Section 4: Particle Swarm Optimization (PSO)::

Ahmed Métwalli

March 9, 2025

Particle Swarm Optimization (PSO) Parameters

PSO is a metaheuristic optimization algorithm inspired by the social behavior of birds flocking and fish schooling. Below are detailed explanations of each PSO parameter:

- **Population Size:** The number of particles (candidate solutions) exploring the solution space simultaneously. A larger population increases exploration but computationally more expensive.
- **Inertia Weight (w):** Controls the influence of a particle's previous velocity on its current velocity. High inertia promotes global exploration, while low inertia promotes local exploitation.
- **Cognitive Coefficient (c_1):** Also known as the personal coefficient, it scales the attraction of a particle towards its personal best-known position. It guides the particle based on its own experience.
- **Social Coefficient (c_2):** It scales the attraction of a particle towards the global best-known position. This coefficient facilitates collective behavior and social influence among particles.
- **Random Scalars (r_1, r_2):** Random numbers between 0 and 1 introduce stochastic behavior into velocity updates, ensuring diversity and avoiding premature convergence.
- **Particle Position (x_i^t):** Represents the current location of particle i at iteration t within the search space.
- **Particle Velocity (v_i^t):** Determines the direction and magnitude of the movement of particle i at iteration t.
- **Personal Best (p_i^{best}):** The best position discovered by a particle itself up to iteration t. Each particle retains memory of this position.
- **Global Best (g^{best}):** The best position discovered by the entire swarm collectively up to iteration t. Particles use this shared information to move towards promising regions.

1 PSO vs ACO: Combined Comparison

Feature / Stage	**Particle Swarm Optimization (PSO)**	**Ant Colony Optimization (ACO)**
Inspiration	Bird flocking, fish schooling	Ant foraging behavior
Representation	Particles (candidate solutions) moving in a continuous or discrete search space	Ants building solutions by traversing a graph
Movement Rule	Uses velocity updates with cognitive (personal best) and social (global best) components	Uses probabilistic transition based on pheromone and heuristic information
Memory	Each particle retains personal and global best positions	Collective memory stored in pheromone trails
Exploration-Exploitation Control	Controlled by inertia weight and randomness in velocity	Controlled by pheromone evaporation and reinforcement
Convergence Control	Adaptive velocity updates to balance local/global search	Dynamic pheromone levels to avoid stagnation
Application Domains	Engineering design, robotics, feature selection, neural network training, control systems	Routing (TSP, VRP), scheduling, graph-based optimization, bioinformatics, image segmentation
Initialization	• Randomly initialize positions and velocities • Evaluate each particle's fitness	• Ants are placed on start nodes (often random) • Initial pheromone levels are set uniformly

Evaluation of Fitness	Each particle's position is evaluated via an objective function	Each ant constructs a full solution and evaluates it
Communication Mechanism	Particles share the current global best solution; each also retains its own best	Ants communicate indirectly through pheromone on paths
Dependency on Distance	Not dependent on physical distance; communication happens through memory and updates	Strongly dependent on distance as pheromones guide movement in the solution space
Update Mechanism	$v_i^{t+1} = wv_i^t + c_1 r_1 (p_i^{best} - x_i^t) + c_2 r_2 (g^{best} - x_i^t)$ $x_i^{t+1} = x_i^t + v_i^{t+1}$	$P_{ij} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{k\in N}\tau_{ik}^{\alpha}\eta_{ik}^{\beta}}$ $\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum \Delta\tau_{ij}$
Termination Condition	• Maximum number of iterations • Convergence (particles stop improving)	• Maximum iterations or time • No improvement in best solution over some iterations
End Comparison	• Often faster on continuous optimization • Well-suited to function optimization, machine learning	• Tailored to discrete, combinatorial problems • Particularly good at routing and scheduling

2 Symbol Notation

Below are the primary symbols used in PSO and ACO formulations:

Symbol	Description
x_i^t	Position of the i-th particle at iteration t
v_i^t	Velocity of the i-th particle at iteration t
w	Inertia weight
c_1, c_2	Acceleration constants (cognitive (personal) and social (global)
r_1, r_2	Random scalars uniformly distributed in $[0, 1]$
p_i^{best}	Personal best (best-known) position of the i-th particle
g^{best}	Global best (best-known) position among all particles
P_{ij}	Probability of moving from node i to node j in ACO
τ_{ij}	Pheromone amount on edge (i, j)
η_{ij}	Heuristic information (e.g., 1/distance) on edge (i, j)
α, β	Exponents for the influence of τ_{ij} and η_{ij}
ρ	Pheromone evaporation rate
$\Delta\tau_{ij}$	Deposited pheromone on edge (i, j)
N	Set of feasible nodes/edges to explore

