asmatrix(A)
b = np.random.rand(m,1)
b = np.asmatrix(b)
c_nom = np.ones((n,1)) + np.random.rand(n,1)
c_nom = np.asmatrix(c_nom)
```

```
print(c_nom.shape)
print(A.shape)
print(b.shape)
print(type(A))
```

```
(10, 1)
(30, 10)
(30, 1)
<class 'numpy.matrix'>
```

```
x = cp.Variable((10,1))
```

Framing F and g

F needs to be formed according to the constraints defined by

$$
\begin{aligned}
c &\leq 1.25\ c_{nom} \\
-c &\leq -0.75\ c_{nom} \\
\vec{1}^T c &\leq \vec{1}^T c_{nom} \\
-\vec{1}^T c &\leq -\vec{1}^T c_{nom}
\end{aligned}
$$

In order to make this in matrix form we do the following

$$
\begin{bmatrix} I_{n\times n} \\ -I_{n\times n} \\ \vec{1}^T c \\ -\vec{1}^T c \end{bmatrix} \cdot \vec{c} = \begin{bmatrix} 1.25\ c_{nom} \\ -0.75\ c_{nom} \\ \vec{1}^T c_{nom} \\ -\vec{1}^T c_{nom} \end{bmatrix}
$$

```
#Now to generate the values of F and g
I = np.eye(n)
Stacked_identitites = np.block([
    [I],
    [-I]
])
F = np.vstack((np.vstack((Stacked_identitites, np.ones(n))), -1 *np.ones(n)))
print(F)
print(F.shape)
```

```
[[ 1.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  1.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  1.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  1.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  1.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  1.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.]
 [-1. -0. -0. -0. -0. -0. -0. -0. -0. -0.]
 [-0. -1. -0. -0. -0. -0. -0. -0. -0. -0.]
 [-0. -0. -1. -0. -0. -0. -0. -0. -0. -0.]
 [-0. -0. -0. -1. -0. -0. -0. -0. -0. -0.]
 [-0. -0. -0. -0. -1. -0. -0. -0. -0. -0.]
 [-0. -0. -0. -0. -0. -1. -0. -0. -0. -0.]
 [-0. -0. -0. -0. -0. -0. -1. -0. -0. -0.]
 [-0. -0. -0. -0. -0. -0. -0. -1. -0. -0.]
 [-0. -0. -0. -0. -0. -0. -0. -0. -1. -0.]
 [-0. -0. -0. -0. -0. -0. -0. -0. -0. -1.]
 [ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]
```

```
 [-1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]]
(22, 10)
```

```
#to calculate g
#print(c_nom)
stacked_noms = np.vstack( ( (1.25 * c_nom), (-0.75 * c_nom) )  )
# print(stacked_noms.shape)
sum = np.sum(c_nom)
#print(sum)
g_1 = np.vstack((stacked_noms, 1.1*sum))
g = np.vstack((g_1, -0.9*sum))
print(g)
print(g.shape)
```

```
[[  2.00180047]
 [  1.37159062]
 [  1.6109186 ]
 [  2.15099988]
 [  1.93850758]
 [  2.29822127]
 [  1.97541419]
 [  1.48071467]
 [  2.01937765]
 [  2.35869386]
 [ -1.20108028]
 [ -0.82295437]
 [ -0.96655116]
 [ -1.29059993]
 [ -1.16310455]
 [ -1.37893276]
 [ -1.18524851]
 [ -0.8884288 ]
 [ -1.21162659]
 [ -1.41521632]
 [ 16.90149013]
 [-13.82849192]]
(22, 1)
```

```
lam = cp.Variable(g.shape)
```

(b),(c) Framing a Dual

THe dual of the problem is

$$f(x) = \max_{\lambda \geq 0} g(\lambda)$$

Where :-

$$g(\lambda) = \underset{c}{min}\ [c^T x + \lambda^T (Fc - g)]$$

Unbounded unless coefficient of c is zero !

$$F^T\lambda = x$$
$$\therefore g(\lambda) = -\lambda^T g$$

Therefore the solution of the primal

$$\min_x \ (\max_\lambda \ g(\lambda)) = \min_{x,\lambda} \ (\lambda^T g)$$
$$\lambda \geq 0$$
$$F^T\lambda = x$$
$$Ax \geq b$$

```
objective = cp.Minimize(lam.T @ g)
constraints = [F.T @ lam == x, lam >= 0 , A@x >= b]
prob = cp.Problem(objective, constraints)
result_dual_form = prob.solve()
print(result_dual_form)
x_robust = x.value
```

```
3.1659610520520705
```

```
#solving the primal using c_nom
x2 = cp.Variable((10,1))
test_obj = cp.Minimize(c_nom.T @ x2)
test_constraints = [A@x2 >= b]
test_prob = cp.Problem(test_obj,test_constraints)
result_nominal = test_prob.solve()
print("Result of Nominal C = ",result_nominal)
```

```
Result of Nominal C =  2.1092714620826003
```

```
print("Result of Robust C = ", (c_nom.T @ x_robust)[0,0])
```

```
Result of Robust C =  2.5232088648898556
```

Final Values

$$c_{nom}^T x_{robust} = 2.52$$
$$c_{nom}^T x_{nominal} = 2.109$$
$$f(x)_{min} = 3.165$$