# MA2000: OTML

*Nachiketa Mishra*

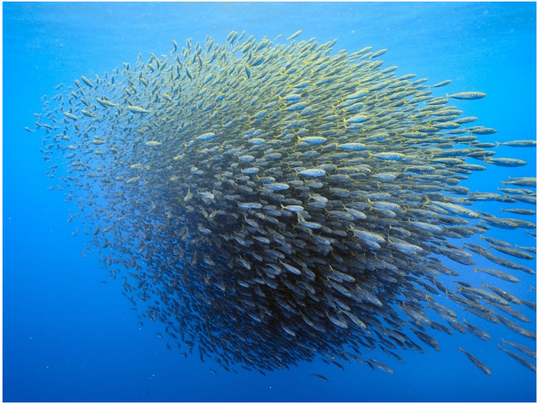
*Indian Institute of Information Technology, Design & Manufacturing, Kancheepuram*

# Particle Swarm Optimization (PSO)

PSO was formulated by R Eberhart and J Kennedy in 1995

Inspired by the social behavior of bird flocking and fish schooling.

**Swarm :** A large or dense group of flying insects



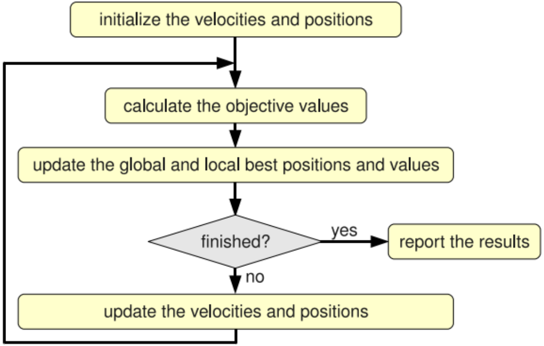
Source: Google

- Suppose a group of birds is searching for food in an area
- Only one piece of food is available
- Birds do not have any knowledge about the location of the food
- But they know how far the food is from their present location
- So what is the best strategy to locate the food?
- The best strategy is to follow the bird nearest to the food

- Each solution is considered as a bird called particle
- All the particles have a fitness value. The fitness values can be calculated using an objective function
- All the particles preserve their individual best performance
- They also know the best performance of their group
- They adjust their velocity considering their best performance and also considering the best performance of the best particle

# Algorithm

- Each particle moves about the cost surface with a velocity.
- The particles update their velocities($v^i$) and positions ($x^i$) based on the local and global best solutions:
  - $\omega$ := Inertia weight
  - $c_i$ := Learning factor
  - $r_i$ := Independent uniform random numbers in (0, 1)
  - $pbest^i(t)$ := Best local solution
  - $G$ := Best global solution




$$\textbf{Velocity}: \quad v^i(t+1) = \underbrace{\omega v^i(t)}_{\text{Inertia Eff.}} + \underbrace{c_1 r_1 \left(pbest^i(t) - x^i(t)\right)}_{\text{Particle memory}} + \underbrace{c_2 r_2 \left(G - x^i(t)\right)}_{\text{Social component}}$$

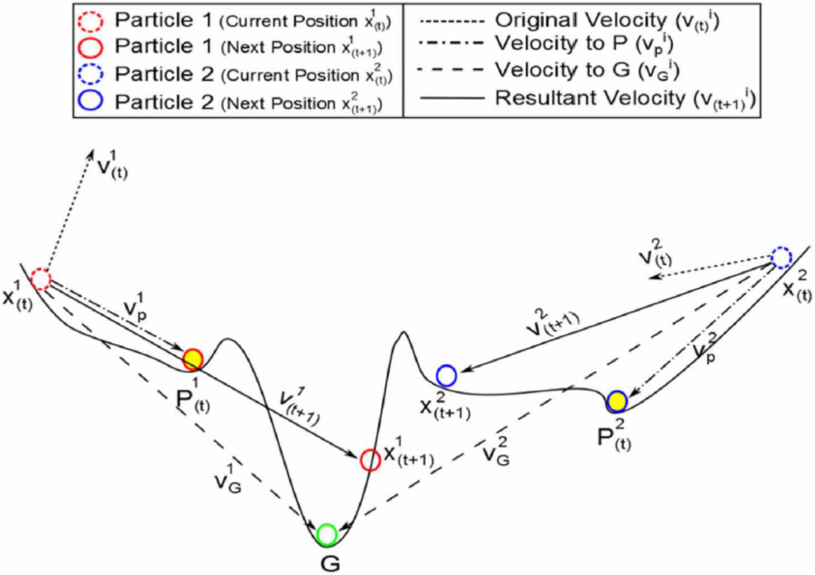
$$\textbf{Position}: \quad x^i(t+1) = x^i(t) + v^i(t+1)$$

- $\omega v^i(t) \Leftarrow$ original velocity or current motion of that particle
- $c_1 r_1 \left( pbst^i(t) - x^i(t) \right) \Leftarrow$ position of the previous best position of that particle; to adjust the velocity towards the best position visited by that particle
- $c_2 r_2 \left( G - x^i(t) \right) \Leftarrow$ position of the best fitness value; to adjust the velocity toward the global best position in all particles

**Remember:**

- High values of the updated velocity make the particles very fast, which may prevent the particles from converging to the optimal solution; thus, the velocity of the particles could be limited to a range $[-V_{max}, V_{max}]$
- This is much similar to the learning rate in the learning algorithms
- A large value of $V_{max}$ expands the search area; thus, the particles may move away from the best solution and it cannot converge correctly to the optimal solution.
- On the other hand, a small value of $V_{max}$ causes the particles to search within a small area, but it may lead to slow convergence.

