Particle swarm optimisation

Algorithm

step 1: Pick a mathematical function f(x) to optimize.

step 2: Set the parameters N, ω, c1, c2 where C1 is weight of personal best position and C2 is weight of global best position

step 3: Define the limits within which particle can move

step 4: Assign N with random velocity

step 5: for each particle calculate its fitness that is the best position

step 6: update velocity based on the best velocity of its own & from based on the best velocity found by the entire swarm

step 7: It undergoes iterations to check the best solution found

step 8: Then in the final iteration it finds out the best value

Pseudocode

```
step 1: def func(x)
            return x**2

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

step 2: Initialise parameters
            N = 30 , W = 0.5, C1 = 1.5 , C2 = 1.5

step 3: for each particle initialise their position and
velocity randomly within the range [-10,10) , [-1,1]

step 4: In this step we evaluate the fitness by sending
values to func

step 5: 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
~~velocity~~ step 7: display the ~~best value~~ best value found
```