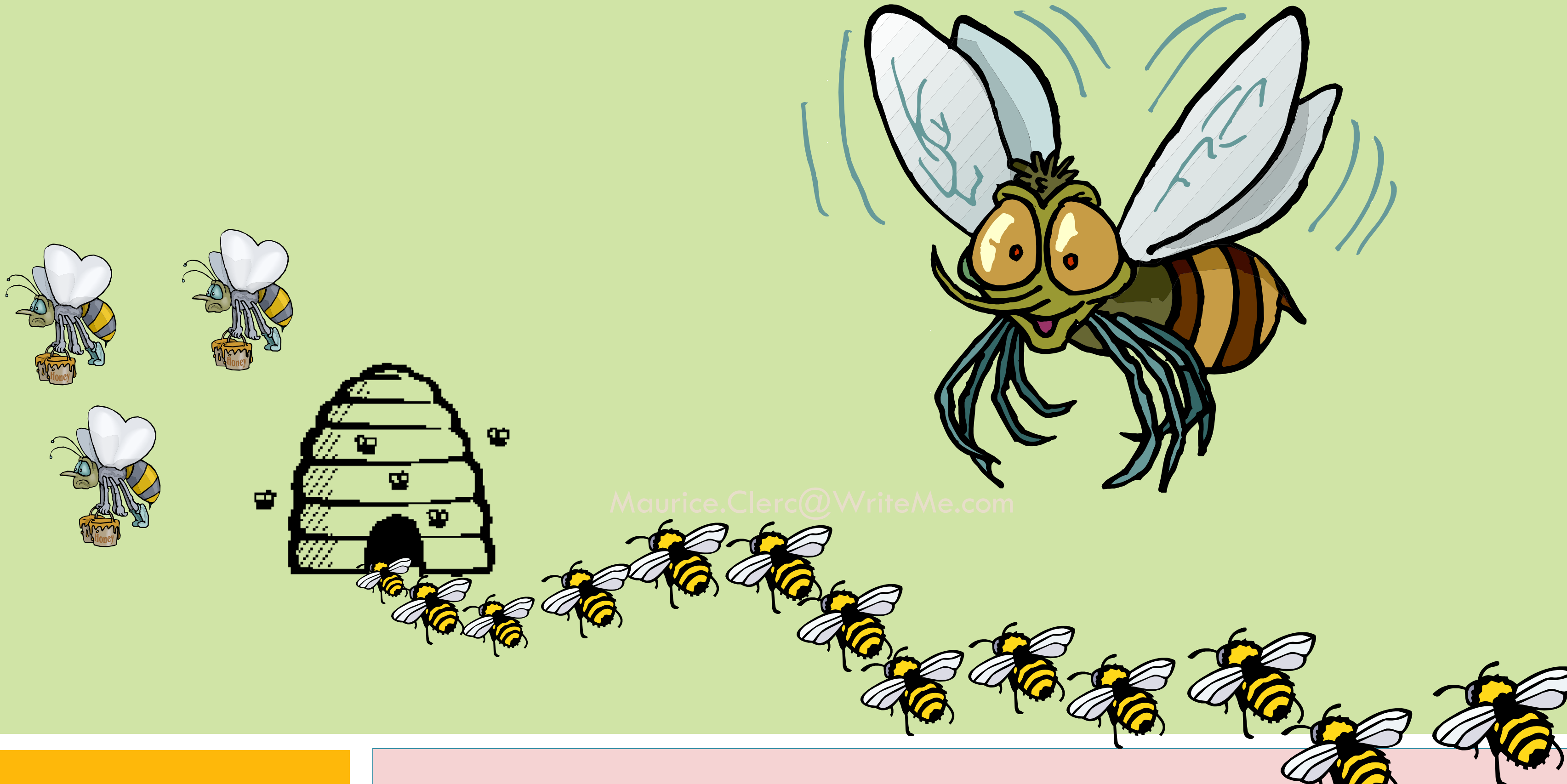
Particle Swarm Optimization (PSO)



Dr. Md. Aminul Haque Akhand, Dept. of CSE, KUET

Introduction : Swarm Intelligence (SI)

- Study of collective behavior in decentralized, self-organized systems.
- Originated from the study of colonies, or swarms of social organisms.
- Collective intelligence arises from interactions.

