Contents

Origin and Inspiration of PSO

- Population based stochastic optimization technique inspired by social behaviour of bird flocking or fish schooling.
- Developed by Jim Kennedy, Bureau of Labor Statistics, U. S. Department of Labor and Russ Eberhart, Purdue University
- A concept for optimizing nonlinear functions using particle swarm methodology

- Inspired by simulation social behavior
- Related to bird flocking, fish schooling and swarming theory
    - steer toward the center
    - match neighbors' velocity
    - avoid collisions
- Suppose
    - a group of birds are randomly searching food in an area.
    - There is only one piece of food in the area being searched.
    - All the birds do not know where the food is. But they know how far the food is in each iteration.
    - So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food.

What is PSO ?

- In PSO, each single solution is a "bird"(point) in the search space
- The particles fly through the problem space by following the current optimum particles
- Bird adjusts its “flying” according to its own flying experience as well as the flying experience of other particles
- All of particles have fitness values
    - which are evaluated by the fitness function to be optimized
- have velocities
    - which direct the flying of the particles

PSO - Algorithm

- Solutions space/ Population - Swarm
- Candidate solutions - Particles
- Every particle maintains :
  - Fitness value and Velocity
  - Individual best fitness -**pBest** : Best fitness value for this particle
  - Individual best solution : Candidate solution that gave best fitness
  - Global best fitness -**gBest** : Best fitness among all particles
  - Global best solution : Candidate solution that gave best fitness

```
For each particle
    Initialize particles with random values
End
Do
    For each particle
        Calculate fitness value
        If fitness value > pBest :
                pBest = fitness value
        If pBest > gBest :
                gBest = pBest
    End
    For each particle
        Calculate particle velocity
        Update particle position
    End
While termination conditions
```

Updating velocity

$$v_i(t + 1) = wv_i(t) + c_1r_1[pBest_i(t) - x_i(t)] + c_2r_2[gBest(t) - x_i(t)]$$

- $v_i(t)$ - Velocity
- $x_i(t)$ - Position of particle ‘i’ at time t
- w, c_1, and c_2 ($0 \le w \le 1.2$, $0 \le c_1 \le 2$, and $0 \le c_2 \le 2$) - user-supplied coefficients
- r_1 and r_2 ($0 \le r_1 \le 1$ and $0 \le r_2 \le 1$) - random values for each velocity update
- $pBest_i(t)$ - local best solution of particle ‘i’ at time t
- gBest(t) - global best solution at time t

Components

- $wv_i(t)$ - inertia component
- $c_1r_1[pBest_i(t) − x_i(t)]$ - cognitive component
- $c_2r_2[gBest(t) − x_i(t)]$ - social component

Inertia Component - $wv_i(t)$

- Responsible for keeping the particle moving in the same direction as it was originally heading
- W - Inertia coefficient
  - $0 \leq w \leq 1.2$
  - High Value : Encourages exploration of entire search space
  - Lower Value : Speeds up the convergence of the swarm to optima

Cognitive Component : $c_1 r_1[pBest_i(t) - x_i(t)]$

- Obtained from particle’s memory
- Return to region of search space in which it has experienced high individual fitness
- Cognitive coefficient : C_1
  - $0 \leq C_1 \leq 2$
  - Affects the size of the step that the particle takes towards the individual best solution $pBest_i(t)$

Social Component : $c_2 r_2[gBest(t) - x_i(t)]$

- Obtained from Swarm Memory
- Move to the region of high fitness as found by the swarm as a whole
- Social coefficient - C_2
  - $0 \leq C_2 \leq 2$
  - Size of the step that a particle takes towards the global best solution : gBest(t)

Update Position

$$x_i(t + 1) = x_i(t) + v_i(t + 1)$$

- $x_i(t)$: Current position
- $v_i(t + 1)$: New velocity calculated as above

Termination Conditions

- Limit on number of iterations or running time
- Limit on number of iterations since last gBest update
- Predefined target fitness value