The Particle Swarm Optimization Algorithm

Gurbanzade Orkhan
Imamverdiyev Yusif

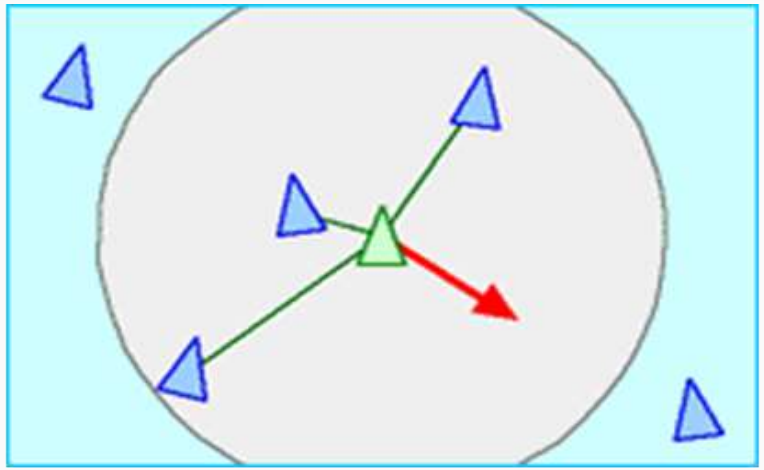
Summary

- ➔ Introduction to Particle Swarm Optimization (PSO)
  - ◆ Origins
  - ◆ Concept
  - ◆ PSO Algorithm

Introduction to the PSO:

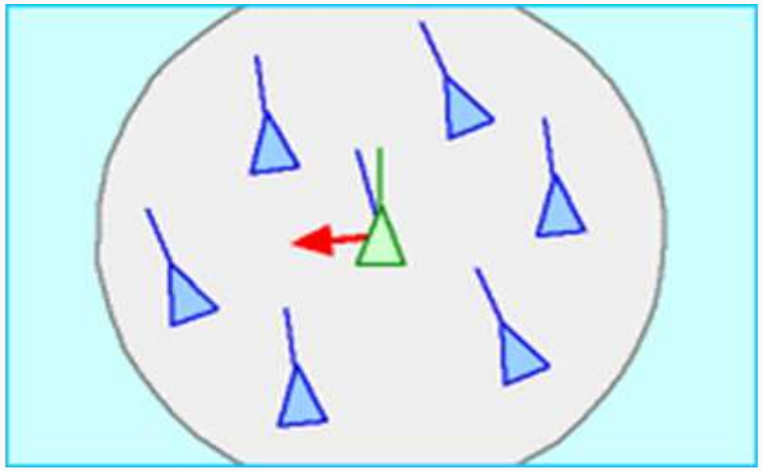
Inspired from the social behavior and dynamic movements with communications of insects, birds and fishnature.

In 1986, Craig Reynolds described this process in 3 simple behaviors



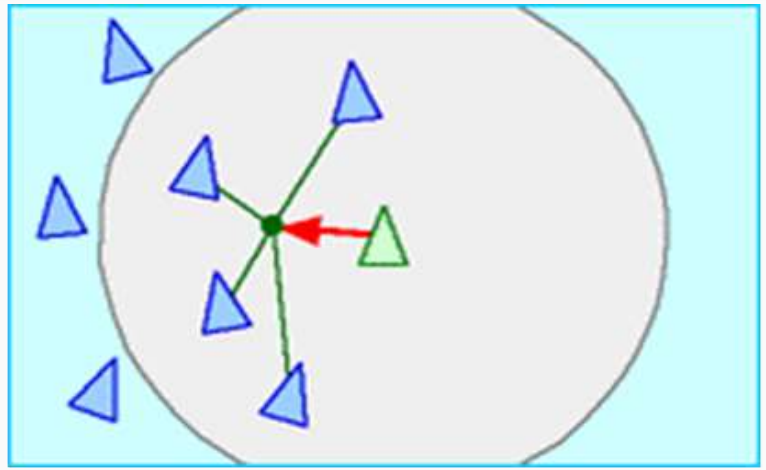
Separation

avoid crowding local flockmates



Alignment

move towards the average heading of local flockmates



Cohesion

move toward the average position of local flockmates

Introduction to the PSO: Concept

- Each particle is searching for the optimum
- Each particle is moving and hence has a velocity.
- Each particle remembers the position it was in where it had its best result so far (its personal best).
    - But this would not be much good on its own; particles need help in figuring out where to search.

