**Practical No. 14**

**A.1 Aim:**

To solve the fixed charge problem.

**A.2 Prerequisite:**

Python, Jupiter notebook, demo in xlsx

**A.3 Outcome:**

The program is written in Python solving fixed charge problem.

**A.4 Theory:**

To be discussed during the lecture hours.

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per the following segments within four hours of the practical. The soft copy must be uploaded on the portal or emailed to the concerned lab in charge faculties at the end of the practical in case there is no portal access available)***

|  |  |
| --- | --- |
| Roll No.: DS2211 | Name: Precia Bhagyaraj |
| Class: MSc (Data Science) | Batch: Part 2 |
| Date of Experiment: 27.02.24 | Date of Submission: 27.02.24 |
| Grade: | |

**B.1 Task to do:**

Implement the functionality shown in the video to solve the fixed charge problem

[(13) Fixed Charge Problem - YouTube](https://www.youtube.com/watch?v=5Ari9WHczd4)

**B.2 Output program**

import numpy as np

# Coefficients of the constraints

coefficients = np.array([[2, 3, 6], [6, 3, 4], [5, 6, 2]])

# Constants of the constraints

constants = np.array([600, 300, 400])

# Calculate maximum values for x1, x2, and x3

x1\_max = min(constants[0] / coefficients[0])

x2\_max = min(constants[1] / coefficients[1])

x3\_max = min(constants[2] / coefficients[2])

print("Maximum value for x1:", x1\_max)

print("Maximum value for x2:", x2\_max)

print("Maximum value for x3:", x3\_max)

**B.3 Equations:**

Maximize = 48X1 + 55X2 + 50X3

-1000Y1 – 800Y2 – 900Y3

Subject to,

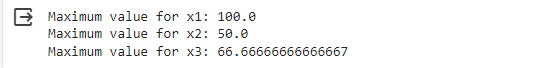
Resource constraints:

Machining: 2X1 + 3X2 + 6X3 <= 600

Grinding: 6X1 + 3X2 + 4X3 <= 300

Assembly: 5X1 + 6X2 + 2X3 <= 400

**B.4 Conclusion:**

******