Particle Swarm Optimization (PSO) in Two Dimensions

1 Introduction

Particle Swarm Optimization (PSO) is a population-based stochastic optimization algorithm inspired by the social behavior of birds and fish schools. It is used to find the minimum of an objective function $f(x, y)$.

2 Problem Definition

We seek to minimize a function $f(x, y)$:

$$\min f(x, y), \quad (x, y) \in R^2$$

where $f(x, y)$ is the objective function.

3 PSO Algorithm

3.1 Step 1: Initialize Parameters

- Number of particles: N
- Maximum iterations: T
- Inertia weight: w
- Acceleration coefficients: c_1, c_2 (cognitive and social factors)
- Randomization factors: $r_1, r_2 \sim U(0, 1)$

3.2 Step 2: Initialize Particles

For each particle i in the swarm:

1. Initialize position $x_i = (x_{ix}, x_{iy})$ randomly within bounds.
2. Initialize velocity $v_i = (v_{ix}, v_{iy})$ randomly.
3. Set personal best $p_i = x_i$.

4. Evaluate fitness $f(p_i)$.
5. Update global best g, the best position among all p_i.

3.3 Step 3: Update Each Particle's Velocity and Position

For each iteration $t = 1, 2, \ldots, T$, update all particles:

3.3.1 Velocity Update

$$v_{ix}^{t+1} = wv_{ix}^{t} + c_1 r_1 (p_{ix} - x_{ix}) + c_2 r_2 (g_x - x_{ix}) \tag{1}$$
$$v_{iy}^{t+1} = wv_{iy}^{t} + c_1 r_1 (p_{iy} - x_{iy}) + c_2 r_2 (g_y - x_{iy}) \tag{2}$$

3.3.2 Position Update

$$x_{ix}^{t+1} = x_{ix}^{t} + v_{ix}^{t+1} \tag{3}$$
$$x_{iy}^{t+1} = x_{iy}^{t} + v_{iy}^{t+1} \tag{4}$$

3.3.3 Evaluate Fitness

Compute $f(x_i)$.

3.3.4 Update Personal Best

If $f(x_i)$ is better than $f(p_i)$, update:

$$p_i = x_i$$

3.3.5 Update Global Best

If $f(p_i)$ is better than $f(g)$, update:

$$g = p_i$$

Repeat the process until convergence or maximum iterations T is reached.

4 Stopping Criteria

The algorithm stops when:

- The maximum number of iterations is reached.
- The global best value $f(g)$ has not changed significantly.
- The velocities become too small, indicating convergence.

5 Summary

1. Initialize swarm (positions, velocities, best values).
2. Compute fitness for all particles.
3. Identify personal bests p_i and global best g.
4. Update velocities using inertia, cognitive, and social components.
5. Update positions based on new velocities.
6. Check stopping criteria and repeat if necessary.