

Particle Swarm Optimization

- meta heuristic algorithm
- contains a population of candidate solutions
- particle i position $\rightarrow x_i(t)$ where $x_i(t)$ is a vector in the set of X
- particle i velocity $\rightarrow v_i(t)$
- particle i memory $\rightarrow p_i(t)$ where $p_i(t)$ is the best solution for particle i
- $g(t)$ is the common swarm experience, no i

Particle Update Functions

- $v_i(t+1) = a \cdot v_i(t) + b \cdot (p_i(t) - x_i(t)) + c \cdot (g(t) - x_i(t))$
- $x_i(t+1) = x_i(t) + v_i(t+1)$
- $v_{ij}(t+1) = w \cdot v_{ij}(t) + \text{rand} \cdot c \cdot (p_{ij}(t) - x_{ij}(t)) + \text{rand} \cdot d \cdot (g_j(t) - x_{ij}(t))$ where v_{ij} is the j th scalar
- $x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$

```
% EXAMPLE
%
% inertia = coef*velocity(i,j);
% cognitive = rand()*accel1*(particleBest(i,j) - particlePos(i,j));
% social = rand()*accel2*(globalBest(j) - particlePos(i,j));
```