REPUBLIQUE DU BENIN
*****
MINISTERE DES ENSEIGNEMENTS SECONDAIRES TECHNIQUES ET DE LA FORMATION PROFESSIONNELLE
****
UNIVERSITE D'ABOMEY CALAVI
ECOLE NATIONALE D'ECONOMIE APPLIQUEE ET MANAGEMENT

FILIERE : Informatique de Gestion (IG2-A) **GROUPE** : N°1

TP de Groupe : Recherche Opérationnelle

Membres du groupe

1- ASSOGBA Ange-Marie
2- ADEGBITE Aristide
3- AGNIDE Akim
4- GANSOU Eddy
5- **MOUSTAPHA Abdul Hafiz**
6- TCHANHOUIN Aurel

Sous la supervision de

Mr HOUNDJI Ratheil

ANNEE ACADEMIQUE : 2023-2024

Page 1 → Exercice 1 ***** Par TCHANHOUIN Aurel

FORMULATION :

‘’

```
- min -((1/5xA + 12)/2 + (2/5xB + 12.5)/2 + (3/5xC + 13.4)/2)/3
           xA +             xB +                xC                 <= 100
        1/5xA                                                      <= 20
                          2/5xB                                    <= 20
                                              3/5xC                <= 20
           xA  ,            xB,                xC                 >= 0
```

‘’

CODE :

‘’

```
from scipy.optimize import linprog

c = [-1/30, -2/30, -3/30]
A = [[1,1,1],[1/5,0,0],[0,2/5,0],[0,0,3/5]]
b = [100,20,20,20]
x1_bounds = (0, None)
x2_bounds = (0, None)
x3_bounds = (0,None)

res = linprog(c, A_ub=A, b_ub=b, bounds=[x1_bounds,x2_bounds,x3_bounds])

print(f'Z* = {-(res.fun - (12/2+12.5/2+13.4/2)/3)}\n X* = {res.x}\n
{res.message}')
```

‘’

Page 1 → Exercice 2 ***** Par ADEGBITE Aristide

FORMULATION :

‘’

```
max(6x + 6y +6z)
     x + 6y +  3z <= 84
    3x +  y +  3z <= 42
     x + 3y +  2z <= 21
    2x + 3y +  4z <= 42
     x,   y,    z  >=  0
```

‘’

CODE :

‘’

```
from scipy.optimize import linprog
C =[-6, -6, -6]
A = [[1,6,3],[3,1,3],[1,3,2],[2,3,4]]
B = [84,42,21,42]
a =(0,None)
b =(0,None)
c =(0,None)

res = linprog(C, A_ub=A, b_ub=B , bounds=[a,b,c])
#res *=-1
print(f"\nLe cout optimal est Z* = {-res.fun} \nLa solution optimale est X* =
{res.x}")
print(res.message)
```

‘’

Page 1 → Exercice 5 ***** Par ASSOGBA Ange-Marie

FORMULATION :

‘’

```
min c1x1 + c2x2 + C3x3
      x1 +   x2         >= 1
      x1 +  2x2         <= 3
      x1,    x2,    x3  >= 0

Transformons les contraintes d'inégalité supérieure en celles inférieure

min c1x1 + c2x2 + C3x3
     -x1 -  -x2         <= -1
      x1 +  2x2         <=  3
      x1,    x2,    x3  >=  0
```

```
→→→Pour c=(1, 0, 1) on a:
min   x1 +          x3
     -x1 -  -x2         <= -1
      x1 +  2x2         <=  3
      x1,    x2,    x3  >=  0


→→→Pour c=(0, 1, 0) on a:
min          x2
     -x1 -  -x2         <= -1
      x1 +  2x2         <=  3
      x1,    x2,    x3  >=  0


→→→Pour c=(0, 0, 1) on a:
min                 x3
     -x1 -  -x2         <= -1
      x1 +  2x2         <=  3
      x1,    x2,    x3  >=  0
```

‘’

CODE :

‘’

```
from scipy.optimize import linprog

c = [1, 0, 1]
a = [[-1, -1, 0], [1, 2, 0]]
b = [-1, 3]
x1 = (0, None)
x2 = (0, None)
x3 = (0, None)
res = linprog(c, A_ub=a, b_ub=b, bounds=[x1, x2, x3])
print("→→Pour c=(1, 0, 1)")
print(f"Cout optimal : {res.fun}")
print(f"Solutions optimales : {res.x}")
print(res.message, "\n")

d = [0, 1, 0]

res1 = linprog(d, A_ub=a, b_ub=b, bounds=[x1, x2, x3])
print("→→Pour c=(0, 1, 0)")
print(f"Cout optimal : {res1.fun}")
```

```
print(f"Solutions optimales : {res1.x}")
print(res1.message, "\n")

g = [0, 0, 1]

res2 = linprog(g, A_ub=a, b_ub=b, bounds=[x1, x2, x3])
print("→→Pour c=(0, 0, 1)")
print(f"Cout optimal :{res2.fun}")
print(f"Solutions optimales : {res2.x}")
print(res2.message, "\n")
```

