Problem Definition: Multi-Objective Optimization in a Production System

Objective

We aim to **minimize both the total production cost and carbon emissions** in a factory while satisfying resource constraints. This is a **multi-objective convex optimization problem**, where we explore the trade-off between cost and emissions.

Objective Functions:

1. **Minimize Total Cost**:

$$\text{Total Cost} = 50x_1 + 60x_2$$

   - x_1: Cost per unit of product 1 = $50
   - x_2: Cost per unit of product 2 = $60

2. **Minimize Total Carbon Emissions**:

$$\text{Total Emissions} = 5x_1 + 25x_2$$

   - x_1: Emissions per unit of product 1 = 5 kg CO_2
   - x_2: Emissions per unit of product 2 = 25 kg CO_2

Constraints:

1. **Labor Constraint**:

$$3x_1 + 5x_2 \leq 150$$

   - Each unit of x_1 requires 3 hours of labor.
   - Each unit of x_2 requires 5 hours of labor.
   - Total labor available is 150 hours.

2. **Raw Material Constraint**:

$$2x_1 + 4x_2 \leq 120$$

   - Each unit of x_1 requires 2 units of raw material.
   - Each unit of x_2 requires 4 units of raw material.
   - Total raw material available is 120 units.

3. **Minimum Production Constraint**:

$$x_1 + x_2 \geq 10$$

   - At least 10 units of products must be produced in total.

4. **Bounds**:

$$0 \leq x_1 \leq 30, \quad 0 \leq x_2 \leq 20$$

   - x_1 can be produced in quantities between 0 and 30.
   - x_2 can be produced in quantities between 0 and 20.

Approach to Solve the Problem

The problem is solved using the **Weighted Sum Method**, which is a common technique for multi-objective optimization. Here's the step-by-step approach:

---

1. **Combine Objectives into a Single Objective Function**

Since we have two objectives (minimizing cost and minimizing emissions), we combine them into a single objective function using a weight w:

$$\text{Objective Function} = w \cdot \text{Total Cost} - (1 - w) \cdot \text{Total Emissions}$$

- w: A weight between 0 and 1 that controls the trade-off between cost and emissions.
  - When $w = 1$, the solver prioritizes minimizing cost.
  - When $w = 0$, the solver prioritizes minimizing emissions.

2. **Solve the Linear Programming Problem**

For each value of w, we solve the following linear programming problem using the `linprog` function from `scipy.optimize`:

$$\text{Minimize: } w \cdot (50x_1 + 60x_2) - (1 - w) \cdot (5x_1 + 25x_2)$$

Subject to:

$$3x_1 + 5x_2 \leq 150$$

$$2x_1 + 4x_2 \leq 120$$

$$x_1 + x_2 \geq 10$$

$$0 \leq x_1 \leq 30, \quad 0 \leq x_2 \leq 20$$

---

3. **Generate the Pareto Front**

- We vary w from 0 to 1 in small increments (e.g., 20 steps).
- For each w, we solve the linear programming problem and record the optimal values of x_1 and x_2, as well as the corresponding total cost and total emissions.
- The set of all optimal solutions forms the **Pareto Front**, which represents the trade-off between cost and emissions.

Brief Explanation of the Code

Libraries Used:

- **numpy (as np):**
  - Used for numerical operations, such as creating arrays and performing vectorized calculations.
  - **Example:** `np.array`, `np.linspace`, `np.dot`.
- **matplotlib.pyplot (as plt):**
  - Used for plotting the Pareto Front.
  - **Example:** `plt.plot`, `plt.xlabel`, `plt.ylabel`, `plt.show`.
- **scipy.optimize.linprog:**
  - Used to solve linear programming problems.
  - Minimizes a linear objective function subject to linear inequality and equality constraints.

Key Functions and Their Roles:

- **np.array:**
  - Defines the coefficients for the objective functions (cost and emissions) and constraints.
- **np.linspace:**
  - Generates 20 evenly spaced weights (**w**) between 0 and 1 for the weighted sum method.
- **linprog:**
  - Solves the linear programming problem for each weight **w**.
  - **Inputs:** Objective function coefficients (`c`), inequality constraints (`A_ub`, `b_ub`), and variable bounds (`bounds`).
  - **Outputs:** Optimal values of $\mathbf{x_1}$ and $\mathbf{x_2}$, and the status of the solution.
- **np.dot:**
  - Computes the total cost and emissions by taking the dot product of coefficients and decision variables.
- **plt.plot:**
  - Plots the Pareto Front, showing the trade-off between cost and emissions.
- **plt.show:**
  - Displays the plot.

How the Code Works:

1. **Problem Setup:**
   - Define the cost and emission coefficients, constraints, and bounds for $\mathbf{x_1}$ and $\mathbf{x_2}$.
2. **Weighted Sum Method:**
   - For each weight **w**, combine the cost and emission objectives into a single objective function.
   - Solve the linear programming problem using `linprog`.
3. **Store Results:**
   - Record the optimal values of $\mathbf{x_1}$, $\mathbf{x_2}$, total cost, and total emissions for each weight.
4. **Plot Pareto Front:**
   - Plot the valid solutions to visualize the trade-off between cost and emissions.
5. **Display Results:**
   - Print the optimal solutions for each weight **w**.

Why These Libraries/Functions?

- **numpy:** Efficiently handles numerical computations and array operations.
- **matplotlib:** Provides a simple interface for creating visualizations.
- **scipy.optimize.linprog:** Solves linear programming problems efficiently, which is essential for optimization tasks.

Results:

w=0.00: x1=16.67, x2=20.00, Cost=2033.33, Emissions=583.33

w=0.05: x1=16.67, x2=20.00, Cost=2033.33, Emissions=583.33

w=0.11: x1=0.00, x2=20.00, Cost=1200.00, Emissions=500.00

w=0.16: x1=0.00, x2=20.00, Cost=1200.00, Emissions=500.00

w=0.21: x1=0.00, x2=20.00, Cost=1200.00, Emissions=500.00

w=0.26: x1=0.00, x2=20.00, Cost=1200.00, Emissions=500.00

w=0.32: x1=0.00, x2=10.00, Cost=600.00, Emissions=250.00

w=0.37: x1=0.00, x2=10.00, Cost=600.00, Emissions=250.00

w=0.42: x1=0.00, x2=10.00, Cost=600.00, Emissions=250.00

w=0.47: x1=0.00, x2=10.00, Cost=600.00, Emissions=250.00

w=0.53: x1=0.00, x2=10.00, Cost=600.00, Emissions=250.00

w=0.58: x1=0.00, x2=10.00, Cost=600.00, Emissions=250.00

w=0.63: x1=0.00, x2=10.00, Cost=600.00, Emissions=250.00

w=0.68: x1=10.00, x2=0.00, Cost=500.00, Emissions=50.00

w=0.74: x1=10.00, x2=0.00, Cost=500.00, Emissions=50.00

w=0.79: x1=10.00, x2=0.00, Cost=500.00, Emissions=50.00

w=0.84: x1=10.00, x2=0.00, Cost=500.00, Emissions=50.00

w=0.89: x1=10.00, x2=0.00, Cost=500.00, Emissions=50.00

w=0.95: x1=10.00, x2=0.00, Cost=500.00, Emissions=50.00

w=1.00: x1=10.00, x2=0.00, Cost=500.00, Emissions=50.00