Introduction to the PSO: Concept II

- The particles in the swarm co-operate. They exchange information about what they've discovered in the places they have visited
- The co-operation is very simple. In basic PSO it is like this:
  - A particle has a neighbourhood associated with it.
  - A particle knows the fitnesses of those in its neighbourhood, and uses the position of the one with best fitness.
  - This position is simply used to adjust the particle's velocity

PSO Search Strategy

Our Team

Search Area

Personal best location

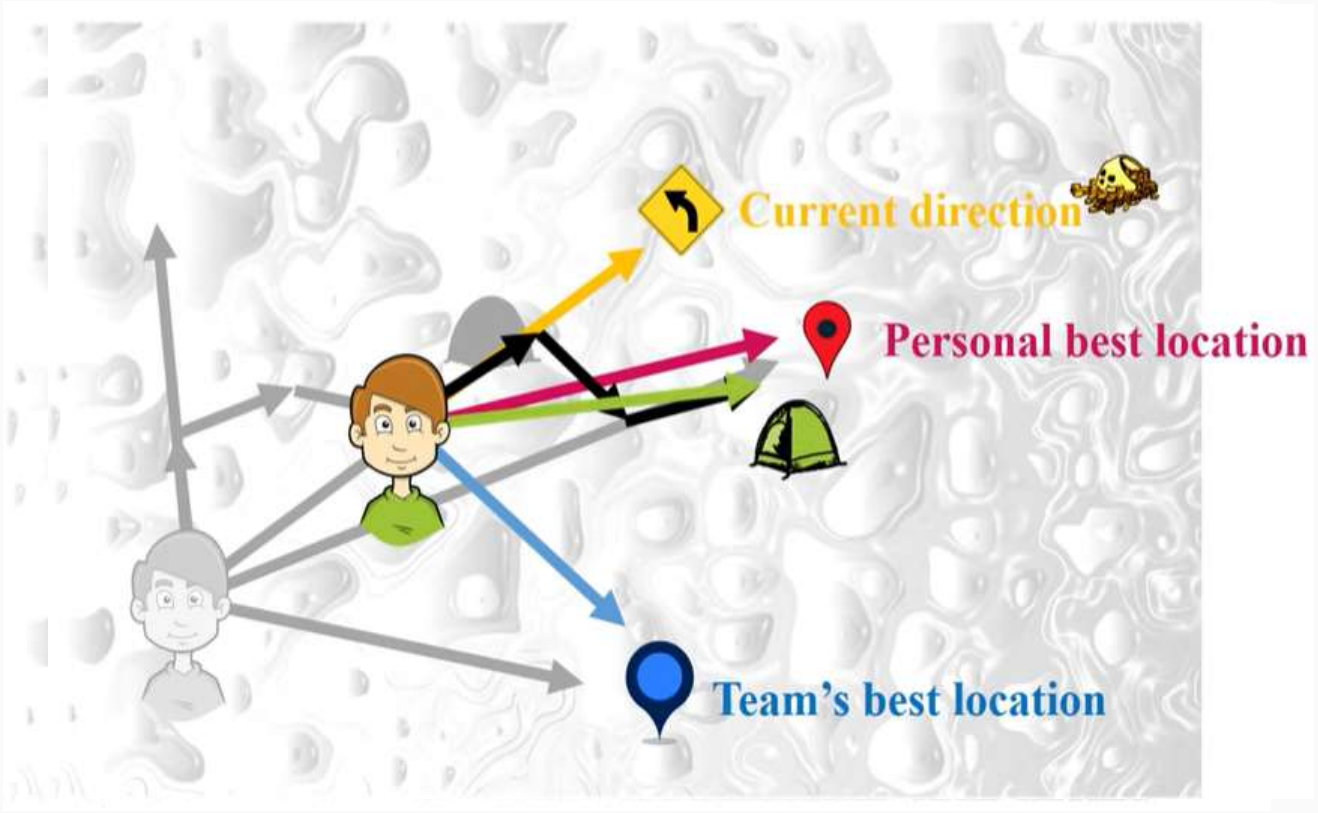
10 km

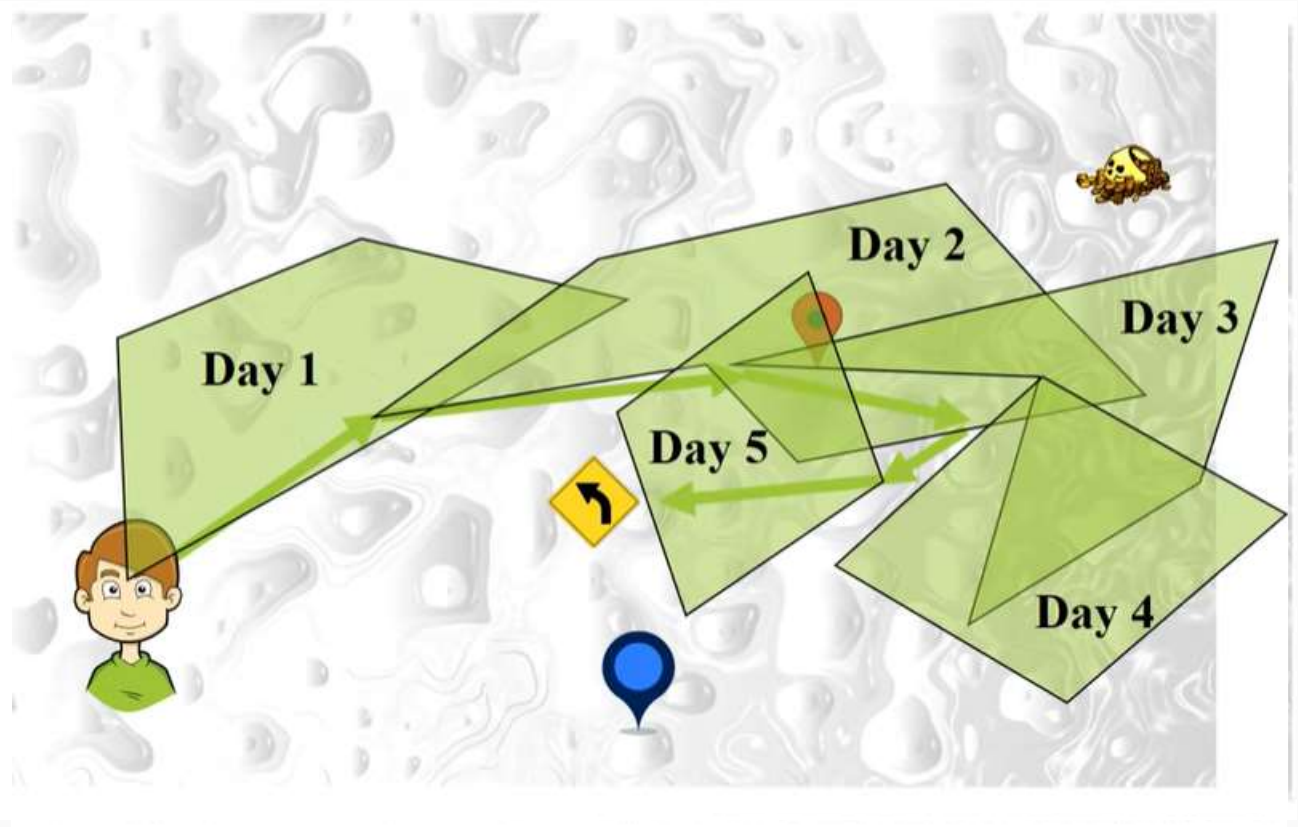
Team best location

