MCSD1133 Operations Research & Optimization

CHAPTER 2 (Part 3): LINEAR PROGRAMMING WITH PYTHON

nzah@utm.my-20232024(S1)

Innovating Solutions

UTM JOHOR BAHRU

Introduction

- A good and popular programming language recommended by many in the OR and Data Science communities is Python. It is easy, flexible, and powerful, and has great libraries for Machine Learning, Optimization, and Statistical Modelling.
- Many optimization solvers (commercial and open-source) have Python interfaces for modelling LP.
- **PuLP** is a powerful open-source modelling package in Python.

PuLP

- PuLP is a library for the Python scripting language that enables users to describe mathematical programs.
- PuLP works entirely within the syntax and natural idioms of the Python language by providing Python objects that represent optimization problems and decision variables, and allowing constraints to be expressed in a way that is very similar to the original mathematical expression.
- To keep the syntax as simple and intuitive as possible, PuLP has focused on supporting linear and mixed-integer models.

PuLP

- PuLP modeling process has the following steps for solving LP problems:
    1) Initialize Model
    2) Define Decision Variable
    3) Define Objective Function
    4) Define the Constraints
    5) Solve Model

Step 1: Initialize Model

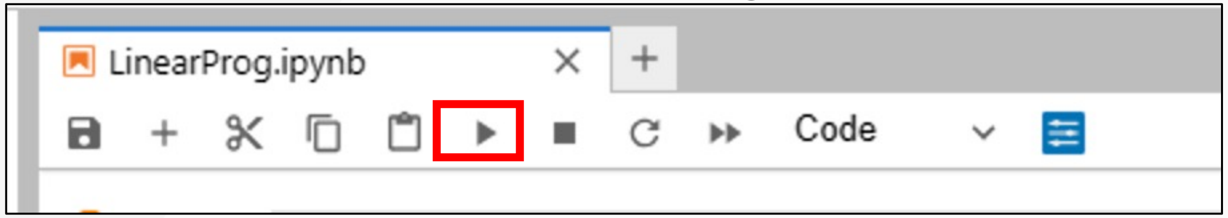
To start using PuLP, we need to tell Python to use the library.

- Install

  The easiest way to install pulp is using pip.

```
#Install PuLP modeler function
pip install pulp
```

  then restart the kernel using



- Import:

```
In [ ]:  # Import PuLP modeler functions
         from pulp import *
```

Step 2: Define Decision Variables

Example: Reddy Mikk Company

Variables:

X_1 = tons produced daily of exterior paint

X_2 = tons produced daily of interior paint

Z = total daily profit (in thousands of dollars)

Objective: Maximize, $Z = 5X_1 + 4X_2$

Constraints:

1. $6X_1 + 4X_2 \leq 24$ (raw material M1)
2. $X_1 + 2X_2 \leq 6$ (raw material M2)
3. $X_2 - X_1 \leq 1$
4. $X_2 \leq 2$
5. $X_1 \geq 0;\ X_2 \geq 0$

Step 2: Define Decision Variables (cont'd)

Create the variable to contain the problem data. Install using **LpProblem** function.

```
In [ ]:  # Create the 'prob' variable to contain the problem data
         prob = LpProblem("Reddy",LpMaximize)
```

It has two parameters:

i) The arbitrary name of this problem (as a string)

ii) Second parameter being either **LpMinimize** or **LpMaximize** depending on the type of LP to solve.

Step 2: Define Decision Variables (cont'd)

- The problem variables X_1 and X_2 are create using **LpVariable** class.

```
In [ ]:  # The 2 variables  are created with a lower limit of zero
         x1=LpVariable("Exterior",0)
         x2=LpVariable("Interior",0)
```

- It has four parameters,
  1. arbitrary name of what this variable represents.
  2. the lower bound on this variable (the bounds can be entered directly as a number, or None as the default)
  3. upper bound
  4. the fourth is essentially the type of data (discrete or continuous) function. With the default as **Lpcontinous**.
- If the first few parameters are entered and the rest are ignored , they take their default values.

