MACHINE INTELLIGENCE

UNIT - 4

Particle Swarm Optimisation

feedback/corrections: vibha@pesu.pes.edu

VIBHA MASTI

Swarm

- Loosely structured collection of interacting agents
- Agents contribute to and benefit from their group
- Eg: swarm of bees, ant colony, flock of birds, human crowds, cells & molecules

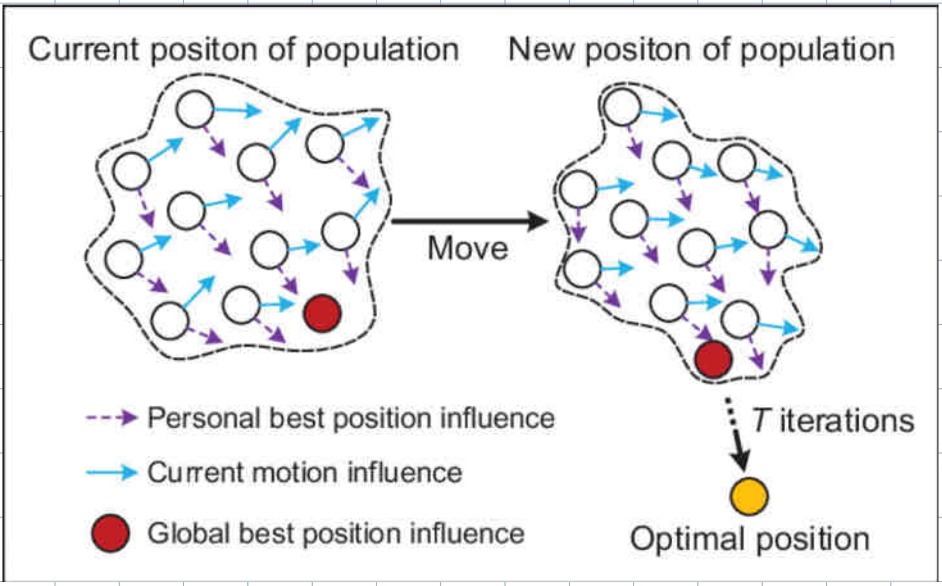
Swarm Intelligence

- No centralised control structures

PARTICLE SWARM OPTIMISATION

- Population-based stochastic optimisation technique
- Potential solutions - particles in the problem space
- Particles fly through the problem space by following current optimum particles
- Each particle searches for optimal solution

- Each particle has velocity
- Each particle remembers its perconal best position (most optimal solution it has visited so far)
- Swarm keeps track of overall best solution


Current positon of population
New positon of population
Move
T iterations
Personal best position influence
Current motion influence
● Global best position influence
Optimal position

Components of PSO

- Each particle: internal state + neighbours
- Let the position of particle i at time t be $x_i(t)$
- The position of particle i at time t+1 is

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

$$x_i(0) = \underset{\uparrow \text{ uniform random}}{U}(xmin, xmax)$$

- Velocity at timestep t+1 influenced by

  1. inertia: previous velocity
  2. pbest : personal best position
  3. lbest : local best (best position found by itself and its social neighbours)

- when entire system taken as neighbourhood, gbest or global best taken

TOPOLOGIES

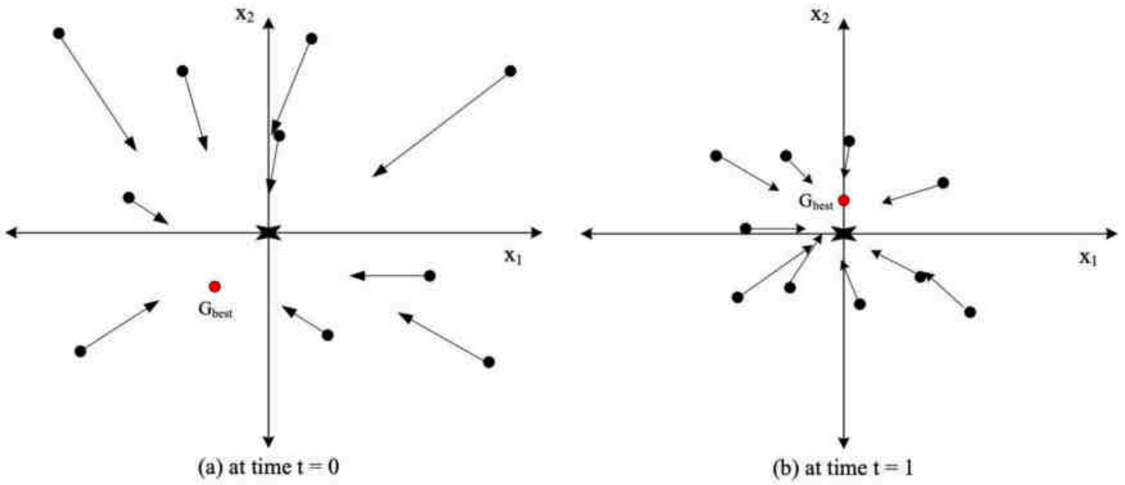
- Topology defines social neighbourhood of each particle

(1) Global Best PSO

- star topology

- all particles communicate with all others

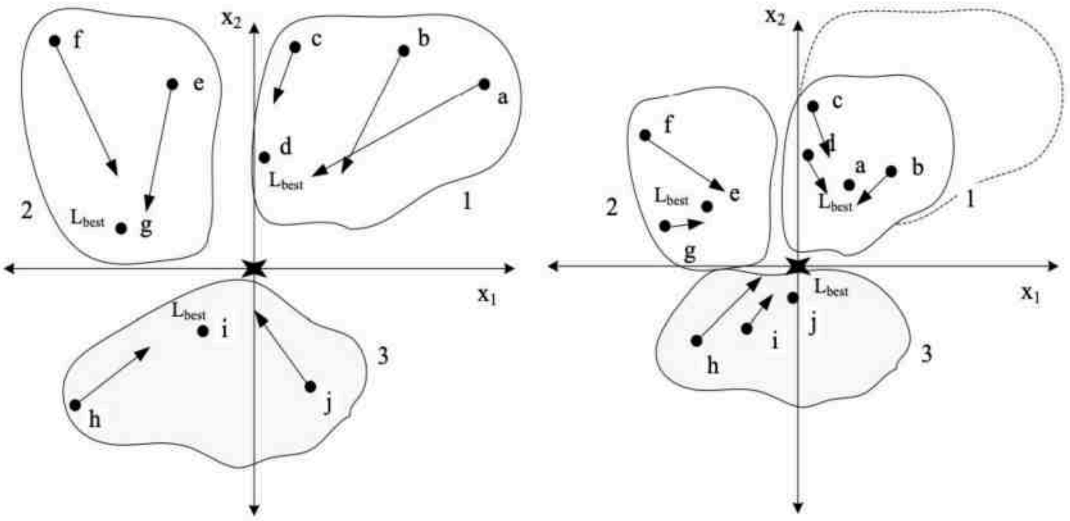
- flaw: local minimum, less diversity
- advantage: converges fast



