1.

import numpy as np

from scipy.optimize import differential\_evolution

defobjective\_function(x):

return -10 \* np.cos(np.pi \* x - 2.2) + (x + 1.5) \* x

bounds = [(-10, 10)]

result = differential\_evolution(objective\_function, bounds)

min\_x = result.x

global\_min\_val = result.fun

print("global min x: ",min\_x)

print("Global Optimal Solution:")

print(f"x = {min\_x[0]}")

print(f"f(x) = {global\_min\_val}")

2.

import numpy as np

import matplotlib.pyplot as plt

# Define the function f(x)

def objective\_function(x):

return -10 \* np.cos(np.pi \* x - 2.2) + (x + 1.5) \* x

# Generate x values

x = np.linspace(-5, 5, 20)

print(x)

y = objective\_function(x)

print(y)

plt.plot(x, y, label='f(x) = -10Cos(pi x - 2.2) + (x + 1.5) \* x')

plt.xlabel('x')

plt.ylabel('f(x)')

plt.title(' Function f(x)')

plt.grid(True)

min\_y = min(y)

min\_x = x[np.argmin(y)]

plt.scatter(min\_x, min\_y, color='blue', label=f'Minimum: ({min\_x}, {min\_y})')

plt.legend()

plt.show()

print("Global optimal solution is", min\_x)

print("Optimal function value is”, min\_y)

3.