10 km

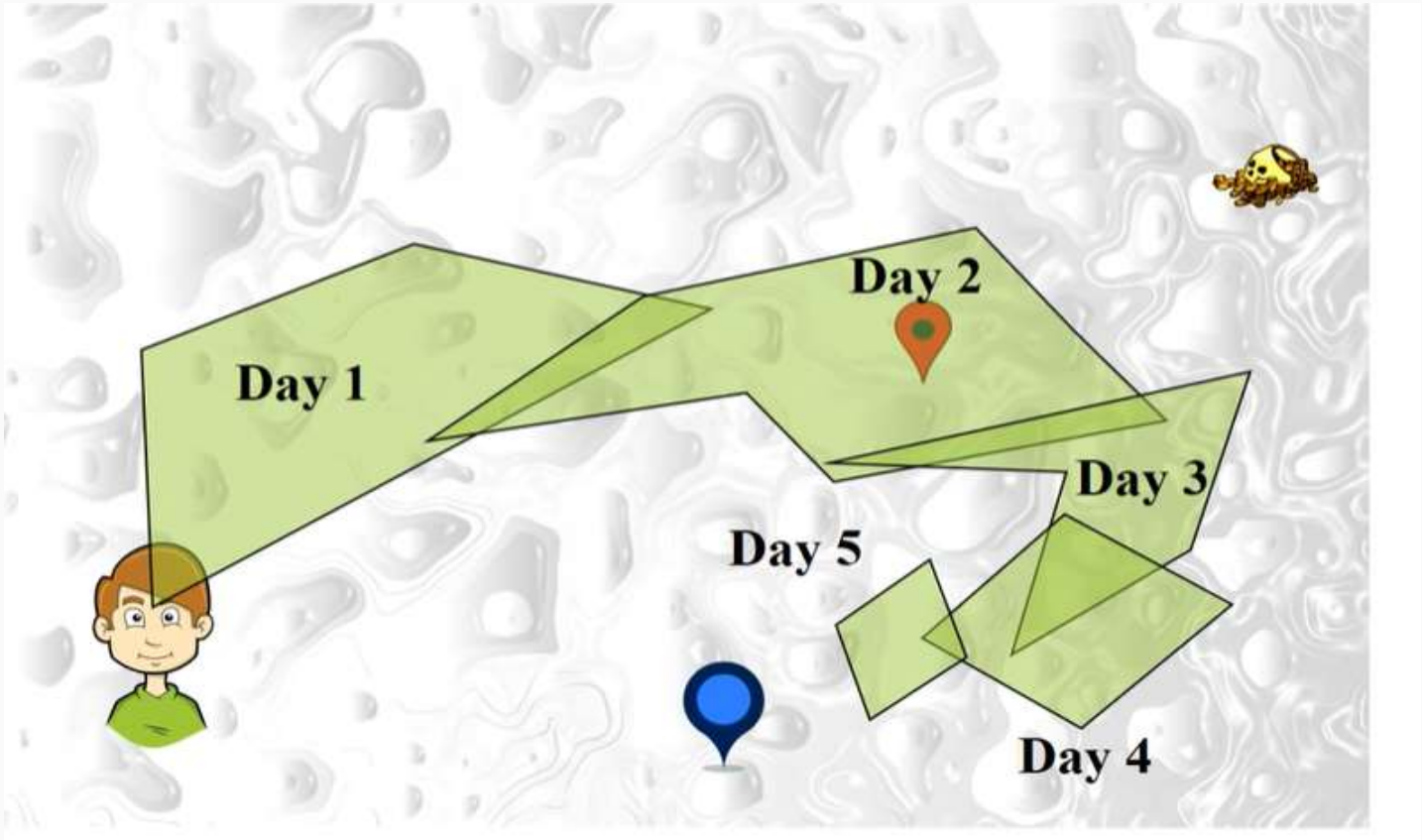
Current direction

10 km


Current direction
Personal best location
Team's best location



$2 \times r \times 10\ km$
$r\ in\ [0,1]$


Day 1
Day 2
Day 3
Day 4
Day 5

$$\overrightarrow{X_i^{d+1}} = \overrightarrow{X_i^{d}} + \overrightarrow{V_i^{d+1}}$$

Position in day *d+1* | **Position in day *d*** | **Velocity in day *d+1***

$d, \ldots]$

$$\overrightarrow{X_i^{t+1}} = \overrightarrow{X_i^t} + \overrightarrow{V_i^{t+1}}$$

$$\overrightarrow{V_i^{t+1}} = w\overrightarrow{V_i^t} + c_1 r_1 \left(\overrightarrow{P_i^t} - \overrightarrow{X_i^t}\right) + c_2 r_2 \left(\overrightarrow{G^t} - \overrightarrow{X_i^t}\right)$$

