Problem Statement: 07

In a 1000 MW boiler system, optimizing the boiler efficiency is crucial to minimize fuel consumption and reduce emissions. Boiler efficiency depends on several key parameters, including:

1. **Oxygen content in flue gas (O2)**: This is a critical indicator of combustion efficiency. Maintaining an optimal oxygen level ensures complete combustion while avoiding excessive energy loss due to high excess air.
2. **Carbon dioxide content in flue gas (CO2)**: High CO2 content in the flue gas indicates better combustion of carbon in fuel, which is desirable for efficiency.
3. **Flue gas temperature (T_fg)**: A higher flue gas temperature indicates heat loss in exhaust gases, which reduces efficiency.
4. **Unburnt carbon in ash (U_C)**: This represents incomplete combustion and reduces overall boiler efficiency.

The goal is to formulate a convex equation for **Boiler Efficiency (η)** based on these parameters and optimize it by varying the oxygen content (O2) while ensuring operational constraints.

Mathematical Analysis:

The boiler efficiency equation can be approximated as:

$$\eta = 100 - (L_{\text{dry flue gas}} + L_{\text{unburnt carbon}} + L_{\text{radiation}})$$

Where:

1. $L_{\text{dry flue gas}} = K_1 \cdot (T_{fg} - T_{\text{ambient}})/\text{GCV}$
   Represents the heat loss due to dry flue gas, dependent on flue gas temperature.
2. $L_{\text{unburnt carbon}} = K_2 \cdot U_C$
   Represents the heat loss due to unburnt carbon in ash.
3. $L_{\text{radiation}} = K_3$
   A constant representing radiation loss.

Given constants:

- $K_1 = 0.5$
- $K_2 = 0.8$
- $K_3 = 1.5$
- $\text{GCV (Gross Calorific Value)} = 25000\,\text{kJ/kg}$
- $T_{\text{ambient}} = 30^\circ\ \text{C}$

Constraints:

- $3\% \le O_2 \le 7\%$ (to ensure complete combustion)
- T_{fg} is measured in $120^\circ\ \text{C} \le T_{fg} \le 180^\circ\ \text{C}$
- U_C is measured in $0.5\% \le U_C \le 1.5\%$

The negative value in the optimization function is used because the optimization process in many libraries, like scipy.optimize.minimize, **by default minimizes a function**. If you want to maximize something, you can turn it into a minimization problem by negating the function.

Why?

If we want to **maximize the boiler efficiency (η\etaη)**, we cannot directly use scipy.optimize.minimize because it minimizes by nature. To handle this:

- The boiler efficiency function is negated, so the optimizer minimizes the **negative efficiency**.

- Mathematically:

$$\text{maximize } \eta \quad \text{is equivalent to minimizing } \; -\eta$$

Boiler Efficiency Equation:

The boiler efficiency η is given by:

$$\eta = 100 - (L_{\text{dry flue gas}} + L_{\text{unburnt carbon}} + L_{\text{radiation}})$$

Where:

1. $L_{\text{dry flue gas}} = K_1 \cdot \frac{(T_{fg} - T_{\text{ambient}})}{\text{GCV}}$
2. $L_{\text{unburnt carbon}} = K_2 \cdot U_C$
3. $L_{\text{radiation}} = K_3$ (a constant)

Thus:

$$\eta = 100 - \left[K_1 \cdot \frac{(T_{fg} - T_{\text{ambient}})}{\text{GCV}} + K_2 \cdot U_C + K_3\right]$$

We are optimizing over three variables:

- O_2 (oxygen content in flue gas)
- T_{fg} (flue gas temperature)
- U_C (unburnt carbon in ash)

The radiation loss K_3 is constant and does not affect convexity.

1. **Convex Functions** are crucial in optimization because:
   - A local minimum is always a global minimum.
   - They are easier to solve due to guaranteed convergence of optimization algorithms.
2. **Concave Functions** are useful when maximizing:
   - A local maximum is always a global maximum.
   - By negating the function, concave problems can be turned into convex problems (e.g., maximizing profit).