Overview of Particle Swarm Optimization(PSO)

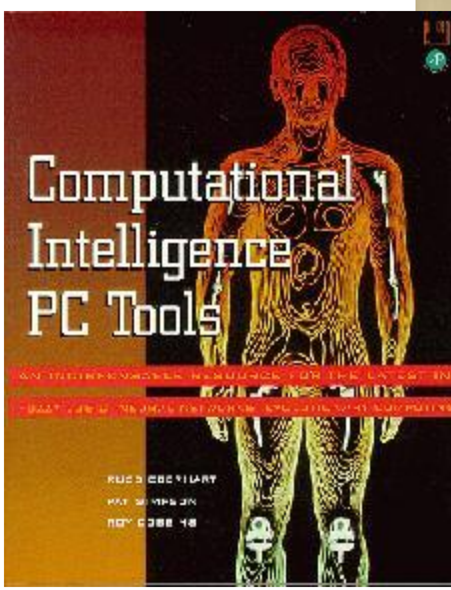
PSO is a population based on stochastic optimization algorithms to find a solution and then solve an optimization problem in a search space.

How can birds or fish exhibit such a coordinated collective behavior?

PSO: History and Properties

- Particle Swarm Optimization:
  - Introduced by Kennedy & Eberhart 1995
  - Inspired by social behavior of birds and shoals of fish
  - Swarm Intelligence-based optimization
  - Population-based optimization
  - Nondeterministic
  - Performance comparable to Genetic Algorithms