Inertia

Cognitive component

Social component

Pseudo code

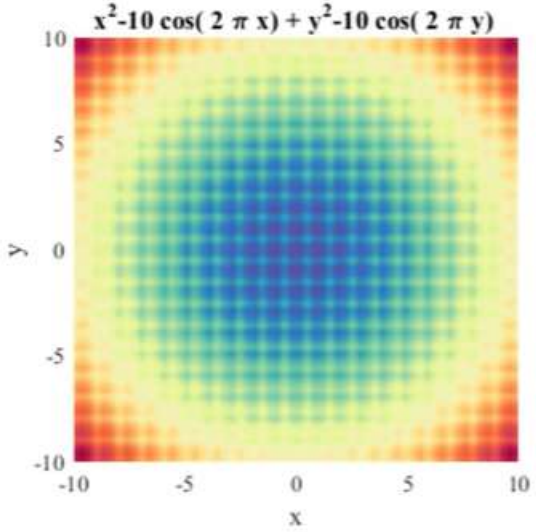
```
Initialize the controlling parameters (N, c1, c2, Wmin, Wmax, Vmax, and MaxIter)

Initialize the population of N particles

do
    for each particle
        calculate the objective of the particle
        Update PBEST if required
        Update GBEST if required
    end for

    Update the inertia weight
    for each particle
        Update the velocity (V)
        Update the position (X)
    end for

white the end condition is not satisfied

Return GBEST as the best estimation of the global optimum
```

