15/4/24

LAB - 06:

Cuckoo Search:

Algorithm:

Step 1: Define the parameters
n = no. of nests / solutions
i = no. of iterations
Pa = probability of finding cuckoo's egg in that particular nest
Also define the range of the levy flight (upper & lower bound)

Step 2: Randomly assign each nest with some eggs.

Step 3: best_nest = nest[0]
prob = prob[0]

Step 4: while (iterations != 0 && flight is within levy range)
{
// calculate the probability of finding the egg in the current nest
curr = prob[i]
nest = nest[i]
// calculating best nest
if nest < best_nest:
best_nest = nest
prob = curr

iterations --

}

Step 5: print("The best nest :", best_nest)
print("The probability of finding an egg there", prob)

Study algorithm implementation in wireless NWs

→ Define $f(x)$ which we need to optimize
→ this algo. allows us to explore large area of search space
↳ promotes diversity
→ it evaluates the fitness of each nest, maximization problm
↳ higher fitness corresponds to better solution

- egg laying
- Levy flight
- Replace eggs
- Fitness

Wireless Networks:

Used in WSN → aim is to manage limited resources (bandwidth) efficiently. network stability

→ Load balancing
prblm: even distribution
soln: exploring various config

→ Resource Allocation
prblm: bandwidth
soln: max throughput & min congestion

Output:

Generation 1/3 – Best Fitness : 406.17
Generation 2/3 – Best Fitness : 406.17
Generation 3/3 – Best Fitness : 406.17
optimized load distribution : [59, 30, 26.04, 57, 6.4, 30]

15/11 8/10