Step 3: Define Objective Function

- Define the objective function by adding it to the variable *prob* with +- operator.
- The objective function is logically entered first, with an important comma , at the end of the statement and a short string explaining what this objective function is.
- The objective function is added to *prob* first.

```
In [ ]:
# The objective function is added to 'prob' first
prob += 5*x1 + 4*x2, "Total daily profit"
```

Step 4: Define Constraints

- “non-negative” constraints were already included when defining the variables.
- Add constraints (**+=**) operator again to add more data to the variable prob.

```
# The four constraints are entered
prob += 6*x1 + 4*x2 <= 24.0, "Raw Material M1"
prob += 1*x1 + 2*x2 <= 6.0, "Raw Material M2"
prob += x2 - x1 <= 1.0, "DifferentDemandRequirement"
prob += x2 <= 2, "MaxInteriorDemand"
```

Step 4: Define Constraints (cont'd)

- **WriteLP** function can be used to copy the information into a .lp file into the directory that the code-block is running from.
- There is no assignment operator (such as an equal's sign) on this line. This is because the function/method WriteLP called is being performed to the variable/object prob.
- The dot . between the variable/object and the function/method is important and is seen frequently in Object Oriented software

```
# The problem data is written to an .lp file
prob.writeLP("Reddy.lp")
```

Step 5: Solve Model

- The LP is solved using the solver that PuLP chooses. The input brackets after solve( )are left empty in this case, however they can be used to specify which solver to use.

```
# The problem is solved using PuLP's choice of Solver
prob.solve()
```

Step 5: Solve Model (cont'd)

- Now the results of the solver call can be displayed as output.
- Request the status of the solution, which can be one of “Not Solved”, “Infeasible”, “Unbounded”, “Undefined” or “Optimal”.
- The value of *prob* is returned as an integer, which must be converted to its significant text meaning using the **LpStatus dictionary.**

```
# The status of the solution is printed to the screen
print ("Status:", LpStatus[prob.status])
```

Step 5: Solve Model (cont'd)

- The variables and their resolved optimum values can now be printed to the screen.
- The for loop is used to called all the problem variable names.
- Prints each variable name, followed by an equal's sign, followed by its optimum value.
- The optimised objective function value is printed to the screen, using the value (**prob.objective**) function.

```
# Each of the variables is printed with it's resolved optimum value
for v in prob.variables():
    print(v.name, "=", v.varValue)

# The optimised objective function value is printed to the screen
print ("Total Profit = ", value(prob.objective))
```

Step 5: Solve Model (cont'd)

Output

```
Status: Optimal
Exterior = 3.0
Interior = 1.5
Total Profit =  21.0
```

Another option to print the output:

```
print("Number of tons produced daily of  exterior paint: ", x1.varValue)
print(" Number of tons produced daily of  interior : ", x2.varValue)
print("Total Profit: ", value(problem.objective))
```

EXERCISES

Use Python to solve the following problem.

#1: The Alex Garment Company manufactures men's shirts and women's blouses. The production process includes cutting, sewing, and packaging. The company employs 25 workers in the cutting department, 35 in the sewing department, and 5 in the packaging department. The factory works one 8-hr shift, 5 days a week. The following table gives the time requirements and profits per unit to produce the two garments. **Determine the optimal** weekly production schedule for Alex Garment Company.

	Minutes per unit			
Garment	*Cutting*	*Sewing*	*Packaging*	Unit profit ($)
Shirts	20	70	12	8
Blouses	60	60	4	12

#2: In Hamid grocery store, shelf space is limited and must be used effectively to increase profit. Two cereal items, Grano and Wheatie, compete for a total shelf space of 60 ft^2 . A box of Grano occupies 0.2 ft^2 and a box of Wheatie needs 0.4 ft^2 . The maximum daily demands of Grano and Wheatie are 200 and 120 boxes, respectively. A box of Grano nets $1.00 in profit and a box of Wheatie $1.35. Hamid thinks that because the unit profit of Wheatie is 35% higher than that of Grano, Wheatie should be allocated 35% more space than Grano, which amounts to allocating about 57% to Wheatie and 43% to Grano. **Determine the optimal** value for the items to be allocated on the shelf to maximise the profit.

EXERCISES

Use Python to solve the following problem.

#3: A firm has two bottling plant. One plant located at Coimbatore and other plant located at Chennai. Each plant produces three types of drinks; Coca-Cola , Fanta and Thumps-up. The following table show the data.

	Number of bottles produced per day by plant at	
Product	Coimbatore	Chennai
Coca-Cola	15000	15000
Fanta	30000	10000
Thumps-Up	20000	50000
Cost per day (per unit)	600	400

Market survey indicates that during the month of April there will be a demand of 200,000 bottles of Coca-cola , 400,000 bottles of Fanta and 440,000 bottles of Thumps-up. For how many days each plant be run in April to **minimize the production cost**, while still meeting the market demand?