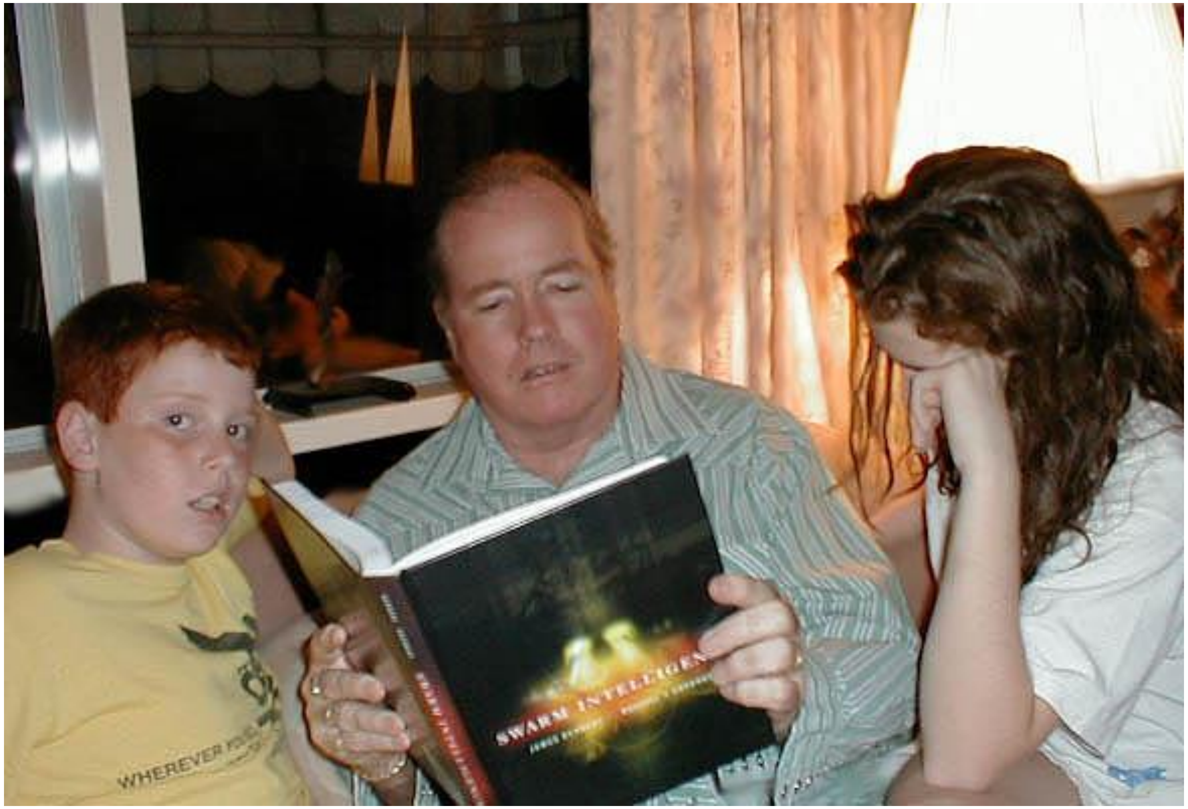
PSO Inventors



Russell Eberhart

eberhart@engr.iupui.edu

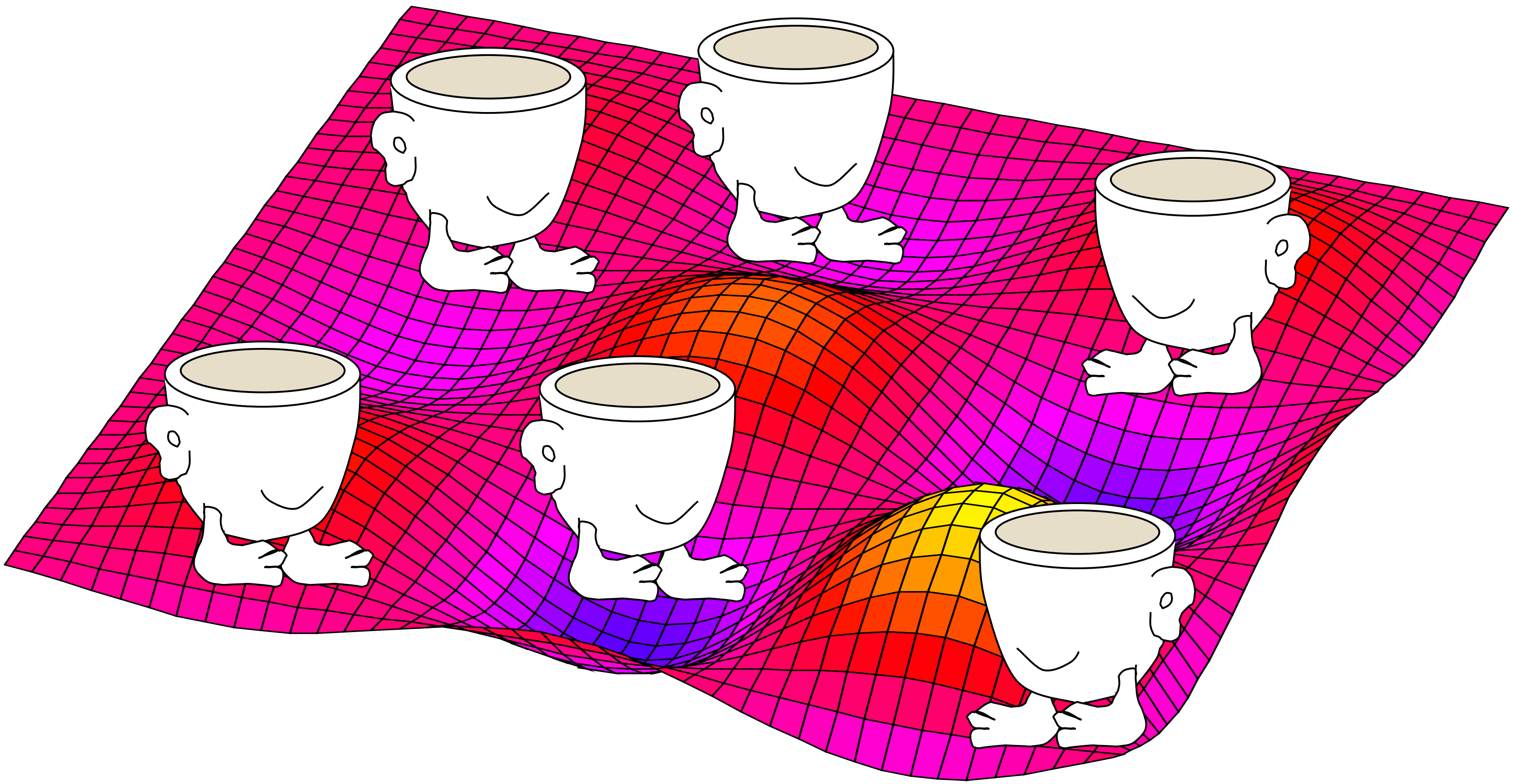
PSO Inventors (2)



