Optimization of Production in an Industrial Plant

Background: In an industrial plant, several machines are used to manufacture products. Each machine has a limited capacity, and each product requires specific resources (e.g., material, time, energy) for production. The goal is to maximize profits while ensuring that the total consumption of resources (machine time, energy, material, etc.) does not exceed the available resources, and the machines operate within their capacities.

Problem Setup:

- We have a set of machines, each with a limited capacity for production.
- There are several products, each requiring a certain amount of machine time, material, and energy to produce.
- The objective is to maximize the profit, which depends on the number of units of each product produced, while respecting the capacity constraints of machines and resource availability.

Given Data:

1. **Machine Data:**
    - **Capacity of each machine**: Each machine has a limited number of hours it can operate in a given time period (e.g., 10 hours).
2. **Product Data:**
    - **Profit per unit of each product**: This represents the revenue or profit earned for each unit of a product produced.
    - **Resource consumption**: Each product consumes a certain amount of machine time, material, and energy.
3. **Objective:** Maximize total profit, considering the costs of production and constraints of resources.

Constraints:

1. The total machine time required by all products should not exceed the available time for each machine.
2. The total amount of material and energy used by all products should not exceed the available resources.

Example: Optimization Problem

Let's assume the following scenario:

- **Machines**:
    - Machine 1: 10 hours available.
    - Machine 2: 12 hours available.
- **Products**:
    - **Product 1**:
        - Profit per unit: $10
        - Time required on Machine 1: 2 hours
        - Time required on Machine 2: 3 hours
        - Material required: 5 units
        - Energy required: 3 units
    - **Product 2**:
        - Profit per unit: $15
        - Time required on Machine 1: 3 hours
        - Time required on Machine 2: 2 hours
        - Material required: 4 units
        - Energy required: 2 units
    - **Product 3**:
        - Profit per unit: $20
        - Time required on Machine 1: 1 hour
        - Time required on Machine 2: 4 hours
        - Material required: 6 units
        - Energy required: 4 units
- **Available Resources**:
    - Material: 50 units
    - Energy: 30 units

Objective:

Maximize the total profit by determining the optimal number of units of each product to produce, considering the constraints on machine time, material, and energy.

Solution Approach:

We will use **linear programming** (LP) to solve this problem. Here are the steps involved:

1. **Decision Variables**: Let:
   - x_1 = number of units of Product 1 produced.
   - x_2 = number of units of Product 2 produced.
   - x_3 = number of units of Product 3 produced.
2. **Objective Function**: The objective is to maximize profit. The profit is given by:

$$\text{Maximize} \quad Z = 10x_1 + 15x_2 + 20x_3$$

   where:
   - $10x_1$ is the profit from Product 1,
   - $15x_2$ is the profit from Product 2,
   - $20x_3$ is the profit from Product 3.
3. **Constraints**:
   - **Machine time constraints**:
     - For Machine 1: $2x_1 + 3x_2 + 1x_3 \leq 10$ (Machine 1 has 10 hours available).
     - For Machine 2: $3x_1 + 2x_2 + 4x_3 \leq 12$ (Machine 2 has 12 hours available).
   - **Material constraint**: $5x_1 + 4x_2 + 6x_3 \leq 50$ (50 units of material available).
   - **Energy constraint**: $3x_1 + 2x_2 + 4x_3 \leq 30$ (30 units of energy available).
   - **Non-negativity**: $x_1, x_2, x_3 \geq 0$.

Problem Statement: Boiler Optimization

The boiler system needs to optimize coal consumption for generating steam, minimizing operational costs, while satisfying the steam demand and meeting environmental constraints. Specifically, the system needs to generate **10,000 kg/hr of steam** at a pressure of **15 bar**. The energy produced by burning 1 kg of coal is **8,000 kcal**, and the combustion process requires **5 kg of air per kg of coal** and generates **2.5 kg of flue gases per kg of coal**.

The objective is to **minimize the fuel consumption** while ensuring the following constraints are met:

1. **Steam generation requirement**: The boiler must produce 10,000 kg/hr of steam.
2. **Air supply**: For every 1 kg of coal, 5 kg of air is required.
3. **Emissions control**: The system must limit the generation of flue gases.
4. **Energy balance**: The energy produced by coal combustion must meet the energy demand for steam generation.

Decision Variables:

- x_1: Amount of coal (in kg/hr) to be burned.
- x_2: Amount of air (in kg/hr) to be supplied.

Objective Function:

Minimize the total coal consumption, i.e., minimize x_1.

Constraints:

1. **Energy constraint**: The energy produced by coal combustion must meet or exceed the energy required for steam generation:

$$8000 \times x_1 = 10,000 \times \text{Energy required for steam generation}$$

(Energy required for steam generation can be approximated based on the latent heat of vaporization or other factors).