(a) at time t = 0 (b) at time t = 1

(2) Local Best PSO

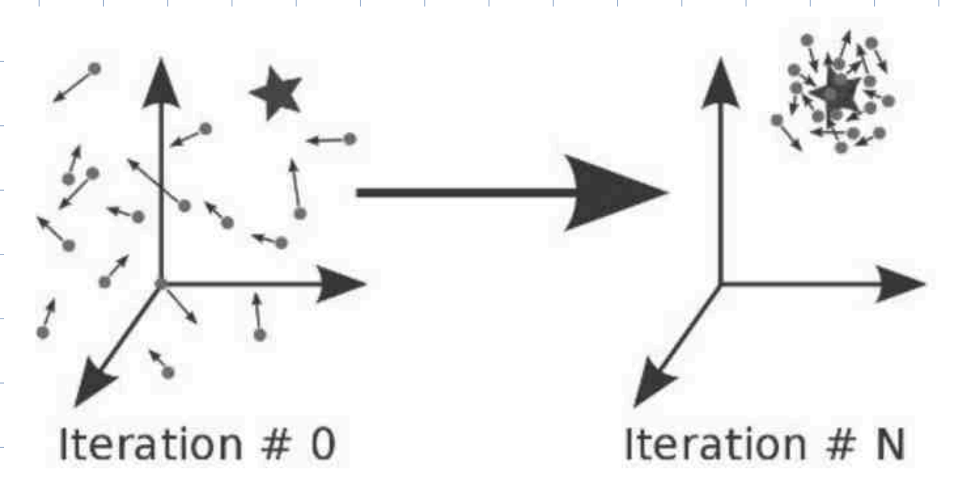
- eg: ring topology — every particle connected to 2 neighbours
- advantage: not easily trapped in local minima, more diversity



(a) at time t = 0 (b) at time t = 1

© vibhas notes 2021

GLOBAL BEST PSO



- Population of particles with random positions and velocities
- Evaluate fitness function for each particle
- compare with pbest and gbest and update accordingly
- Assume D dimensional space
- Position x

