Lab 4

Cuckoo Search (CS) algorithm

code:

import numpy as np

def objective\_function(x):

return np.sum(x\*\*2)

nests = 10 # Number of nests

dimension = 2 # Number of dimensions

iterations = 50 # Number of iterations

discovery\_rate = 0.25 # Probability of worst nests being abandoned

population = np.random.uniform(-5, 5, (nests, dimension))

best\_solution = population[0]

best\_fitness = objective\_function(best\_solution)

for iteration in range(iterations):

new\_population = population + np.random.randn(nests, dimension)

for i in range(nests):

fitness = objective\_function(new\_population[i])

if fitness < objective\_function(population[i]):

population[i] = new\_population[i]

if fitness < best\_fitness:

best\_fitness = fitness

best\_solution = new\_population[i]

num\_abandoned = int(discovery\_rate \* nests)

worst\_indices = np.argsort([objective\_function(n) for n in population])[-num\_abandoned:]

population[worst\_indices] = np.random.uniform(-5, 5, (num\_abandoned, dimension))

output:

