PRACTICAL: 3

**Aim:** Let’s use it to optimize a simple linear equation with 4 inputs and 1 output.

Y = w1X1 + w2X2 + w3X3 + w4X4

We want to get the values of w1 to w4 to make the following equation hold: Y = w1(4) + w2(-2) + w3(3.5) + w4(5)

Implement the above simple linear equation using pygad library.

**Code:**

import pygad import numpy as np

import matplotlib.pyplot as plt

inputs = np.array([4, -2, 3.5, 5])

target\_output = 44

def fitness\_func(ga\_instance, solution, solution\_idx): output = np.sum(solution \* inputs)

fitness = 1.0 / (abs(output - target\_output) + 0.0001) return fitness

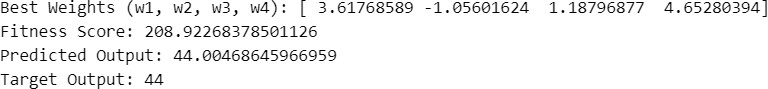
ga\_instance = pygad.GA(

num\_generations=100, # number of generations num\_parents\_mating=2, # parents for mating fitness\_func=fitness\_func, # our custom fitness function sol\_per\_pop=10, # population size num\_genes=len(inputs), # number of weights to optimize mutation\_probability=0.2, # chance of mutation mutation\_type="random", # mutation operator mutation\_percent\_genes=50 # % of genes to mutate

)

ga\_instance.run()

solution, solution\_fitness, solution\_idx = ga\_instance.best\_solution() print("Best Weights (w1, w2, w3, w4):", solution)



print("Fitness Score:", solution\_fitness)

predicted\_output = np.sum(solution \* inputs) print("Predicted Output:", predicted\_output) print("Target Output:", target\_output)

plt.plot(ga\_instance.best\_solutions\_fitness, label="Best Fitness") plt.xlabel("Generation")

plt.ylabel("Fitness")

plt.title("Genetic Algorithm Fitness Over Generations") plt.legend()

plt.grid(True) plt.show()

**Output:**