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$$

- Velocity V

$$V_i = (V_{i1}, V_{i2}, \dots, V_{iD})^T$$

- Personal best

$$P_i = (P_{i1}, P_{i2}, \ldots, P_{iD})^T$$

- Global best

$$P_g = (P_{g1}, P_{g2}, \ldots, P_{gD})^T$$

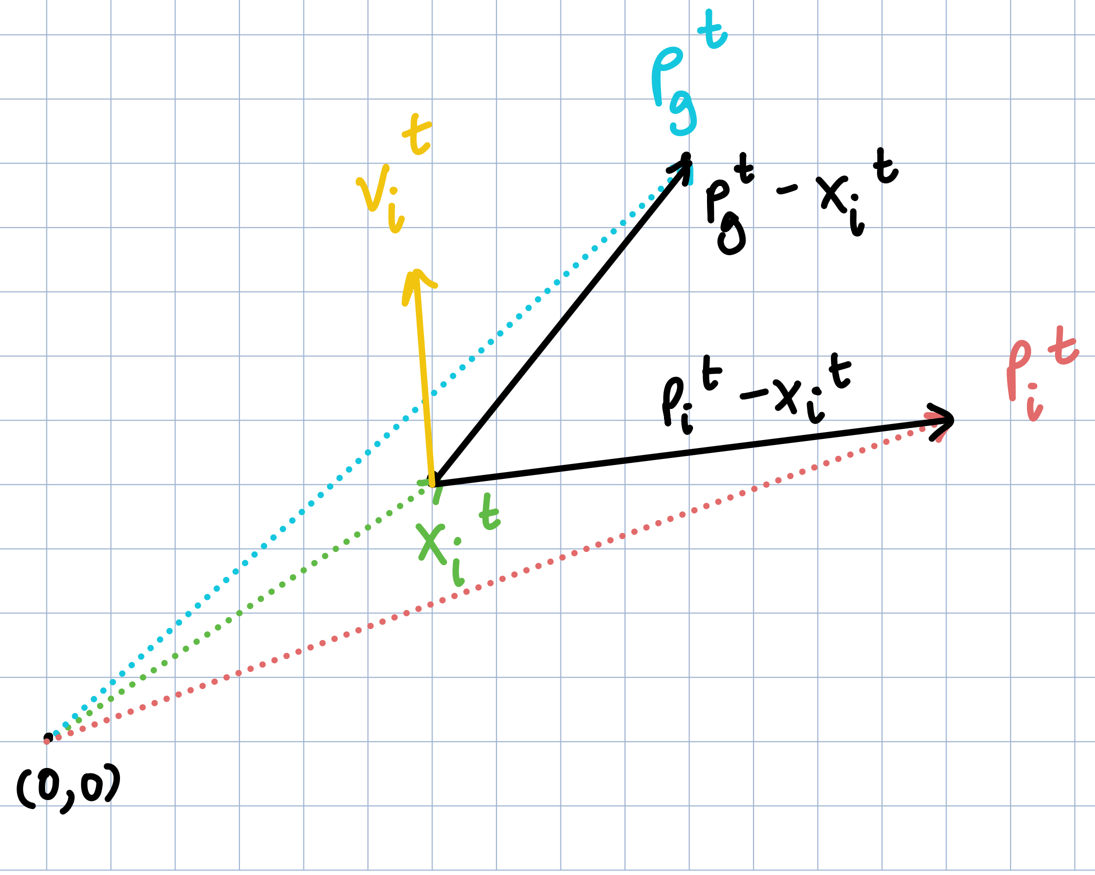
Updating Position

cognitive component c_1

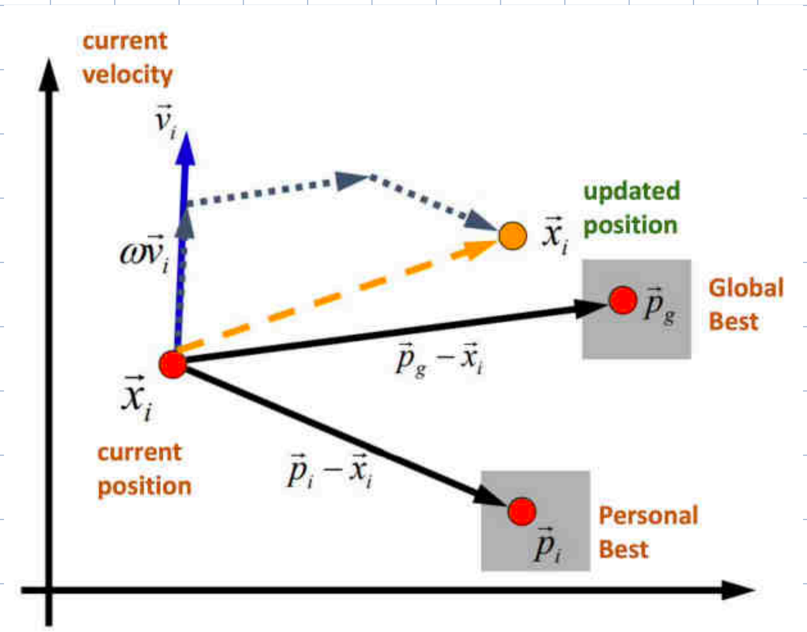
social component c_2

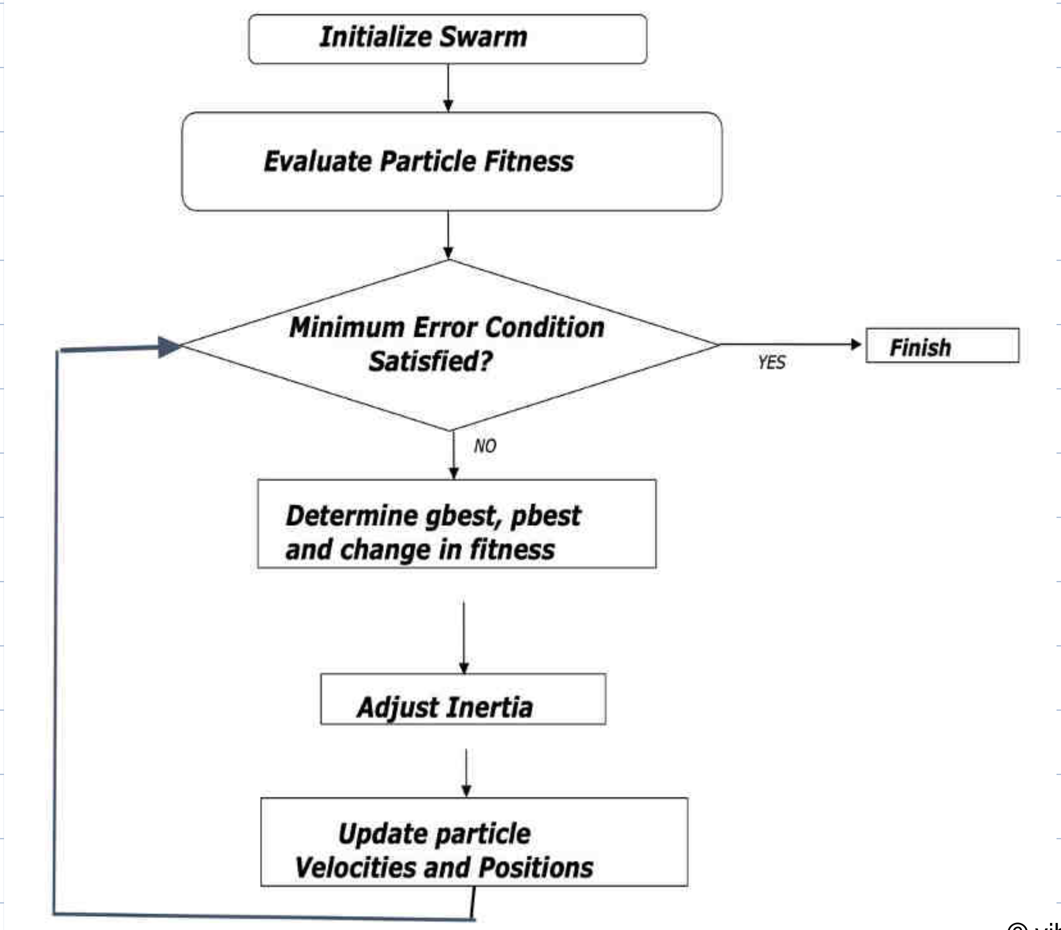
$$V_i^{t+1} = V_i^t + c_1\, rand_1 \times (P_i^t - X_i^t) + c_2\, rand_2 \times (P_g^t - X_i^t)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}$$



- c_1 & c_2 : acceleration coefficients
- $rand_1$ & $rand_2$: uniformly ([0,1]) dist. random numbers




Initialize Swarm
Evaluate Particle Fitness
Minimum Error Condition Satisfied?
YES
Finish
NO
Determine gbest, pbest and change in fitness
Adjust Inertia
Update particle Velocities and Positions

Exploratory vs Exploitative Behaviour

- Exploration: explore diff regions of search space to find optimum
- Exploitation: concentrate search around promising area
- PSO parameters and algorithm should be chosen to balance exploration and exploitation
- Avoid premature convergence
  1. Velocity clamping
  2. Inertia weight
  3. Constriction factor

1. Velocity clamping

if $v_i^t > V_{max}$, $v_i^t = V_{max}$

if $v_i^t < -V_{max}$, $v_i^t = -V_{max}$

- Small V_{max} : encourages exploitation (local)
- Large V_{max} : encourages exploration (global)

