22/11/24

LAB-07:

Grey Wolf Optimizer

N = no. of wolves
T = total no. of iterations
D = no. of dimensions of search space
alpha, beta, delta = none
Evaluate the fitness of each wolf and assign the best three to alpha, beta and delta wolves.

a = 2 // Initialize the coefficient

for t=1 to T:
  for each wolf i in wolves:
    for each d in dimensions D:

$A_1 = 2 * a * random() - a$

$B_1 = 2 * a$

$X_1 = abs(2 * wolves[i,d] * B_1 - A_1)$

$A_2 = 2 * a * random() - a$

$B_2 = 2 * a$

$X_2 = abs(2 * wolves[i,d] * B_2 - A_2)$

$A_3 = 2 * a * random() - a$

$B_3 = 2 * a$

$X_3 = abs(2 * wolves[i,d] * B_3 - A_3)$

$X_i[d] = (X_1 + X_2 + X_3) / 3$ // Stores the average values for each i,

  (end for)
(end for)

Evaluate the best wolves (alpha, beta, delta) for each iteration.

a = 2 (1 - (t/T))

(end for)

Return the alpha wolf as the best solution.

Implementation wise - Image Processing
+ Disadv

22/11/24
10/10

Disadvantages:

→ Scalability issues → struggles with high dimensional problems

→ High Sensitivity to parameters → pop size, iteration count.

→ Premature convergence → converges too quickly to local optimum especially in complex problems.

→ No Guarantee of optimality → doesnot not guarantee finding global optimum

To overcome these disadvantages, we can use PSO-GWO

↳ PSO's velocity & position update → to maintain diversity & explore
↳ GWO's heirarchical mech. for exploitation to refine solutions.