# Movement of two particles using PSO algorithm in one-dimensional space



# Algorithm

Initialize the particles' positions ($x^i$), velocity ($v^i$), previous best positions ($p^i$), and the number of particles N.

**while** (t <maximum number of iterations (T)) **do**

**for all** Particles (i) **do**

Calculate the fitness function for the current position xi of the i$^{th}$ particle (F ($x^i$)).

**if** ($F(x^i) < F(p^i)$) **then**

$p^i = x^i$ **end if**

**if** ($F(x^i) < F(G)$) **then**

G=$x^i$

end if

Adjust the velocity and positions of all particles according to Equations (1 and 2.

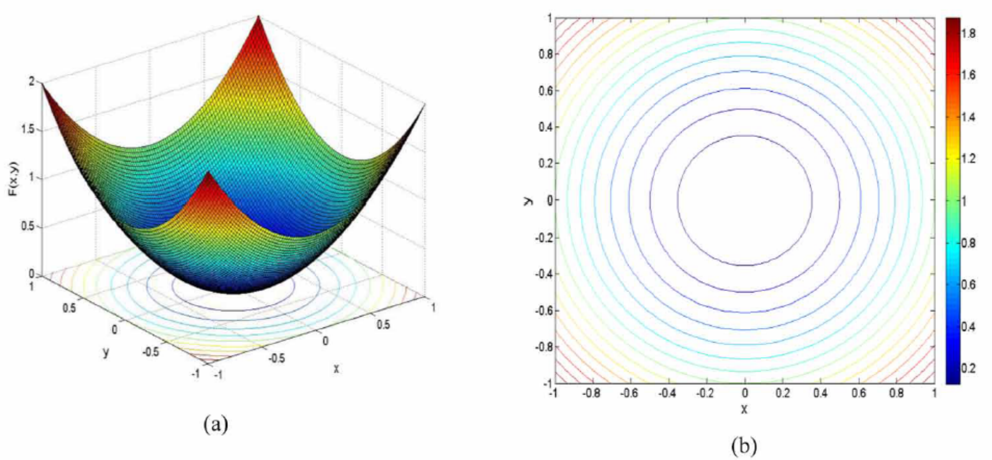
end for

Stop the algorithm if a sufficiently good fitness function is met.

14: **end while**

# Numerical Examples

- Consider a fitness function: $F(x, y) = x^2 + y^2$ with $-1 \leq x, y \leq 1$.
- Goal to minimize the fitness function



# Initial settings:

- Assume the PSO algorithm has five particles $p^i, i = 1, 2, \ldots, 5$

*Table 1. Initial positions, velocity, and best positions of all particles*

| Particle No. | Initial Positions | | Velocity | | Best Solution | Best Position | | Fitness Value |
|---|---|---|---|---|---|---|---|---|
| | $x$ | $y$ | $x$ | $y$ | | $x$ | $y$ | |
| P1 | 1 | 1 | 0 | 0 | 1000 | - | - | 2 |
| P2 | -1 | 1 | 0 | 0 | 1000 | - | - | 2 |
| P3 | 0.5 | -0.5 | 0 | 0 | 1000 | - | - | 0.5 |
| P4 | 1 | -1 | 0 | 0 | 1000 | - | - | 2 |
| P5 | 0.25 | 0.25 | 0 | 0 | 1000 | - | - | 0.125 |

- Assume $\omega = 0.3, \quad c_1 = c_2 = 2, \; r_1 = 0.5$ and $r_2 = 0.5$
- Velocity of all the particles is 0 i.e., $v^i(t) = (0, 0)$ for all $i$
- Calculate $G = (0.25, 0.25)$

# 1st Iteration:

- The position and the velocity of the first particle were calculated as follows

$$
\begin{aligned}
v^1(t+1) &= \omega v^1(t) + c_1 r_1(p^1(t) - x^1(t)) + c_2 r_2(G - x^1(t)) \\
&= 0.3 \times (0,0) + (0,0) + 2 \times 0.5 \times \big((0.25-1),(0.25-1)\big) = (-0.75,-0.75) \\
x^1(t+1) &= x^1(t) + v^1(t+1) = (1,1) + (-0.75,-0.75) = (0.25,0.25)
\end{aligned}
$$

| **Particle No** | **Initial Pos.** | | **Velocity** | | **Best Solution** | **Best Pos.** (pbest) | | **Fitness value** |
|---|---|---|---|---|---|---|---|---|
| | x | y | $v_x$ | $v_y$ | | x | y | |
| $p^1$ | 1 | 1 | -0.75 | -0.75 | 0.125 | 0.25 | 0.25 | 2 |
| $p^2$ | -1 | 1 | 1.25 | -0.75 | 0.125 | 0.25 | 0.25 | 2 |
| $p^3$ | 0.5 | -0.5 | -0.25 | 0.75 | 0.125 | 0.25 | 0.25 | 0.5 |
| $p^4$ | 1 | -1 | -0.75 | 1.25 | 0.125 | 0.25 | 0.25 | 2 |
| $p^5$ | 0.25 | 0.25 | 0 | 0 | 0.125 | 0.25 | 0.25 | 0.125 |