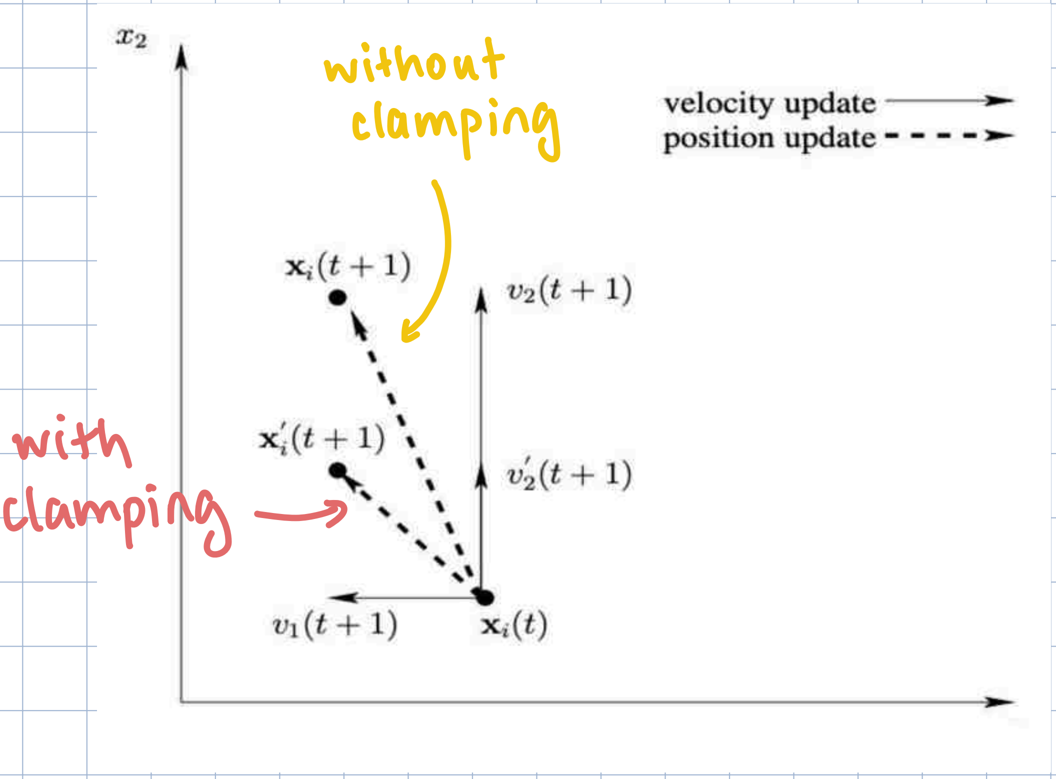
- V_{max} must balance between exploration and exploitation

- Usually, V_{max} chosen to be fraction of domain of each dimension of the search space

$$V_{max} = \delta (X_{max} - X_{min})$$

$$\delta \in [0, 1]$$

- Velocity clamping changes direction of velocity



- Problem: all velocities may eventually equal V_{max} and particles search only boundries of hypercube

- Solutions

  1. change V_{max} when gbest does not change for k iterations

  2. Exponentially decay V_{max}

  3. Introduce inertia weight

  4. Constriction factor

2. Inertia Weight w

- Controls momentum of particle

$$V_i^{t+1} = wV_i^t + c_1\, rand_1 \times (P_i^t - X_i^t) + c_2\, rand_2 \times (P_g^t - X_i^t)$$

- If $w \geq 1$, velocities increase over time and swarm diverges
  - particles fail to change direction

- If $w < 1$, veocities decrease and reach 0

- Dynamic w

$$w = w_{max} + \frac{\overbrace{(w_{min} - w_{max})}^{\text{negative value}}(iter-1)}{itermax - 1}$$

- w and acceleration coefficients

$$w > \frac{1}{2}(c_1+c_2) - 1$$

- Typically, $c_1 = c_2 = 2.0$

3. Constriction coefficient φ

- Velocities constricted by φ

$$V_i^{t+1} = \varphi\left(wV_i^t + c_1\, rand_1 \times \left(P_i^t - X_i^t\right) + c_2\, rand_2 \times \left(P_g^t - X_i^t\right)\right)$$

- where φ is given by

$$\varphi = \frac{2}{\left|2 - \psi - \sqrt{\psi^2 - 4\psi}\right|}$$

- and φ is $(\psi \geq 4)$

$$\psi = c_1 + c_2$$

COGNITION ONLY MODEL

- $c_2 = 0$
- Excludes social component

$$v_i^{t+1} = wv_i^t + c_1 \, rand_1 * (p_i^t - x_i^t)$$

SOCIAL ONLY MODEL

- $c_1 = 0$
- Excludes cognitive component

$$v_i^{t+1} = wv_i^t + c_2 \, rand_2 * (p_g^t - x_i^t)$$

- Faster than full & cognitive models for dynamic environments

Q: Minimise $f(x,y) = x^2+y^2$

ST constraints $-1 \le x \le 1$
$-1 \le y \le 1$

Assume 5 particles

Let ω (inertia) = 0.3

Let $c_1 = c_2 = 2$

Assume $P_i^0 = P_g^0 = 10000$ (large no.)