James Kennedy

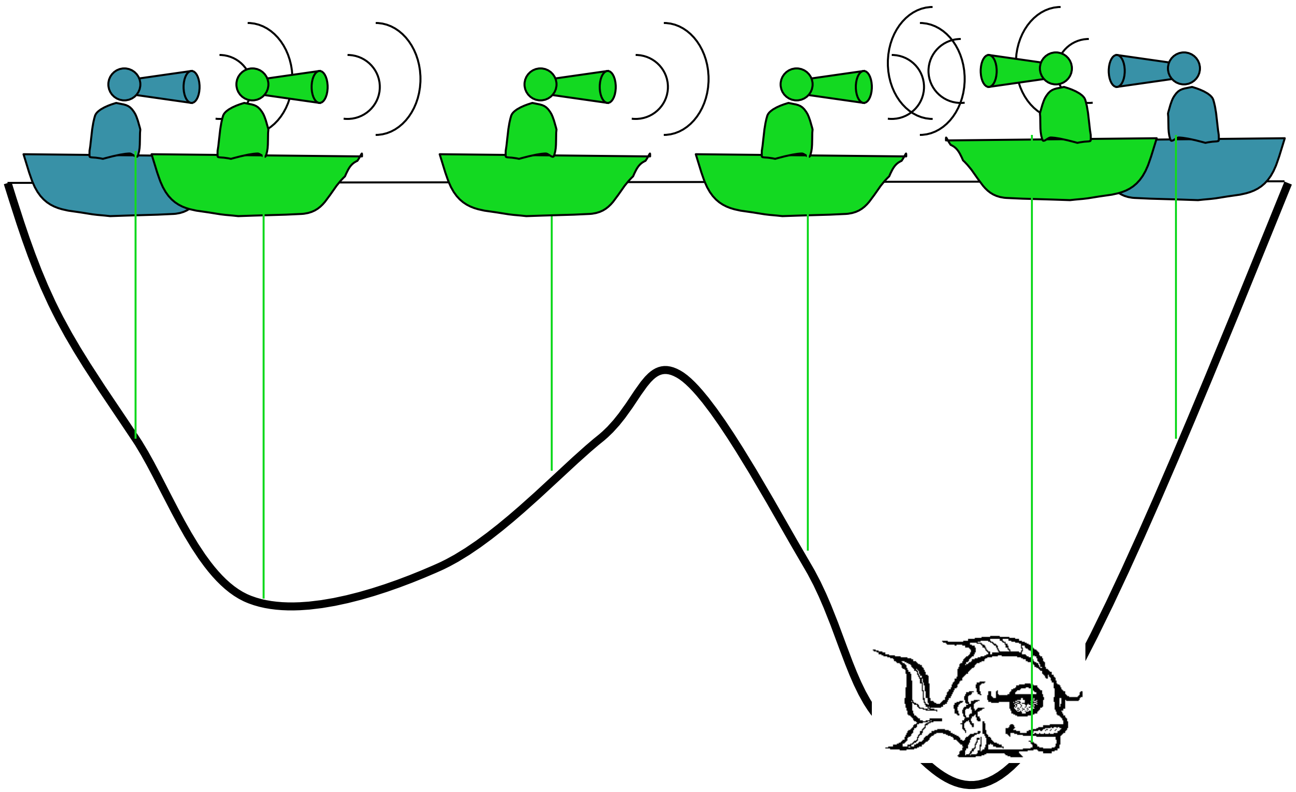
Kennedy_Jim@bls.gov

PSO: United We Stand



Maurice.Clerc@WriteMe.com 2002-04-24

Cooperation Example for PSO



Maurice.Clerc@WriteMe.com 2002-04-24

PSO Formulation

- Swarm : a set of particles (S)
- Particle: a potential solution
  - Position, $X_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in \mathbb{R}^n$
  - Velocity , $V_i = (v_{i1}, v_{i2}, \ldots, v_{in}) \in \mathbb{R}^n$
- Each particle maintains
  - Individual best position: $P_i = (p_{i1}, p_{i2}, \ldots, p_{in}) \in \mathbb{R}^n$

    $pbest_i = f(P_i)$
- Swarm maintains its global best: $P_g \in \mathbb{R}^n$

  $gbest = f(P_g)$

PSO Algorithm

- Basic Algorithm of PSO:
  1. Initialize the swarm from the solution space
  2. Evaluate fitness of each particle
  3. Update individual and global bests
  4. Update velocity and position of each particle
  5. Go to Step 2, and repeat until termination condition

PSO Algorithm (cont.)