# 2nd Iteration:

| Particle No | Initial Pos. | | Velocity | | Best Solution | Best Pos. (pbest) | | Fitness value |
|---|---|---|---|---|---|---|---|---|
| | x | y | $v_x$ | $v_y$ | | x | y | |
| $p^1$ | 0.25 | 0.25 | -0.225 | -0.225 | 0.00125 | 0.025 | 0.025 | 0.125 |
| $p^2$ | 0.25 | 0.25 | 0.375 | -0.225 | 0.125 | 0.25 | 0.25 | 0.125 |
| $p^3$ | 0.25 | 0.25 | -0.075 | 0.225 | 0.125 | 0.25 | 0.25 | 0.125 |
| $p^4$ | 0.25 | 0.25 | -0.225 | 0.375 | 0.125 | 0.25 | 0.25 | 0.125 |
| $p^5$ | 0.25 | 0.25 | 0 | 0 | 0.125 | 0.25 | 0.25 | 0.125 |

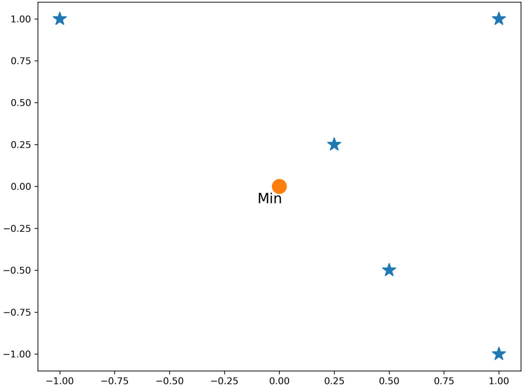



Figure: Initial

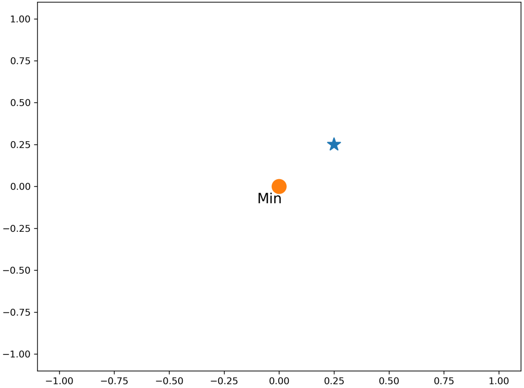



Figure: 1st Iteration

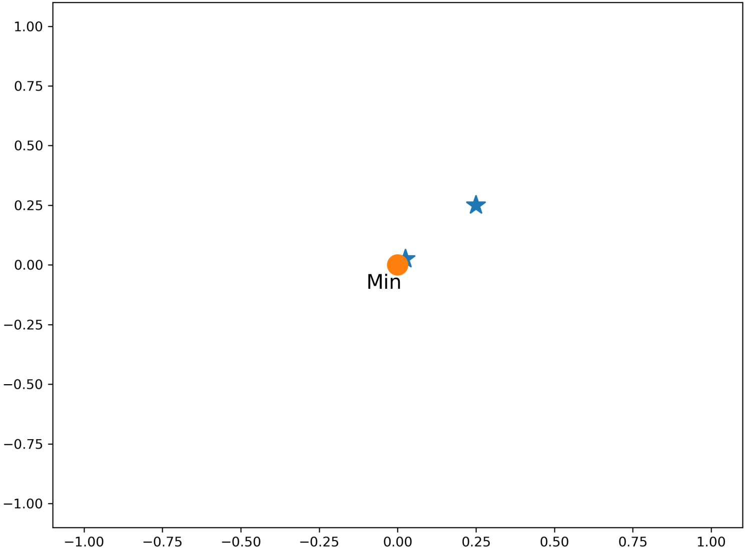



Figure: 2nd Iteration

- **Rastrigin function:** has several local minima but global minima at $x = \mathbf{0}$.

$$F(x, y) = 10d + \sum_{i=1}^{d} \left(x_i^2 - 10\cos(2\pi x_i)\right), \text{ where } x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$$

- **Python code:** https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/
- **Matlab code:** https://www.researchgate.net/publication/296636431_Codes_in_MATLAB_for_Particle_Swarm_Optimization
- **More detailed about PSO:** https://web2.qatar.cmu.edu/~gdicaro/15382/additional/CompIntelligence-Engelbrecht-ch16.pdf