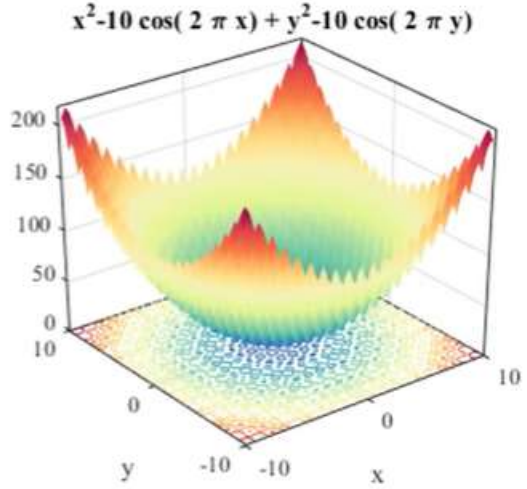
Test function

w 

c_1 

c_2 





Effect of the parameters

Effect of the parameters c1 and c2

What a particle does

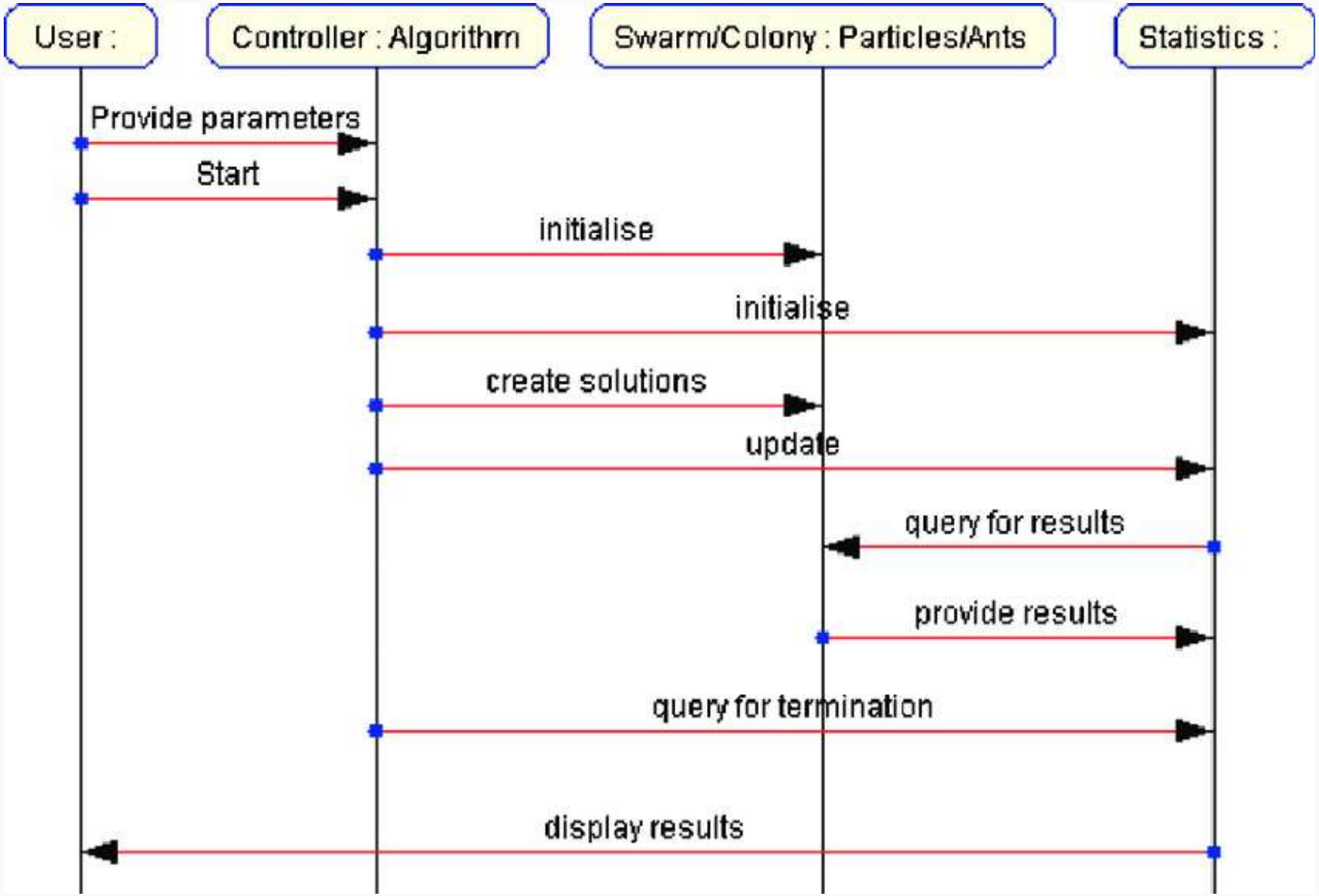
- In each timestep, a particle has to move to a new position. It does this by adjusting its velocity.
  - The adjustment is essentially this:
  - The current velocity PLUS
  - A weighted random portion in the direction of its personal best PLUS
  - A weighted random portion in the direction of the neighbourhood best.
- Having worked out a new velocity, its position is simply its old position plus the new velocity.