Initialisation

Choose $P_i \sim U(1,1)$ for 5 particles in 2 dimensions

$P_1 = (1,1)$	$v_1 = (0,0)$
$P_2 = (-1,1)$	$v_2 = (0,0)$
$P_3 = (0.5,-0.5)$	$V_3 = (0,0)$
$P_4 = (1,-1)$	$V_4 = (0,0)$
$P_5 = (0.25, 0.25)$	$V_5 = (0,0)$

Fitness value = $f(x,y) = x^2+y^2$

Run 1

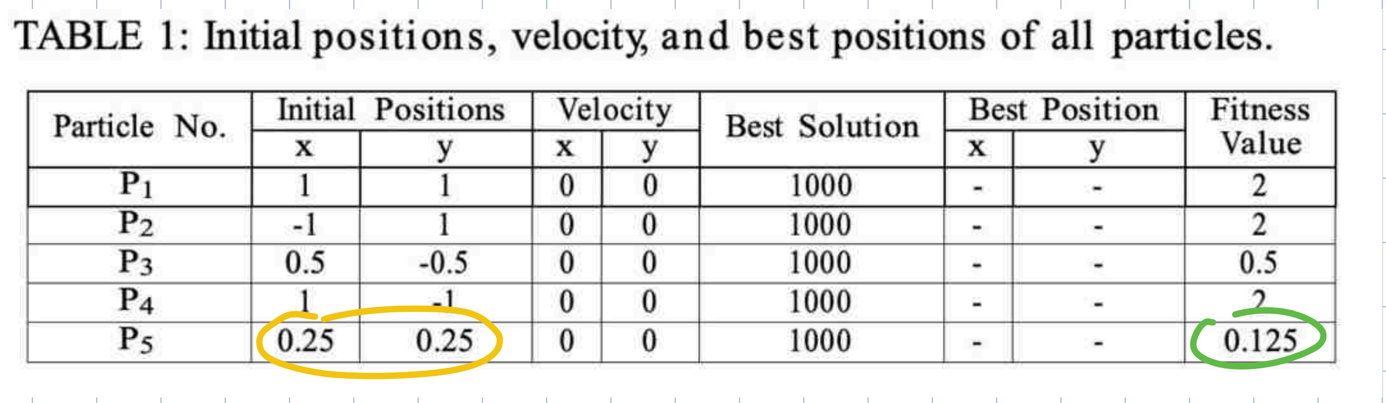


TABLE 1: Initial positions, velocity, and best positions of all particles.

Particle No.	Initial Positions		Velocity		Best Solution	Best Position		Fitness Value
	x	y	x	y		x	y	
P_1	1	1	0	0	1000	-	-	2
P_2	-1	1	0	0	1000	-	-	2
P_3	0.5	-0.5	0	0	1000	-	-	0.5
P_4	1	-1	0	0	1000	-	-	2
P_5	0.25	0.25	0	0	1000	-	-	0.125

Global best value = 0.125

For P1:

$P_g^1 = (0.25, 0.25)$

Let $r_1 = r_2 = 0.5$

Local best of $P_1 = (1,1)$

$$v_1^1 = wv_1^0 + c_1r_1(P_1^0 - x_1^0) + c_2r_2(P_g^0 - x_1^0)$$

$$= 0.3 \times (0,0) + 2 \times 0.5 (0,0) + 2 \times 0.5 (1-0.25, 1-0.25)$$

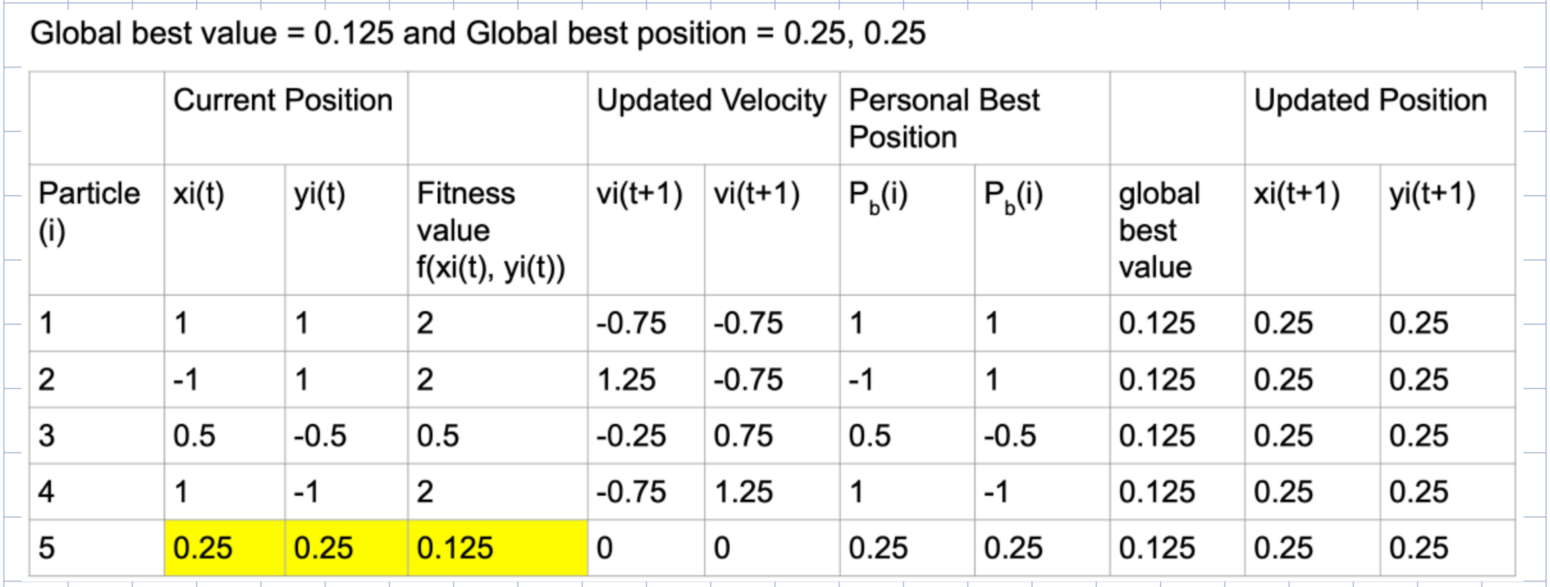
$$v_1^1 = (-0.75, -0.75)$$

$$P_1^1 = (1,1) + (-0.75, -0.75)$$

$P_1' = (0.25, 0.25) \rightarrow$ new PB!

$f(P_1') = 0.125$

and so on



Global best value = 0.125 and Global best position = 0.25, 0.25

	Current Position			Updated Velocity		Personal Best Position			Updated Position	
Particle (i)	xi(t)	yi(t)	Fitness value f(xi(t), yi(t))	vi(t+1)	vi(t+1)	P_b(i)	P_b(i)	global best value	xi(t+1)	yi(t+1)
1	1	1	2	-0.75	-0.75	1	1	0.125	0.25	0.25
2	-1	1	2	1.25	-0.75	-1	1	0.125	0.25	0.25
3	0.5	-0.5	0.5	-0.25	0.75	0.5	-0.5	0.125	0.25	0.25
4	1	-1	2	-0.75	1.25	1	-1	0.125	0.25	0.25
5	0.25	0.25	0.125	0	0	0.25	0.25	0.125	0.25	0.25

Applications

1. Transportation planning
2. Neural networks
3. Clustering