Tutorial- 5:

Non linear Programming using Python

Min $x_1x_4\ (x_1+x_2+x_3)+x_3$

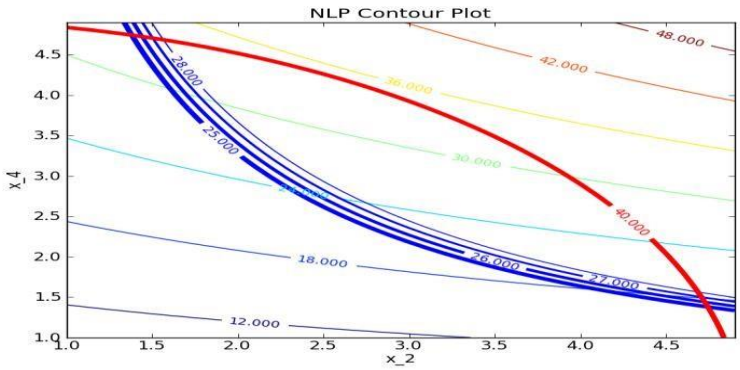
s.t. $x_1x_2x_3x_4 \geq 25$

$x_{21}+x_{22}+x_{23}+x_{24}=40$

$1 \leq x1,x2,x3,x4 \leq 5$

$x_0=(1,5,5,1)$

This problem has a nonlinear objective that the optimizer attempts to minimize. The variable values at the optimal solution are subject to (s.t) both equality(=40) and inequality (>25) constraints. The product of the four variables must be greater than 25 while the sum of squares of the variables must also equal 40. In addition, all variables must be between 1 and 5 and the initial guess is $x_1 = 1$, $x_2 = 5$, $x_3 = 5$, and $x_4 = 1$.



For this problem determine:

1. A potential feasible solution
2. Identify the constraints on the contour plot
3. Mark the set of feasible solutions on the contour plot
4. Identify the minimum objective feasible solution

5. Identify the maximum objective feasible solution
6. Use a nonlinear programming solver to find a solution

Example 2:

maximise $2x_1 + x_2 - 5\log_e(x_1)\sin(x_2)$
subject to
$x_1x_2 <= 10$
$| x_1 - x_2 | <= 2$
$0.1 <= x_1 <= 5$
$0.1 <= x_2 <= 3$

Here we have a nonlinear objective and nonlinear constraints. Using the package we have the input:

Maximize	2*x1 + x2 - 5*log(x1)*sin(x2)
C1	x1*x2 <= 10
C2	abs(x1-x2) <= 2
X1	>=0.1, <=5
X2	>=0.1, <=3

and it clear from this there there are values of x_1 and x_2 that exceed the supposed maximum objective function value of 8.8166 given by the package.

CODE

```
import numpy as np
from scipy.optimize import minimize

def objective(x):
  return x[0]*x[3]*(x[0]+x[1]+x[2])+x[2]
def constraint1(x,sign=1.0):
  return x[0]*x[1]*x[2]*x[3]-25.0
def constraint2(x,sign=1.0):
  sum_eq = 40.0
  for i in range(4):
      sum_eq = sum_eq - x[i]**2
  return sum_eq

#initial guess
n = 4
x0 = np.zeros(n)
```

```
x0[0] = 1.0
x0[1] = 5.0
x0[2] = 5.0
x0[3] = 1.0

print('Initial SSE Objective:'+str(objective(x0)))
```

#output:- `Initial SSE Objective:16.0`

```
b=(1.0,5.0)
bnds=(b,b,b,b)
con1={'type':'ineq','fun': constraint1}
con2={'type':'eq','fun': constraint2}
cons=([con1,con2])
solution=minimize(objective,x0,method='SLSQP',bounds=bnds,constraints=c
ons)
x=solution.x
print('Final SSE Objective:' + str(objective(x)))
```

#output:- `Final SSE Objective:17.01401724563517`

```
print('Solution')
print('x1='+str(x[0]))
print('x2='+str(x[1]))
print('x3='+str(x[2]))
print('x4='+str(x[3]))
```

#output:-

```
Solution
x1=1.0
x2=4.742996096883977
x3=3.8211546234095715
x4=1.379407645075325
```