Table 2: Notation for Symbols in PSO and ACO

Example Problem: PSO Optimization of Sphere Function

Problem Statement:

Consider the Sphere function, a classical (convex) benchmark problem for optimization algorithms, defined as:

$$f(\mathbf{x}) = \sum_{j=1}^{n} x_j^2 \tag{1}$$

where $x \in \mathbf{R}^n$. Our goal is to minimize this function for a 2-dimensional case ($x = [x_1, x_2]$).

Initialization:

Assume we have 2 particles with the following initial positions and velocities:

$$\begin{aligned} x_1^0 &= [4, 5], & v_1^0 &= [0, 0] \\ x_2^0 &= [3, -2], & v_2^0 &= [1, -1] \end{aligned}$$

Parameters:

- Inertia weight: $w = 0.5$
- Cognitive coefficient: $c_1 = 1.5$
- Social coefficient: $c_2 = 2.0$

Objective: Minimize $f(x) = x_1^2 + x_2^2$

Evaluate Initial Fitness:

$$\begin{aligned} f(x_1^0) &= 4^2 + 5^2 = 16 + 25 = 41 \\ f(x_2^0) &= 3^2 + (-4)^2 = 9 + 16 = 25 \end{aligned}$$

Particle 2 is currently the global best.

Set initial best positions:

$$\begin{aligned} p_1^{best} &= [4, 5], \quad f(p_1^{best}) = 41 \\ p_2^{best} &= [3, -4], & f(p_2^{best}) &= 25 \\ g^{best} &= [3, -4], & f(g^{best}) &= 25 \end{aligned}$$

Velocity Update:

Update velocity using the formula:

$$v_i^1 = wv_i^0 + c_1 r_1 (p_i^{best} - x_i^0) + c_2 r_2 (g^{best} - x_i^0)$$

Let's randomly choose: $r_1 = 0.4$ and $r_2 = 0.6$

For Particle 1:

$$\begin{aligned} v_1^1 &= 0.5 \times [0, 0] + 1.5 \times 0.4([4, 5] - [4, 5]) + 2.0 \times 0.6([3, -4] - [4, 5]) \\ &= [0, 0] + 0.6([-1, -9]) \\ &= [-0.6 \times 1, -0.6 \times 9] = [-0.6, -5.4] \end{aligned}$$

For Particle 2:

$$\begin{aligned} v_2^1 &= 0.5 \times [0, -1] + 1.5 \times 0.4([0, 0]) + 2.0 \times 0.6([3, -4] - [3, -4]) \\ &= [0, -0.5] + [0, 0] + [0, 0] = [0, -0.5] \end{aligned}$$

Position Update:

$$x_i^1 = x_i^0 + v_i^1$$

$$\begin{aligned} x_1^1 &= [4, 5] + [-0.6, -5.4] = [3.4, -0.4] \\ x_2^1 &= [3, -4] + [0, -0.5] = [3, -4.5] \end{aligned}$$

Fitness Evaluation After Update:

$$\begin{aligned} f(x_1^1) &= 3.4^2 + (-0.4)^2 = 11.56 + 0.16 = 11.72 \\ f(x_2^1) &= 3^2 + (-4.5)^2 = 9 + 20.25 = 29.25 \end{aligned}$$

Updated best solutions:

$$\begin{aligned} p_1^{best} &= [3.4, -0.4], & f(p_1^{best}) &= 11.72 \\ p_2^{best} &= [3, -4], & f(p_2^{best}) &= 25 \\ g^{best} &= [3.4, -0.4], & f(g^{best}) &= 11.72 \end{aligned}$$

Particle 1 becomes the new global best.
Final Solution (after 1 iteration):

$$g^{best} = [3.4, -0.4], \quad f(g^{best}) = 11.72$$

Note: In practice, this procedure is repeated for multiple iterations until convergence or another termination condition is met.
Reference Book > Swarm Intelligence: Introduction and Applications - Section 3: Particle Swarm Optimization