- **Original velocity update equation:**

$$V_i^{t+1} = \underbrace{V_i^t}_{\text{Inertia}} + \underbrace{\varphi_1 . r_1 (P_i - X_i^t)}_{\text{Cognitive Component}} + \underbrace{\varphi_2 . r_2 (P_g - X_i^t)}_{\text{Social Component}}$$

  - with $r_1, r_2 \sim U(0,1)$
  - φ_1, φ_2 : acceleration constant

PSO Algorithm (cont.)

- **Original velocity update equation:**

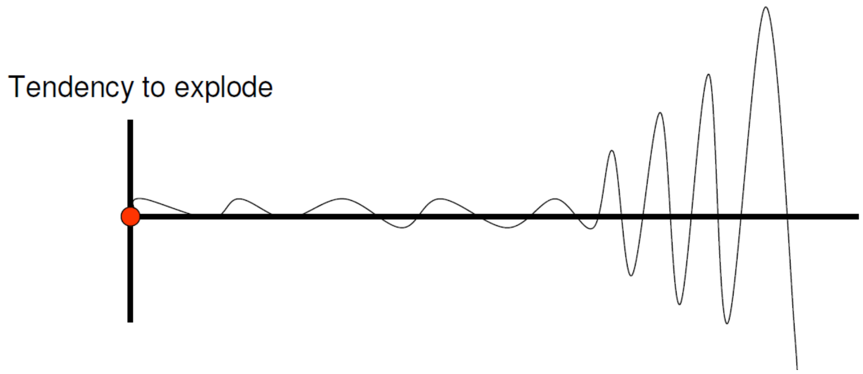
$$V_i^{t+1} = \underbrace{V_i^t}_{\text{Inertia}} + \underbrace{\varphi_1 . r_1 (P_i - X_i^t)}_{\text{Cognitive Component}} + \underbrace{\varphi_2 . r_2 (P_g - X_i^t)}_{\text{Social Component}}$$

  - with $r_1, r_2 \sim U(0,1)$
  - φ_1, φ_2 : Acceleration constants or coefficients
  - Acceleration constants sometimes define as C_1 and C_2

- **Position Updat** $X_i^{t+1} = X_i^t + V_i^{t+1}$

PSO Algorithm - Parameters

- Acceleration constant φ_1, φ_2
  - Small values limit the movement of the particles
  - Large values : tendency to explode toward infinity
  - In general $\varphi_1 + \varphi_2 \leq 4$

