Optimisation

Hedia smida 1er gt3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ex1 :

import pulp as p

Lp\_prob = p.LpProblem('Problem', p.LpMaximize)

# Create problem Variables

x = p.LpVariable("x", *lowBound* = 0)   # Create a variable x >= 0

y = p.LpVariable("y", *lowBound* = 0)   # Create a variable y >= 0

# Objective Function

Lp\_prob += 40 \* x + 50 \* y

# Constraints:

Lp\_prob += 2 \* x +  y <= 800

Lp\_prob += x + 2 \* y <= 700

Lp\_prob += y <= 300

# Display the problem

print(Lp\_prob)

status = Lp\_prob.solve()   # Solver

print(p.LpStatus[status])   # The solution status

# Printing the final solution

print(p.value(x), p.value(y), p.value(Lp\_prob.objective))

import pulp as p

Lp\_prob = p.LpProblem('Problem', p.LpMaximize)

# Create problem Variables

x = p.LpVariable("x", *lowBound* = 0)   # Create a variable x >= 0

y = p.LpVariable("y", *lowBound* = 0)   # Create a variable y >= 0

# Objective Function

Lp\_prob += 40 \* x + 50 \* y

# Constraints:

Lp\_prob += 2 \* x +  y <= 800

Lp\_prob += x + 2 \* y <= 700

Lp\_prob += y <= 300

# Display the problem

print(Lp\_prob)

status = Lp\_prob.solve()   # Solver

print(p.LpStatus[status])   # The solution status

# Printing the final solution

print(p.value(x), p.value(y), p.value(Lp\_prob.objective))

resultat :

Une image contenant texte

Description générée automatiquement

Ex2: (fabrication)

import pulp as p

Lp\_prob = p.LpProblem('Problem', p.LpMaximize)

# Create problem Variables

x = p.LpVariable("x", *lowBound* = 0)   # Create a variable x >= 0

y = p.LpVariable("y", *lowBound* = 0)   # Create a variable y >= 0

# Objective Function

Lp\_prob += 10 \* x + 18 \* y

# Constraints:

Lp\_prob += 2 \* x + 3 \* y <= 100

Lp\_prob += 0.4 \* x + 0.6 \* y <= 700

# Display the problem

print(Lp\_prob)

status = Lp\_prob.solve()   # Solver

print(p.LpStatus[status])   # The solution status

# Printing the final solution

print(p.value(x), p.value(y), p.value(Lp\_prob.objective))

résultat :

Une image contenant texte

Description générée automatiquement

Exer 3: (chaise)

import pulp as p

Lp\_prob = p.LpProblem('Problem', p.LpMaximize)

# Create problem Variables

a = p.LpVariable("a", *lowBound* = 0)   # Create a variable x >= 0

b = p.LpVariable("b", *lowBound* = 0)   # Create a variable y >= 0

# Objective Function

Lp\_prob += 450 \* a + 800 \* b

# Constraints:

Lp\_prob += 1.5 \* a +  2 \* b <= 250

Lp\_prob += 0.5 \* a + 0.75 \* b <= 100

Lp\_prob += 2 \* a + 3 \* b <= 300

Lp\_prob += a >= 100

Lp\_prob += b >= 53

Lp\_prob += b <= 99

Lp\_prob += a <= 99

# Display the problem

print(Lp\_prob)

status = Lp\_prob.solve()   # Solver

print(p.LpStatus[status])   # The solution status

# Printing the final solution

print(p.value(a), p.value(b), p.value(Lp\_prob.objective))

Résultat :

Une image contenant texte

Description générée automatiquement