lin_prog_example

September 11, 2019

[ ]:

```
# src: https://www.kaggle.com/mchirico/linear-programming
```

[9]:

```
import matplotlib.pyplot as plt
```

[1]:

```
# scipy.linprog
# https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.
 ↪html
```

[2]:

```
# Consider the following problem:
# Minimize: f = -1x[0] + 4x[1]
# Subject to:
# -3x[0] + 1x[1] <= 6
# 1x[0] + 2x[1] <= 4
# x[1] >= -3
# -inf <= x[0] <= +inf

c = [-1, 4]  # minimize function
A = [ [-3, 1], [1, 2] ]  # subject to 1, 2
b = [6, 4]  # subject to 1, 2 (free term)

x0_bounds = (None, None)
x1_bounds = (-3, None)
```

[7]:

```
from scipy.optimize import linprog

result = linprog(
    c,  # coeffs of the linear objective function to be minimized
    A_ub=A,  # inequality constraint matrix: coeffs of a linear inequality
    b_ub=b,  # inequality constraint vector: upper bound A_ub @ x
    bounds=(x0_bounds, x1_bounds),  # sequence of (min, max) pairs for each␣
 ↪element in x
    options={'disp': True}
)

display(result)
```

```
Primal Feasibility  Dual Feasibility    Duality Gap         Step
Path Parameter      Objective
```

```
1.0                 1.0                 1.0                 -                1.0
-8.0
0.09885158404625    0.09885158404625    0.09885158404625    0.903461537018
0.09885158404625    -6.284698425658
0.05788429348353    0.05788429348355    0.05788429348355    0.4273037994111
0.05788429348355    -7.864724729573
0.04539867008243    0.04539867008244    0.04539867008244    0.2387091287399
0.04539867008244    -12.78916804766
0.00666151448168    0.006661514481681   0.006661514481682   0.8665142913493
0.006661514481682   -21.3520715063
6.299626472829e-06  6.299626472597e-06  6.299626472583e-06  1.0
6.299626472568e-06  -21.99681708159
3.150184161152e-10  3.150192895998e-10  3.150192773305e-10  0.9999499939736
3.150193133197e-10  -21.99999984082
Optimization terminated successfully.
         Current function value: -22.000000
         Iterations: 6

     con: array([], dtype=float64)
     fun: -21.99999984082497
 message: 'Optimization terminated successfully.'
     nit: 6
   slack: array([3.89999997e+01, 8.46872172e-08])
  status: 0
 success: True
       x: array([ 9.99999989, -2.99999999])
```

[14]:

```
# Example 2

# A trading company is looking for a way to maximize profit per transportation␣
 ↪of their goods.
# The company has a train available with 3 wagons.
# When stocking the wagons they can choose between 4 types of cargo, each with␣
 ↪its own specifications.
# How much of each cargo type should be loaded on which wagon in order to␣
 ↪maximize profit?

data_matrix = [['Train Wagon', 'Item Capacity', 'Space Capacity'],
               ['w1', 10, 5000],
               ['w2', 8, 4000],
               ['w3', 12, 8000],]

data_matrix_2 = [['Cargo<br>Type', '#Items Available', 'Volume','Profit'],
               ['c1', 18, 400,2000],
               ['c2', 10, 300,2500],
               ['c3', 5, 200,5000],
               ['c4', 20, 500,3500]]
```

```
# Objective function
# max: +2000 C1 +2500 C2 +5000 C3 +3500 C4 +2000 C5 +2500 C6 +5000 C7 +3500 C8␣
 ↪+2000 C9 +2500 C10 +5000 C11 +3500 C12;
# Flip sign above to get MIN PROBLEM

# Constraints
# +C1 +C2 +C3 +C4 <= 10;
# +C5 +C6 +C7 +C8 <= 8;
# +C9 +C10 +C11 +C12 <= 12;
# +400 C1 +300 C2 +200 C3 +500 C4 <= 5000;
# +400 C5 +300 C6 +200 C7 +500 C8 <= 4000;
# +400 C9 +300 C10 +200 C11 +500 C12 <= 8000;
# +C1 +C5 +C9 <= 18;
# +C2 +C6 +C10 <= 10;
# +C3 +C7 +C11 <= 5;
# +C4 +C8 +C12 <= 20;

# What if we get rid of item constraint?
# Change min to max
c = [-2000,-2500,-5000,-3500,-2000,-2500,-5000,-3500,-2000,-2500,-5000,-3500]
xb=[]
for i in range(0,12):
    xb.append((0, None))

A = [
     [400,300,200,500,0,0,0,0,0,0,0,0,],
     [0,0,0,0,400,300,200,500,0,0,0,0,],
     [0,0,0,0,0,0,0,0,400,300,200,500],
     [1,0,0,0,1,0,0,0,1,0,0,0],
     [0,1,0,0,0,1,0,0,0,1,0,0],
     [0,0,1,0,0,0,1,0,0,0,1,0],
     [0,0,0,1,0,0,0,1,0,0,0,1],
    ]

b = [5000,4000,8000,18,10,5,20]

res = linprog(c, A_ub=A, b_ub=b, bounds=xb,
              options={"disp": True})
print(res)
```

```
Primal Feasibility  Dual Feasibility    Duality Gap         Step
Path Parameter      Objective
1.0                 1.0                 1.0                 -                1.0
-39000.0
0.01857376342384    0.01857376342407    0.01857376342394    0.9814325002781
0.01857376342401    -41923.6805552
```

```
0.008273681833716   0.008273681833816   0.00827368183376    0.6023022624057
0.008273681833793   -57685.84657308
0.001369502404791   0.001369502404933   0.001369502404924   0.850031657623
0.001369502405595   -104680.9016415
2.177560547685e-05  2.177560547873e-05  2.177560547858e-05  0.9866690285217
2.177560548592e-05  -134394.9097331
1.238954819161e-09  1.238954871486e-09  1.238954870803e-09  0.9999431604847
1.238956052819e-09  -134999.965423
6.2835484587e-14    6.259364057527e-14  6.259349674501e-14  0.9999494786712
6.202546866317e-14  -134999.9999983
Optimization terminated successfully.
         Current function value: -134999.999998
         Iterations: 6
     con: array([], dtype=float64)
     fun: -134999.99999825243
 message: 'Optimization terminated successfully.'
     nit: 6
   slack: array([6.86113708e-08, 4.87984835e-08, 1.26165105e-07, 1.05000000e+01,
       1.14406262e-10, 2.61639599e-11, 3.10620862e-10])
  status: 0
 success: True
       x: array([ 2.46339402,  3.24582181,  1.67878867,  5.41027623,
2.22657249,
        2.82995238,  1.64300888,  3.86356703,  2.81003349,  3.92422581,
        1.67820245, 10.72615674])
```