Optimization

Prob:2.2

N Chakradhar[1] Havish[2]

[1]EE16BTECH11022

[2]EE16BTECH11023

February 28, 2019

Problem 2.2

- Solve

$$\max_x f(x) = 6x_1 + 5x_2 \tag{1}$$

with constraints

$$x_1 + x_2 \leq 5 \tag{2}$$

$$3x_1 + 2x_2 \leq 12 \tag{3}$$

$$x_1, x_2 \geq 0 \tag{4}$$

using tabular method

Solution

Convert the given problem into standard form by using slack variables. After this, our problem looks like:

$$\max_{x} f(x) = 6x_1 + 5x_2 \tag{5}$$

with constraints

$$x_1 + x_2 + x_3 = 5 \tag{6}$$

$$3x_1 + 2x_2 + x_4 = 12 \tag{7}$$

$$x_1, x_2, x_3, x_4 \geq 0 \tag{8}$$

Solution

This problem is same as solving the following set of equations

$$-6x_1 - 5x_2 + p = 0 \tag{9}$$

$$x_1 + x_2 + x_3 = 5 \tag{10}$$

$$3x_1 + 2x_2 + x_4 = 12 \tag{11}$$

$$x_1, x_2, x_3, x_4 \geq 0 \tag{12}$$

The approach used here is similar to Gauss-Jordan method of solving equations. Basically, we would like to express the given objective function only in terms of slack variables.

Solution

The initial table will look like the following:

	x_1	x_2	x_3	x_4	p	RHS
x_3	1	1	1	0	0	5
x_4	3	2	0	1	0	12
p	-6	-5	0	0	1	0

After the first set of transformations(choosing the pivot element and doing row operations), the table is as follows:

	x_1	x_2	x_3	x_4	p	RHS
x_3	0	1/3	1	-1/3	0	1
x_1	1	2/3	0	1/3	0	4
p	0	-1	0	2	1	24

Solution

We end the problem once we get all non-negative numbers in the bottom row. So, after the final transformation, the table looks like:

	x_1	x_2	x_3	x_4	p	RHS
x_3	0	1	3	-1	0	3
x_1	1	0	-2	1	0	2
p	0	0	3	1	1	27

The solution will be $x_1 = 2$, $x_2 = 3$ and the maximum is$p = 27$.

Code

```
from cvxpy import *
from numpy import matrix
A = matrix([ [1.0, 1.0], [3.0, 2.0 ]])
b = matrix([ 5.0, 12.0 ])
c = matrix([ 6.0, 5.0 ])
x = Variable((2,1),nonneg=True)
f = c*x
obj = Maximize(f)
constraints = [A*x ≤ b.transpose()]
Problem(obj, constraints).solve()
(x1,x2) = x.value
print("Maximum value is " + str(f.value[0][0]))
print("x1: " + str(x1[0]))
print("x2: " + str(x2[0]))
```

Output

Maximum value: 27.00000000000000
x1: 2.000000000000000
x2: 3.0