CH4020
Project

Aaryan, CO21BTECH11001

Question 1

Solution:

Reactions are

$$A \rightarrow B \quad Rate = k_1$$

$$B \rightarrow C \quad Rate = k_2$$

Rate equations are

$$\frac{dC_A}{dt} =- (k_1 C_A + k_2 C_B)$$

$$\frac{dC_B}{dt} = k_1 C_A$$

$$\frac{dC_c}{dt} = k_2 C_A$$

Solving for rate equations assuming initial concentration of A to be C_0

We get

$$C_A = C_0 exp(- (k_1 + k_2)t)$$

$$C_B = \frac{k_1}{k_1+k_2}(C_0 - C_0 exp(- (k_1 + k_2)t) = \frac{k_1}{k_1+k_2}(C_0 - C_A)$$

$$C_c = \frac{k_2}{k_1+k_2}(C_0 - C_0 exp(- (k_1 + k_2)t) = \frac{k_2}{k_1+k_2}(C_0 - C_A)$$

Let us suppose that N values of C_A^i, C_B^i, C_C^i are given in the data file.

Define cost function as mean squared error of actual and approx values.

$$cost(k_1, k_2) = \sum_{i=1}^{N} [(C_A^i - C_A)^2 + (C_B^i - C_B)^2 + (C_C^i - C_C)^2]$$

Now, we can calculate optimal values of k_1, k_2 using $scipy.optimize.fmin$ function by passing cost function and initial guess value.
The reason for that is because k_1, k_2 do not have any constraint or bound.

—---

Question 2

Solution:

Himmelblau function is:

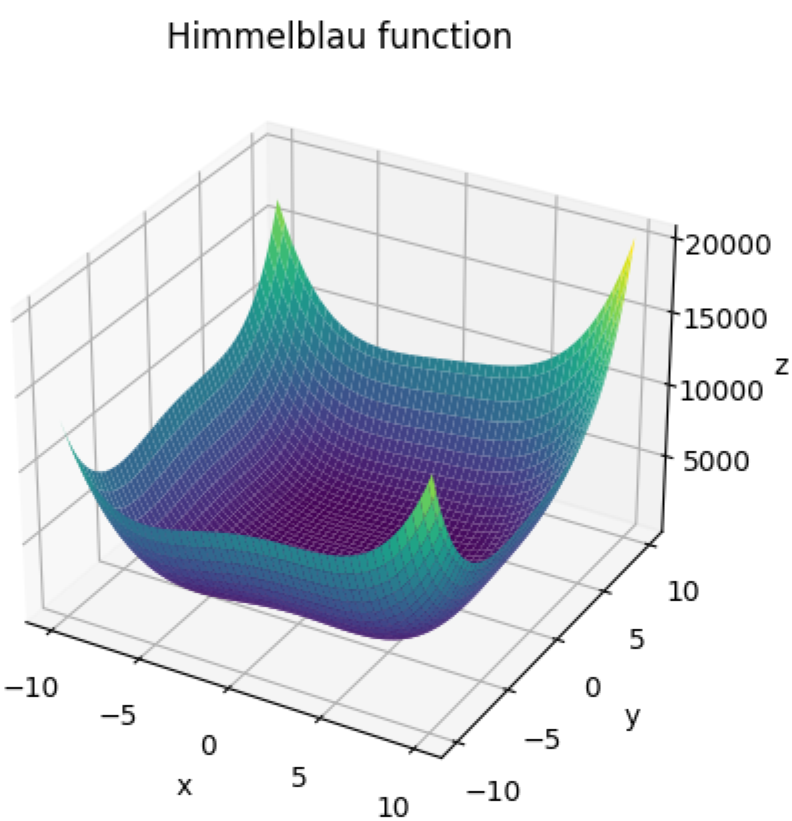
$$f(x_1, x_2) = min_{x_1, x_2}(x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

Given constraint:

$$x_1^2 + x_2^2 \geq 25$$

Given bounds:

$$-5 \leq x_1, x_2 \leq 5$$



a. <u>Using deterministic algorithm:</u>

We can calculate optimal value of x_1, x_2 by $scipy.optimize.minimize$ function and passing function, initial guess, constraints, bounds and setting method as $SLSQP$.

Since the Himmelblau function has multiple minimum values, we have to take an appropriate guess value so that the constraint is satisfied at the optimal point (solution).

By trying different guess values, I arrived at a guess value of $(-4.0, -4.0)$.

The optimal point found is $x_1 = -3.779$, $x_2 = -3.283$, which satisfies all constraints and bounds.

b. <u>Using genetic algorithm:</u>

A class named $GenAlgo$ is created which initializes the objective, bounds and constraints.
Then, the module GA of package $pymoo$ is used by passing population size as 100 and optimal value of x_1, x_2 is calculated by using $pymoo.optimize.minimize$ function.

The optimal point found is $x_1 = -3.779$, $x_2 = -3.283$, which satisfies all constraints and bounds.