190031187 **NERELLA VENKATA RADHAKRISHNA**

MP-1 PRACTICAL-1

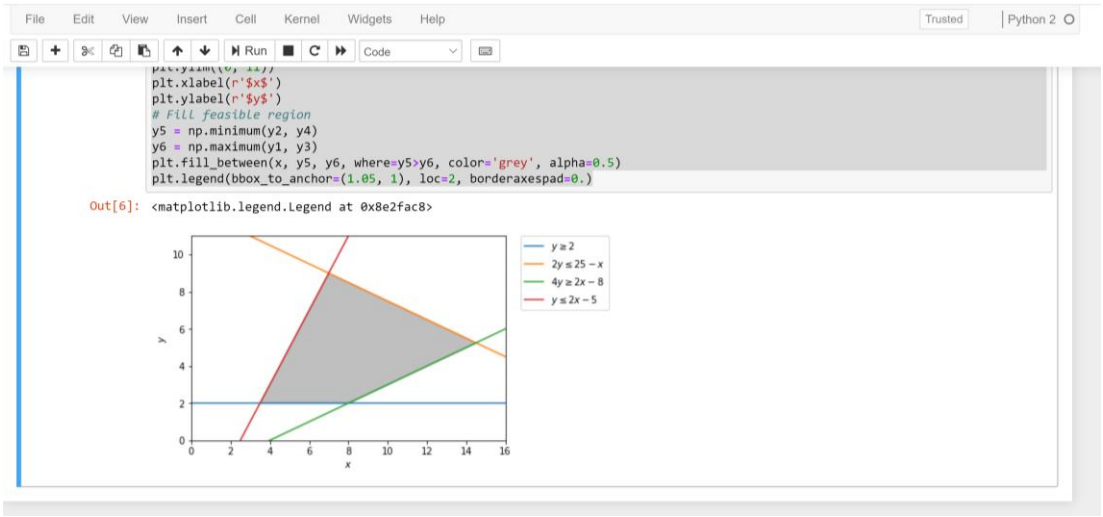
1. **Develop a python program to demonstrate the Graphical method in Linear Programming.**

QUESTION:

Maximize
Z=4x+3yZ=4x+3y
Subject TO:
x≥0
y≥2
2y≤25−x
4y≥2x−8
y≤2x−5
Solve LP graphically using python

Code:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
# Construct lines
# x > 0
x = np.linspace(0, 20, 2000)
# y >= 2
y1 = (x*0) + 2
# 2y <= 25 - x
y2 = (25-x)/2.0
# 4y >= 2x - 8
y3 = (2*x-8)/4.0
# y <= 2x - 5
y4 = 2 * x -5
# Make plot
plt.plot(x, y1, label=r'$y\geq2$')
plt.plot(x, y2, label=r'$2y\leq25-x$')
plt.plot(x, y3, label=r'$4y\geq 2x - 8$')
plt.plot(x, y4, label=r'$y\leq 2x-5$')
plt.xlim((0, 16))
plt.ylim((0, 11))
plt.xlabel(r'$x$')
plt.ylabel(r'$y$')
# Fill feasible region
y5 = np.minimum(y2, y4)
y6 = np.maximum(y1, y3)
plt.fill_between(x, y5, y6, where=y5>y6, color='grey', alpha=0.5)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```



2. **Develop a python program to demonstrate the Simplex method**

QUESTION:

Maximize Z=3 x1 + 5 x2

Subject To:

3 x1 + 2 x2 =18

X1 <= 4

2 x2 <= 12

X1 >=0

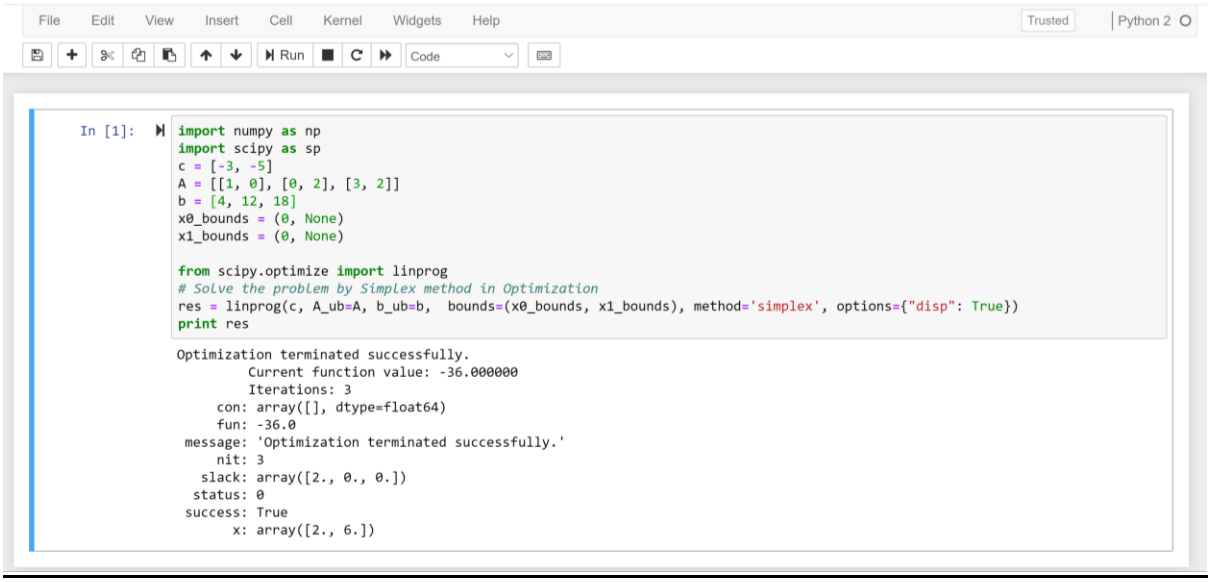
X2 >= 0

Solve LP using simplex method using Python

Code:

```
import numpy as np
import scipy as sp
c = [-3, -5]
A = [[1, 0], [0, 2], [3, 2]]
b = [4, 12, 18]
x0_bounds = (0, None)
x1_bounds = (0, None)

from scipy.optimize import linprog
# Solve the problem by Simplex method in Optimization
res = linprog(c, A_ub=A, b_ub=b,  bounds=(x0_bounds, x1_bounds), method='simplex',
options={"disp": True})
print res
```


File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 2
Run
Code
In [1]:
import numpy as np
import scipy as sp
c = [-3, -5]
A = [[1, 0], [0, 2], [3, 2]]
b = [4, 12, 18]
x0_bounds = (0, None)
x1_bounds = (0, None)
from scipy.optimize import linprog
# Solve the problem by Simplex method in Optimization
res = linprog(c, A_ub=A, b_ub=b, bounds=(x0_bounds, x1_bounds), method='simplex', options={"disp": True})
print res
Optimization terminated successfully.
Current function value: -36.000000
Iterations: 3
con: array([], dtype=float64)
fun: -36.0
message: 'Optimization terminated successfully.'
nit: 3
slack: array([2., 0., 0.])
status: 0
success: True
x: array([2., 6.])