2. **Air supply constraint**: The amount of air needed is directly proportional to the amount of coal burned:

$$x_2 = 5 \times x_1$$

3. **Flue gas constraint**: The flue gases produced by burning coal should be controlled:

$$2.5 \times x_1 \leq \text{Maximum permissible flue gases}$$

Solution Approach:

We will solve the problem using both **linear optimization** (for the simpler part of the problem) and **global optimization** (to handle any complexities such as non-linearity or multiple local optima).

Linear Optimization Solution (using scipy.optimize.linprog):

In the linear optimization approach, we will simplify the problem by treating the energy constraint as linear, assuming a direct relationship between coal burned and steam output. We will use linear programming to minimize the amount of coal used, while adhering to the air and flue gas constraints.

See: boiler-linear-optimzation.py

Global Optimization Solution (using scipy.optimize.differential_evolution):

In case the problem involves non-linear relationships or more complex constraints, we can use a global optimization method such as **Differential Evolution**, which is a metaheuristic that can handle non-linearity.

See: boiler-multi-optimization.py

Notes:

1. **Linear Optimization Solution**:
    - We define the objective function to minimize coal input.
    - The constraints represent the energy balance (coal energy vs. steam requirement), the relationship between coal and air, and the flue gas generation constraint.
    - The linprog function is used to solve this linear optimization problem.
2. **Global Optimization Solution**:
    - In the global optimization approach, we define an objective function that penalizes violating constraints (energy and flue gas limits).
    - The differential_evolution function is used, which is a global optimization method capable of finding a good solution even in the presence of non-linearity or complex constraints.

Boiler System Details:

Boiler input dataset:

`time`	Timestamp in milliseconds since the epoch.	Input
`APH-B O/L O2`	Outlet oxygen level from Air Preheater-B.	Input
`AH A OUTLET FGS OXYGEN CONTENT`	Outlet oxygen content from Air Heater-A.	Input
`AH INLET FGS OXYGEN CONTENT`	Inlet oxygen content to Air Heater.	Input
`APH-A PRIMARY AIR I/L TEMP`	Primary air inlet temperature to Air Preheater-A.	Input
`LSIDE ECONOMIZER OUTLET FGS TEMP`	Outlet flue gas temperature from Left-Side Economizer.	Input
`CHIMNEY FGS CO2 CONTENT`	CO2 content of flue gas at the chimney.	Input
`FDF A OUTLET SA TEMP.`	Secondary air temperature at the outlet of FDF-A.	Input
`AH B INLET FGS CO CONTENT`	CO content of flue gas at the inlet of Air Heater-B.	Input
`SECOND AIR FLOW TOTAL`	Total secondary air flow.	Input
`WTHR_RH_prc`	Weather relative humidity percentage.	Input
`Main Steam Flow`	Flow rate of main steam.	Output
`PRIMARY AIR FLOW`	Flow rate of primary air.	Input
`WTHR_PRS_pa`	Weather atmospheric pressure in Pascals.	Input
`WTHR_TEMP_degC`	Weather temperature in degrees Celsius.	Input
`AH A OUTLET FGS TEMP.`	Outlet flue gas temperature from Air Heater-A.	Input
`dates`	Human-readable timestamp.	Input

Coal dataset:

Feature Name	Definition	Input/Output
`time`	Timestamp in milliseconds since the epoch.	Input
`Sulphur in Fuel`	Percentage of sulfur content in the fuel.	Input
`Moisture in Fuel`	Percentage of moisture content in the fuel.	Input
`Hydrogen In Fuel`	Percentage of hydrogen content in the fuel.	Input
`GCV`	Gross calorific value of the fuel (kcal/kg).	Input
`Carbon In Fuel`	Percentage of carbon content in the fuel.	Input
`JSW_AshInFuel.MAN`	Percentage of ash content in the fuel.	Input
`Nitogen In Fuel`	Percentage of nitrogen content in the fuel.	Input
`Oxygen In Fuel`	Percentage of oxygen content in the fuel.	Input
`Unburnt carbon in bed ash`	Percentage of unburnt carbon in bed ash.	Output
`Unburnt carbon in fly ash`	Percentage of unburnt carbon in fly ash.	Output
`dates`	Human-readable timestamp.	Input

O2 dataset:

Feature Name	Definition	Input/Output
`BF GAS FLOW BEFORE FGTV`	Blast furnace gas flow rate before FGTV (Flue Gas Treatment Vessel).	Input
`COAL FLOW TOTAL`	Total flow rate of coal.	Input
`Main Steam Flow`	Flow rate of main steam.	Output
`AH INLET FGS OXYGEN CONTENT`	Oxygen content in flue gas at the inlet of Air Heater.	Input
`PRIMARY AIR FLOW`	Flow rate of primary air.	Input
`Steam To Fuel Ratio`	Ratio of steam generated to fuel used.	Input
`GasCV`	Calorific value of gas.	Input
`Air to Fuel Ratio`	Ratio of air to fuel.	Input
`SEC AIR FLOW B`	Secondary air flow for path B.	Input
`SEC AIR FLOW A`	Secondary air flow for path A.	Input
`Bucket`	Bucket category indicating operational grouping.	Input
`O2 category`	Category based on oxygen levels (e.g., "Low", "High").	Output

Boiler output dataset:

Feature Name	Definition	Input/Output
`time`	Timestamp in milliseconds since the epoch.	Input
`Loss Due To H2 O In Fuel`	Heat loss due to water content in the fuel.	Output
`Loss Due To Dry Flue Gas`	Heat loss due to the dry components of flue gas.	Output
`Loss Total`	Total heat loss from the boiler.	Output
`Loss Due To H2 In Fuel`	Heat loss due to hydrogen content in the fuel.	Output
`Loss Due To H2 O In Air`	Heat loss due to water vapor in the air.	Output
`Boiler Efficiency`	Overall efficiency of the boiler (%).	Output
`Loss Due To Unburnt Carbon`	Heat loss due to unburnt carbon in the fuel.	Output
`Loss Due To Partial Combustion`	Heat loss due to incomplete combustion.	Output
`Loss Due To Radiation`	Heat loss due to radiation and convection.	Output
`dates`	Human-readable timestamp.	Input

Overall Picture of the Datasets

1. Boiler Input Dataset

- Primary Inputs: Variables related to oxygen levels, temperatures, airflows, weather conditions, and chimney gases.
- Primary Outputs: None directly; acts as input data for other datasets to analyze system performance.

2. Coal Dataset

- Primary Inputs: Characteristics of the fuel (moisture, sulfur, carbon, hydrogen, etc.).
- Primary Outputs: Percentages of unburnt carbon in bed ash and fly ash, indicating combustion quality.

3. O2 Dataset

- Primary Inputs: Air and fuel flow parameters, oxygen content, and related performance metrics.
- Primary Outputs: Oxygen category and indirect indications of system efficiency.

4. Boiler Output Dataset

- Primary Inputs: None directly (dependent on the above datasets).
- Primary Outputs: Losses in various forms (e.g., due to hydrogen in fuel, radiation, etc.), boiler efficiency, and total losses.

Final Structure:

Dataset	Primary Purpose	Inputs	Outputs
Boiler Input Dataset	Operating conditions and ambient parameters.	Temperatures, airflows, weather data	Feeds into other datasets.
Coal Dataset	Fuel characteristics influencing combustion.	Sulfur, moisture, GCV, etc.	Combustion quality indicators.
O2 Dataset	Air and fuel dynamics affecting combustion.	Airflows, oxygen content, fuel ratios	Oxygen category and efficiency.
Boiler Output Dataset	Efficiency and loss analysis for boiler performance.	Combustion indicators and system outputs	Boiler efficiency and losses.

Additional work:

Align Input-Output Mapping:

- Ensure that all key outputs in one dataset feed into another logically. For example:
  - Combustion quality (coal dataset outputs) should be linked to total losses and efficiency (boiler output dataset).
  - Air and fuel ratios (O2 dataset) should influence combustion outputs in the boiler input and output datasets.

Incorporate Derived Features:

- Add derived features such as:
  - Excess air ratio: Derived from air-to-fuel ratio and oxygen content.
  - Flue gas heat recovery potential: Using outlet temperatures and flow rates.
- These metrics provide better insight into boiler performance.

Handle Redundancy:

- Some datasets have overlapping features, e.g., Main Steam Flow appears in both the boiler input and O2 datasets. It is better to centralize such features to avoid duplication and potential inconsistencies.

Output Dataset Specificity:

- Add more detailed outputs, such as:
    - Total emissions (CO_2, NOx, SOx) for environmental impact analysis.
    - Ash content segregation to study slagging and fouling.

Integrate Feedback Loops:

- Boiler systems often have feedback loops, e.g., outputs like Boiler Efficiency or O2 category directly influence control parameters like Primary Air Flow and Secondary Air Flow.

Temporal Resolution Consistency:

- Ensure that the timestamps across datasets align precisely to facilitate time-series analysis.

WORK:

1. **Exploratory Data Analysis**
2. **Airflow Modulation Analysis in Boiler Systems**
3. **Optimization Problem: Ensuring Optimal Combustion Quality with Varying Coal Properties**