

## LAB EXAM-QUESTION 4-CUCKOO SEARCH

```
main.py
1 import numpy as np
2 import random
3 def evaluate_fitness(solution):
4     return np.sum(np.square(solution))
5 def random_solution(lb, ub):
6     return np.random.uniform(lb, ub)
7 def levy_flight(best_nest, alpha, lb, ub):
8     step = np.random.normal(0, 1, len(best_nest)) * (np.abs(np.random.normal(0, 1, len(best_nest))) ** alpha)
9     cuckoo = best_nest + step
10    cuckoo = np.clip(cuckoo, lb, ub)
11    return cuckoo
12 def cuckoo_search(num_nests, max_iter, lb, ub, alpha, pa):
13    nests = [random_solution(lb, ub) for _ in range(num_nests)]
14    fitness = [evaluate_fitness(nest) for nest in nests]
15    best_nest = nests[np.argmin(fitness)]
16    best_fitness = min(fitness)
17    for iteration in range(max_iter):
18        cuckoo = levy_flight(best_nest, alpha, lb, ub)
19        cuckoo_fitness = evaluate_fitness(cuckoo)
20        random_index = random.choice(range(num_nests))
21        if cuckoo_fitness < fitness[random_index]:
22            nests[random_index] = cuckoo
23            fitness[random_index] = cuckoo_fitness
24        best_index = np.argmin(fitness)
25        best_nest = nests[best_index]
26        best_fitness = fitness[best_index]
27        for i in range(num_nests):
28            if random.uniform(0, 1) < pa:
29                nests[i] = random_solution(lb, ub)
30                fitness[i] = evaluate_fitness(nests[i])
31        print(f"Iteration {iteration + 1}: Best Fitness = {best_fitness}")
32    print("Best Solution:", best_nest)
33    return best_nest, best_fitness
34 num_nests = 25
35 max_iter = 50
36 dim = 10
37 lb = -5
38 ub = 5
39 alpha = 1.5
40 pa = 0.25
41 best_solution, best_fitness = cuckoo_search(num_nests, max_iter, np.full(dim, lb), np.full(dim, ub), alpha, pa)
42
```

OUTPUT:

```
Iteration 1: Best Fitness = 44.5377052175667
Iteration 2: Best Fitness = 36.90952655878219
Iteration 3: Best Fitness = 40.47610989054335
Iteration 4: Best Fitness = 45.35537821405227
Iteration 5: Best Fitness = 45.35537821405227
Iteration 6: Best Fitness = 45.35537821405227
Iteration 7: Best Fitness = 7.049227775166582
Iteration 8: Best Fitness = 3.174019955211656
Iteration 9: Best Fitness = 3.174019955211656
Iteration 10: Best Fitness = 4.470220023359263
Iteration 11: Best Fitness = 4.937519770708263
Iteration 12: Best Fitness = 4.937519770708263
Iteration 13: Best Fitness = 10.057020113910507
Iteration 14: Best Fitness = 10.057020113910507
Iteration 15: Best Fitness = 10.057020113910507
Iteration 16: Best Fitness = 10.057020113910507
Iteration 17: Best Fitness = 13.684411801454038
Iteration 18: Best Fitness = 13.684411801454038
Iteration 19: Best Fitness = 13.684411801454038
Iteration 20: Best Fitness = 14.54195310798317
Iteration 21: Best Fitness = 14.54195310798317
Iteration 22: Best Fitness = 24.645717068610708
Iteration 23: Best Fitness = 22.789049544897743
Iteration 24: Best Fitness = 22.789049544897743
Iteration 25: Best Fitness = 22.789049544897743
Iteration 26: Best Fitness = 29.12871294892262
Iteration 27: Best Fitness = 29.12871294892262
Iteration 28: Best Fitness = 19.898835046379237
Iteration 29: Best Fitness = 19.898835046379237
Iteration 30: Best Fitness = 21.89471142728356
Iteration 31: Best Fitness = 21.89471142728356
Iteration 32: Best Fitness = 21.89471142728356
Iteration 33: Best Fitness = 17.888040868007938
Iteration 34: Best Fitness = 41.020430314299745
Iteration 35: Best Fitness = 31.58990940423674
Iteration 36: Best Fitness = 31.58990940423674
Iteration 37: Best Fitness = 31.58990940423674
Iteration 38: Best Fitness = 31.58990940423674
Iteration 39: Best Fitness = 31.58990940423674
Iteration 40: Best Fitness = 24.57242011397105
Iteration 41: Best Fitness = 29.40668988049852
Iteration 42: Best Fitness = 29.40668988049852
Iteration 43: Best Fitness = 31.58990940423674
Iteration 44: Best Fitness = 31.183267446665862
Iteration 45: Best Fitness = 31.183267446665862
Iteration 46: Best Fitness = 31.183267446665862
Iteration 47: Best Fitness = 34.28274580215061
Iteration 48: Best Fitness = 17.90664705564125
Iteration 49: Best Fitness = 17.90664705564125
Iteration 50: Best Fitness = 21.994191903231496
Best Solution: [-0.03842518 -1.0917276 0.07705018 -1.6396505 -2.21503363 -1.05360078
-1.30915692 2.84789114 0.49482163 -1.42154245]
```

input