25/11/24

LAB-04:

Particle swarm optimization

Algorithm:

Step1: Pick a mathematical fn $f(x)$ to optimize

Step2: Set parameters N, w, c1, c2 where c1 is weight of personal best position and c2 is weight of global best position.

Step3: Define the limits within which particle can move

Step4: Assign N with random velocity

Step5: for each particle calculate its fitness that is the best position

Step6: Update velocity based on the best velocity of its own and based on the best velocity found by the swarm.

Step7: It undergoes iterations to check the best solution found

Step8: Then in the final iteration it finds out the best value

Pseudocode:

velocity = w * velocity + c1 * r1 * (best_position - position) + c2 * r2 * (global_best_position - position)

Step1: def func(x)
    return x**2

Step2: Initialize parameters
N = 30, w = 0.5, c1 = 1.5, c2 = 1.5

Step3: for each particle initialize their position and velocity randomly within the range [-10, 10], [-1, 1]

Step4: In this step we evaluate the fitness by sending values to function

Step5: Updating values if curr_value < best_value
update the best of its own particle

If curr_value < global_best update global best value that is velocity based on entire swarm

Step 6: Update velocity which will determine how the particle will move.

Step 7: display the best value found.