- ★ Advantages
  - Insensitive to scaling of design variables
  - Simple implementation
  - Easily parallelized for concurrent processing
  - Derivative free
  - Very few algorithm parameters
  - Very efficient global search algorithm
- ★ Disadvantages
  - Tendency to a fast and premature convergence in mid optimum points
  - Slow convergence in refined search stage (weak local search ability)

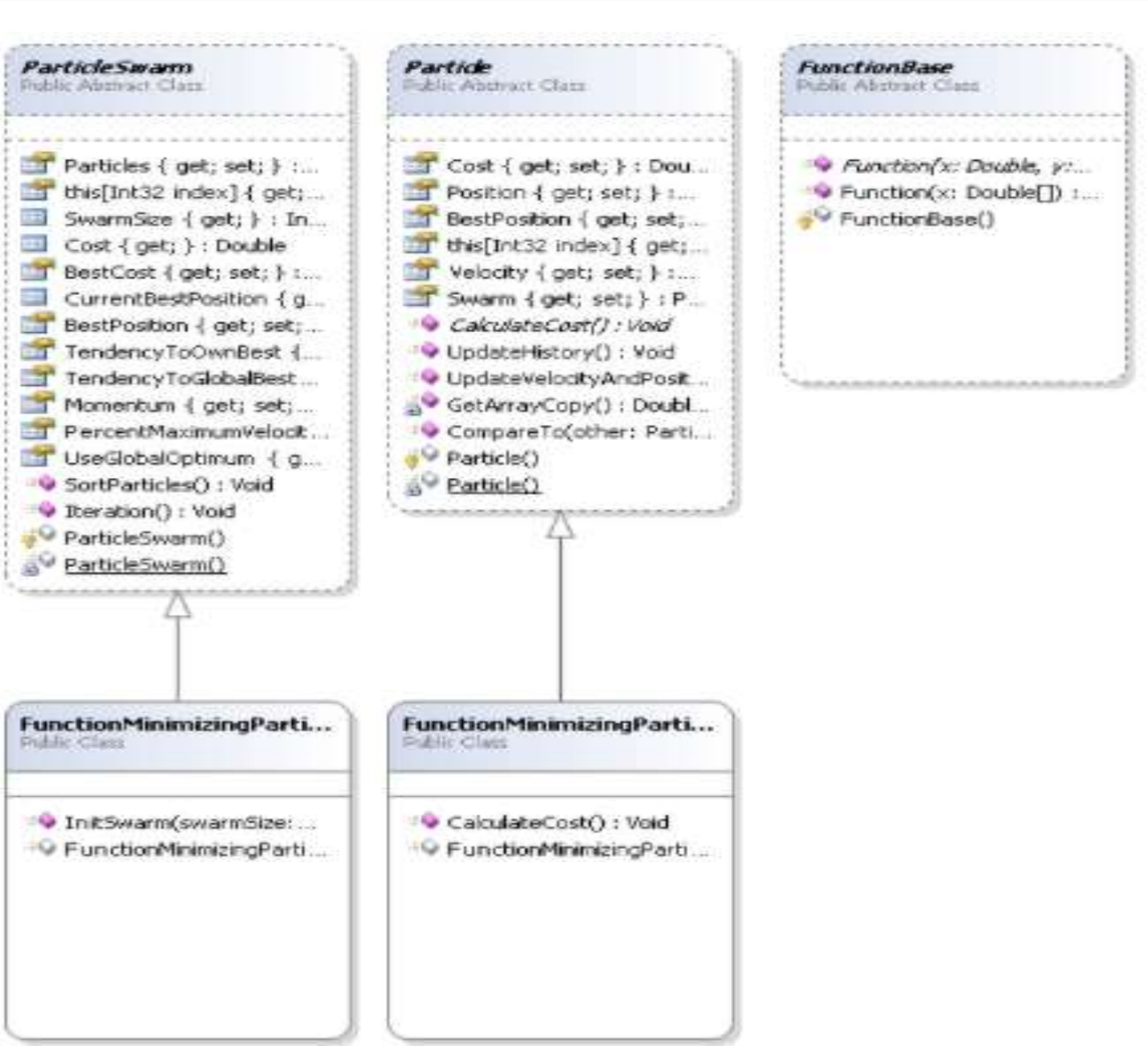
Different Approaches

- **Several approaches**
  - *2-D Otsu PSO*
  - *Active Target PSO*
  - *Adaptive PSO*
  - *Adaptive Mutation PSO*
  - *Adaptive PSO Guided by Acceleration Information*
  - *Attractive Repulsive Particle Swarm Optimization*
  - *Binary PSO*
  - *Cooperative Multiple PSO*
  - *Dynamic and Adjustable PSO*
  - *Extended Particle Swarms*
  - ...