‘’

Page 6 → Exercice 2 ***** Par MOUSTAPHA Abdul Hafiz

FORMULATION :

‘’

```
Min 4,5x1 + 7,8x2 + 3,6x3 + 3,1x4 + 4,9y1 + 7,2y2 + 4,3y3 + 2,9y4
       x1 +    x2 +    x3 +    x4                                  = 2
                                  y1 +    y2 +    y3 +    y4 = 2
       x1 +                       y1                          = 1
               x2 +                       y2                  = 1
                       x3 +                       y3          = 1
                               x4 +                       y4 = 1

       x1 ;    x2 ;    x3 ;    x4 ;    y1 ;    y2 ;    y3 ;    y4 £ {0, 1}
```

‘’

CODE :

‘’

```
from scipy.optimize import linprog
c= [4.5, 7.8, 3.6, 3.1, 4.9, 7.2, 4.3, 2.9]
A = [
        [1, 0, 0, 0, 1, 0, 0, 0],
        [0, 1, 0, 0, 0, 1, 0, 0],
        [0, 0, 1, 0, 0, 0, 1, 0],
        [0, 0, 0, 1, 0, 0, 0, 1],
```

```
        [1, 1, 1, 1, 0, 0, 0, 0],
        [0, 0, 0, 0, 1, 1, 1, 1],
    ]
b = [
        1,
        1,
        1,
        1,

        2,
        2
]
x1_bounds = (0, None)
x2_bounds = (0, None)
x3_bounds = (0, None)
x4_bounds = (0, None)
y1_bounds = (0, None)
y2_bounds = (0, None)
y3_bounds = (0, None)
y4_bounds = (0, None)

res = linprog(c, A_eq=A, b_eq=b, bounds=[x1_bounds, x2_bounds, x3_bounds,
x4_bounds, y1_bounds, y2_bounds, y3_bounds, y4_bounds ], integrality=1)
print(f'Z* = {res.fun} \nX* = {res.x} \n{res.message}')
```

‘’

Page 6 → Exercice 4 ***** Par AGNIDE Akim

FORMULATION :

‘’

```
Max (10x1 + 15x2)
    (0.02x1 + 0.04x2) <= 700
    x1 et x2 sont des variables entières
```

‘’

CODE :

‘’

```
from scipy.optimize import linprog

print("CAS DE CHOIX 1 : ")
c = [-10,-15]
A = [[0.02,0.04]]
b = [500]
x1_bounds = (0,None)
x2_bounds = (0,None)

resulta = linprog(c,A_ub=A,
b_ub=b,bounds=[x1_bounds,x2_bounds],integrality=1)
resultaCorrigee=-resulta.fun
print("COÛT OPTIMALE : ", resultaCorrigee)
print("SOLUTION OPTIMALE : ",resulta.x)
print("MESSAGE DE TRAVAIL 1 : ",resulta.message)


print("\nCAS DE CHOIX 2 : ")

c = [-10,-15]
A = [[0.02,0.04]]
b = [700]
x1_bounds = (0,None)
x2_bounds = (0,None)

resulta = linprog(c,A_ub=A,b_ub=b,bounds=[x1_bounds,x2_bounds],integrality=1)

print("COÛT OPTIMALE 2 : ", -resulta.fun)
print("SOLUTION OPTIMALE : ",resulta.x)
print("MESSAGE DE TRAVAIL 2 : ",resulta.message)
```

‘’

Page 6 → Exercice 5 ***** Par GANSOU Eddy

FORMULATION :

‘’

```
Max 17x1 + 10x2 +  25x3  +  7x4
     5x1 +  3x2 +   8x3 +   7x4   < 12
      x1                          € {0,1}
             x2                   € {0,1}
                     x3           € {0,1}
                             x4   € {0,1}
```

‘’

CODE :

‘’

```
from scipy.optimize import linprog
c = [-17, -10, -25, -7]
A = [[5, 3, 8, 7]]
b = [12]
x1_bounds = (0, 1)
x2_bounds = (0, 1)
x3_bounds = (0, 1)
x4_bounds = (0, 1)

res = linprog(c,A_ub=A,b_ub=b, bounds=[x1_bounds,x2_bounds, x3_bounds,
x4_bounds], integrality=1)
solution = res.x
cout = - res.fun
print(f'Z* = {cout} \nX* = {solution} \n{res.message}')
```

‘’

CONCLUSION

Les travaux se sont déroulés comme attendu grâce à la participation de tous les membres. Aucune difficulté particulière n’a été rencontrée lors des travaux.