CS69011: Computing Lab
Assignment 5: Linear Programming (Part - A)

September 6, 2023

==

1. Regarding submission: Create separate Python file(s): **<RollNo>_Q1.py, <RollNo>_Q2.py.**
2. Create a .zip file containing the two Python file(s) with the name: **<RollNo>_A5_Part_A.zip** and submit it to Moodle.
3. The input to the program will be available in a .txt file given as **command line arguments**.
4. The final output for the program needs to be stored in a separate .txt file as 'Summary_Q1.txt' for Q1 and 'Summary_Q2.txt' for Q2.
5. Feel free to modify the problem to suit your needs and implement the linear programming optimization using libraries like `ortools`, `SciPy`, or others that provide LP solvers, **but you need to restrict yourselves to using only LP solvers to solve this problem.**

==

Q 1: Basic Production Planning

Consider a manufacturing company that produces a set of products using various resources. You are the production manager and your goal is to optimize the production plan to maximize profit. You are given **N** products $\{p_1, p_2, \ldots, p_N\}$ that add a profit margin of $\{P_1, P_2, \ldots, P_N\}$ respectively to the company. Due to limited funding, you can acquire only **M** resources $\{R_1, R_2, \ldots, R_M\}$. You can manufacture only a u_{ij} number of units for product **i** and resource **j**.

1. **Input Format:**
   - The first line contains the number of products, `N`.
   - The second line contains the number of resources, `M`.
   - The third line contains `N` space-separated numbers denoting the profit per unit for each product.
   - The fourth line contains `M` space-separated numbers denoting the availability of each resource.
   - Then `N' lines follow.
   - Each line has 'M' space-separated numbers denoting the resource usage (consumption) for each product.

2. **Output:**
   - Display the optimal production plan for each product, along with the maximum achievable profit.

3. **Sample Input:**

3
2
100 150 200
300 200
2 1
1 2
3 2

4. **Sample Output:**
Optimal production plan found:
Product 0: Quantity = 100.0
Product 1: Quantity = 50.0
Product 2: Quantity = 0.0
Maximum Profit: 25000.0

Q 2: Production Planning with Production Capacity Constraints

Consider a manufacturing company that produces a set of products using various resources. You are the production manager and your goal is to optimize the production plan to maximize profit. You are given **N** products $\{p_1, p_2, \ldots., p_N\}$ that add a profit margin of $\{P_1, P_2, \ldots., P_N\}$ respectively to the company. Due to limited funding, you can acquire only **M** resources $\{R_1, R_2, \ldots., R_M\}$. You can manufacture only a u_{ij} number of units for product **i** and resource **j**. Due limited marketing budget, the company has decided now that it can only sell $\{x_1, x_2, \ldots, x_N\}$ units for the products $\{p_1, p_2, \ldots., p_N\}$ respectively.

1. **Input Format:**
   - The first line contains the number of products, `N`.
   - The second line contains the number of resources, `M`.

- The third line contains `N` space-separated numbers denoting the profit per unit for each product.
- The fourth line contains `M` space-separated numbers denoting the availability of each resource.
- The fifth line contains 'N' space-separated numbers denoting the maximum production capacity for each product.
- Then `N' lines follow.
- Each line has 'M' space-separated numbers denoting the resource usage (consumption) for each product.

2. **Output:**
   - Display the optimal production plan for each product, considering maximum production capacities, along with the maximum achievable profit.

3. **Sample Input**

3
2
100 150 200
300 200
2 1
1 2
3 2
150 100 200

4. **Sample output:**

Optimal production plan found:
Product 0: Quantity = 75.0
Product 1: Quantity = 0.0
Product 2: Quantity = 50.0
Maximum Profit: 17500.0

****

Algorithm: Basic Production Planning

- Formulate the production planning problem as a linear programming problem.
- Define decision variables for the quantity of each product to produce.
- Create constraints to ensure that resource consumption does not exceed availability.
- Define the objective function to maximize the total profit.

Pseudo-code for Basic Production Planning Optimization:

Input:
- Number of products P
- Number of resources R
- Profit per unit for each product (profit[P])
- Availability of each resource (availability[R])
- Resource usage for each product and resource (usage[P][R])

Define:
- Decision variables: quantity[P] (quantity of each product to produce)

Create a linear programming model:
- Create a solver

For each product p in P:

    Define the quantity of product p as a non-negative integer variable

Create constraints:
- For each resource r in R:
    - Sum of (quantity[p] * usage[p][r]) <= availability[r]
- For each product p in P:
    - quantity[p] >= 0 (non-negativity constraint)

Define the objective function:
- Maximize: sum of (profit[p] * quantity[p])

Solve the linear programming problem:
- Call solver.Solve()

If the solver finds an optimal solution:

    Display the optimal production plan (quantity[p] for each product) and the maximum profit

Else:

    Display an appropriate message

****

Algorithm: Production Planning with Constraints

- Extend the basic production planning algorithm to incorporate maximum production capacities.
- Create constraints to ensure that the production quantity for each product does not exceed its capacity.

Pseudo-code for Production Planning with Constraints:

Input:
- Number of products P
- Number of resources R
- Profit per unit for each product (profit[P])
- Availability of each resource (availability[R])
- Resource usage for each product and resource (usage[P][R])
- Maximum production capacity for each product (max_capacity[P])

Define:
- Decision variables: quantity[P] (quantity of each product to produce)

Create a linear programming model:
- Create a solver

For each product p in P:

    Define the quantity of product p as a non-negative integer variable

Create constraints:
- For each resource r in R:
    - Sum of (quantity[p] * usage[p][r]) <= availability[r]
- For each product p in P:
    - quantity[p] >= 0 (non-negativity constraint)
    - quantity[p] <= max_capacity[p] (capacity constraint)

Define the objective function:
- Maximize: sum of (profit[p] * quantity[p])

Solve the linear programming problem:
- Call solver.Solve()

If the solver finds an optimal solution:

Display the optimal production plan (quantity[p] for each product) and the maximum profit

Else:

Display an appropriate message