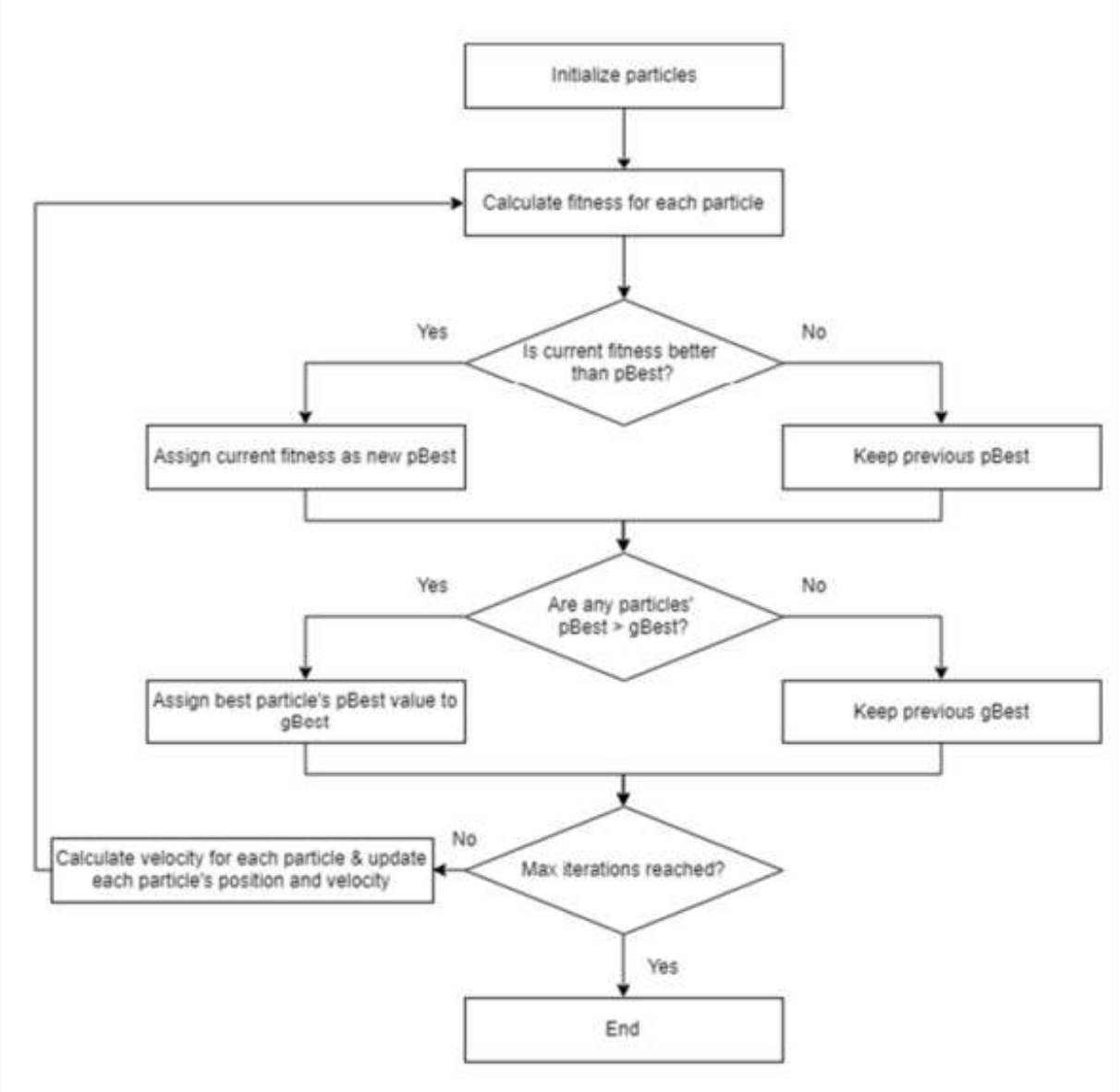
Sequence Diagram



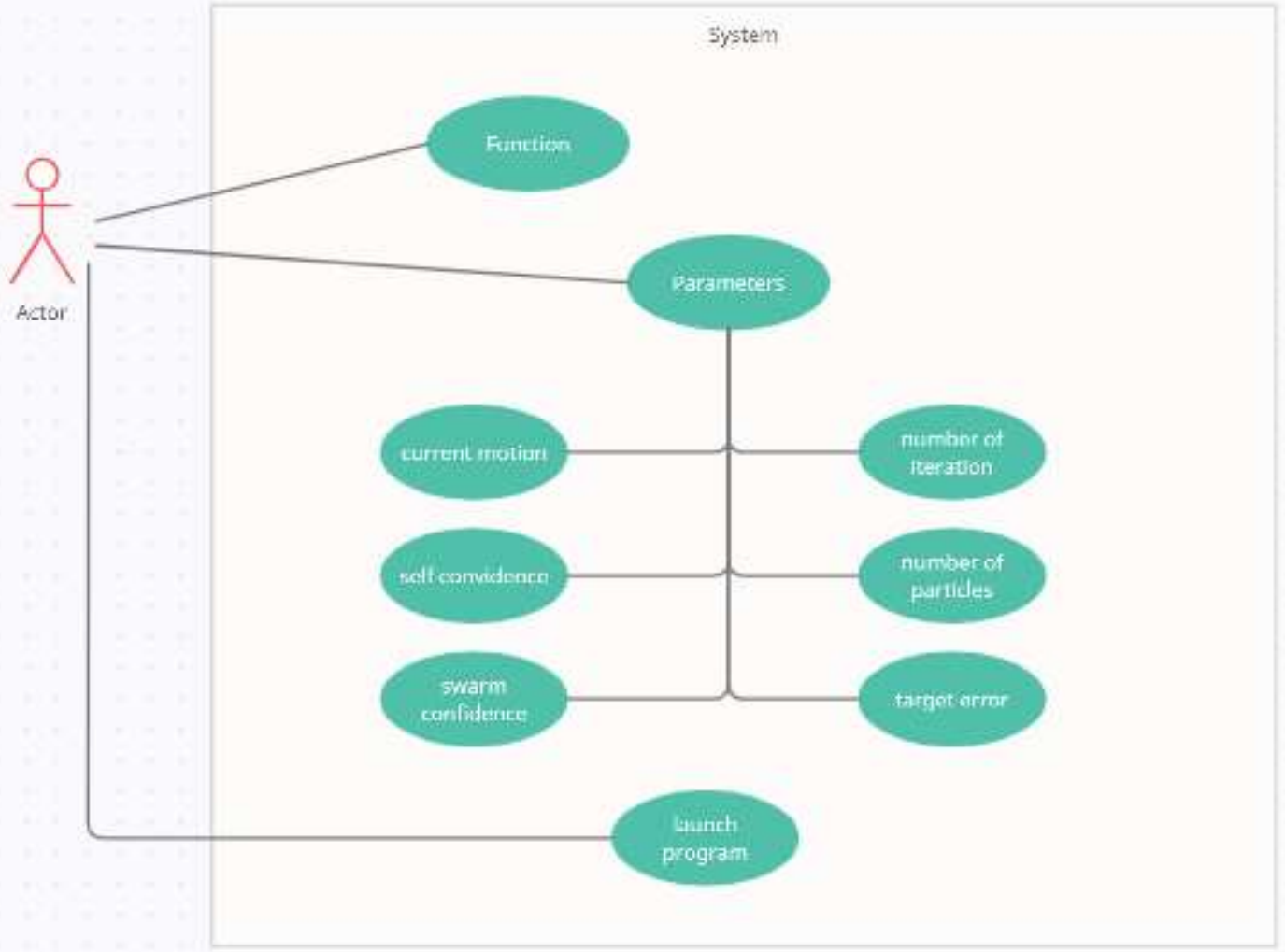
Class Diagram



Flow Diagram



Use-case diagram



Refences

- https://analyticsindiamag.com/a-tutorial-on-particle-swarm-optimization-in-python/
- https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/
- https://pyswarms.readthedocs.io/en/latest/
- https://nathanrooy.github.io/posts/2016-08-17/simple-particle-swarm-optimization-with-python/
- https://towardsdatascience.com/particle-swarm-optimization-visually-explained-46289eeb2e14
- https://www.geeksforgeeks.org/implementation-of-particle-swarm-optimization/
- https://www.researchgate.net/publication/3903911_Particle_swarm_optimization_Development_applications_and_resources
- https://www.researchgate.net/publication/3810335_Empirical_Study_of_Particle_Swarm_Optimization
- https://www.hindawi.com/journals/mpe/2021/5574501/

How can you optimize particle swarms?

a)By iteratively trying to improve a candidate solution

b)Regardless of how the swarm operates, by covering to a local optimum

c)Both of them

What is some different approaches to PSO?

a) Adaptive Mutation PSO, Adaptive PSO Guided by Acceleration Information

b) Adaptive PSO Using Iteration, Compatible PSO

c) Hybrid PSO, Gradient PSO

What is advantage of PSO?

a)Tendency to a fast and premature convergence in mid optimum points

b)Very few algorithm parameters

c)Optimized run time