Untitled

February 13, 2020

[1]:
```
# I'm looking for 25 bonus points
# Created by Alisher Siddikov

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

[2]:
```
# problem 1 - Fertilizer

## Objective: 4x1 + 2x2 - 0.5x1^2 - 0.25x2^2

## Constraints:
##8,000 + 5,000 <= 40,000
```

[3]:
```
def fun(x, sign = 1.0):
    return sign*(4*x[0] + 2*x[1] - 0.5*x[0]**2 - 0.25*x[1]**2)

### test
x = [3.16, 2.95]
fun(x)
```

[3]:
```
11.371574999999998
```

[4]:
```
cons = ({'type': 'ineq', 'fun': lambda x: 40000 - (8000*x[0] + 5000*x[1])})

bnds = ((0, 40000), (0, 40000))
```

[5]:
```
x = [2, 2] #initial values
res = minimize(fun,x, args=(-1.0,),method='SLSQP',\
                    bounds=bnds,constraints=cons)
res
```

[5]:
```
     fun: -11.36842105263158
     jac: array([-0.84210539, -0.52631581])
 message: 'Optimization terminated successfully.'
    nfev: 16
     nit: 4
    njev: 4
  status: 0
 success: True
```

```
       x: array([3.15789472, 2.94736845])
```

[6]:
```
print('objective = ', round(res.fun, 4))

print('x1 = ', round(res.x[0],4))
print('x2 = ', round(res.x[1],4))
```

```
objective =  -11.3684
x1 =  3.1579
x2 =  2.9474
```

[7]:
```
# problem 2 - Fencing

## a side of triangle length
## b side of triangle length
## c side of triangle length

## Objective: s = x*0.5; SQRT(s*(s-a)*(s-b)*(s-c))

## Constraints:
## a + b + c <= 60
# -a + b + c <= 60
## a - b + c <= 60
## a + b - c <= 60
```

[8]:
```
def fun(x, sign = 1.0):
    s = np.sum(x) * 0.5
    return sign*(sqrt(s * (s - x[0]) * (s - x[1]) * (s - x[2])))

### test
x = [20.0, 20.0, 20.0]
fun(x)
```

[8]:
```
173.20508075688772
```

[9]:
```
cons = ({'type': 'ineq', 'fun': lambda x: 60 -( x[0] + x[1] + x[2])},
        {'type': 'ineq', 'fun': lambda x: 60 -(-x[0] + x[1] + x[2])},
        {'type': 'ineq', 'fun': lambda x: 60 -( x[0] - x[1] + x[2])},
        {'type': 'ineq', 'fun': lambda x: 60 -( x[0] + x[1] - x[2])})

bnds = ((0, 60), (0, 60), (0, 60))
```

[10]:
```
x = [2, 2, 2] #initial values
res = minimize(fun,x, args=(-1.0,),method='SLSQP',\
                   bounds=bnds,constraints=cons)
res
```

[10]:
```
     fun: -173.2050807569032
     jac: array([-5.77350235, -5.77350426, -5.77350426])
 message: 'Optimization terminated successfully.'
```

```
     nfev: 20
      nit: 4
     njev: 4
   status: 0
  success: True
        x: array([20.        , 20.        , 19.99999999])
```

[11]:
```
print('objective = ', round(res.fun, 4))

print('a = ', round(res.x[0],4))
print('b = ', round(res.x[1],4))
print('c = ', round(res.x[2],4))
print('perimeter = ', round(np.sum(res.x),4))
```

```
objective =  -173.2051
a =  20.0
b =  20.0
c =  20.0
perimeter =  60.0
```