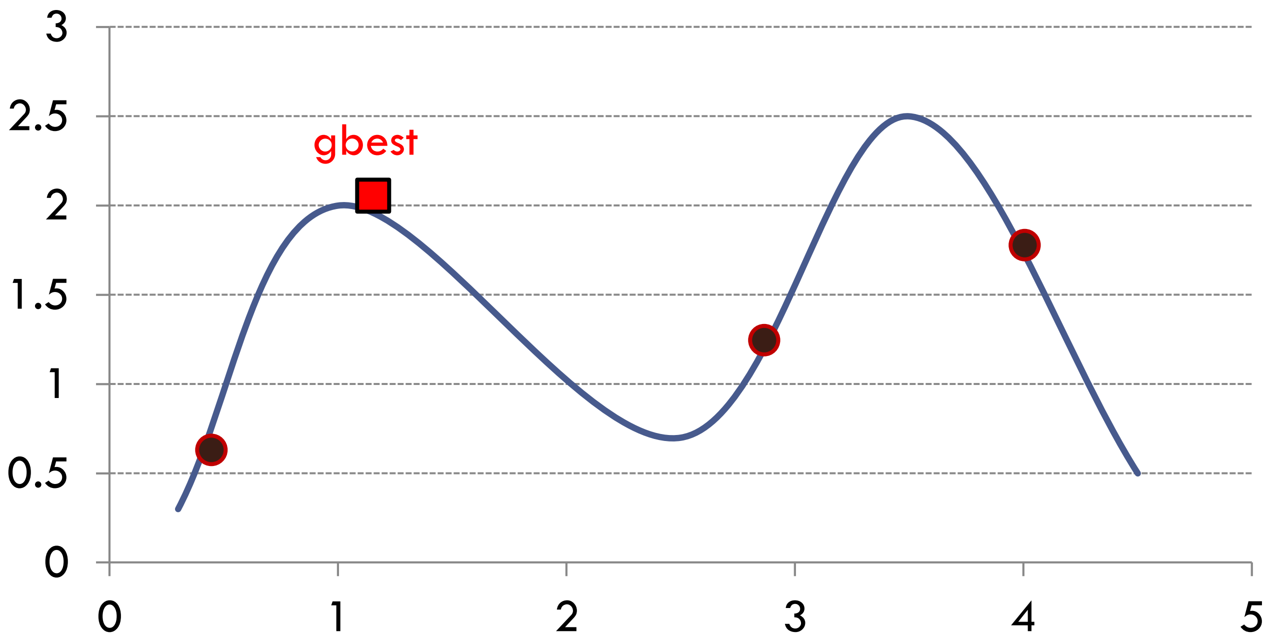


- Maximum velocity
  - Velocity is a stochastic variable => uncontrolled trajectory

$$\text{If} \quad v_{\text{id}} > v_{\max} \text{ then } v_{\text{id}} = v_{\max}$$
$$\text{else if } v_{\text{id}} < -v_{\max} \text{ then } v_{\text{id}} = -v_{\max}$$

Simple 1D Example

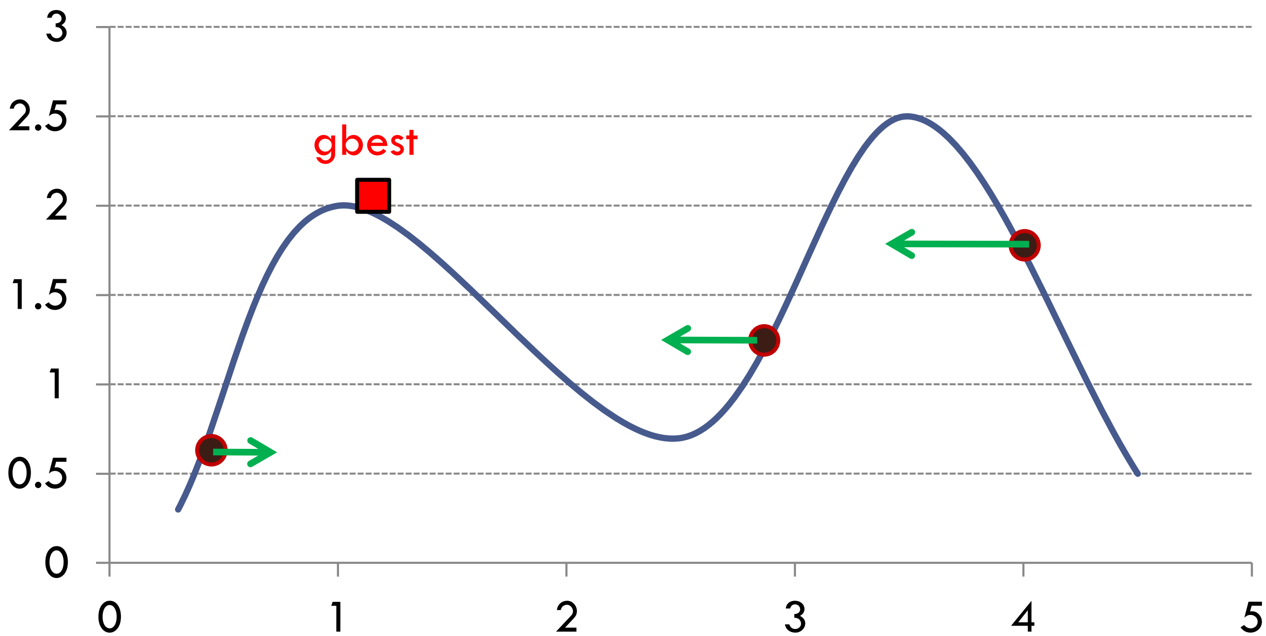
Initialize swarm and evaluate fitness (t=0)



Simple 1D Example

Update velocity and position (t=1)

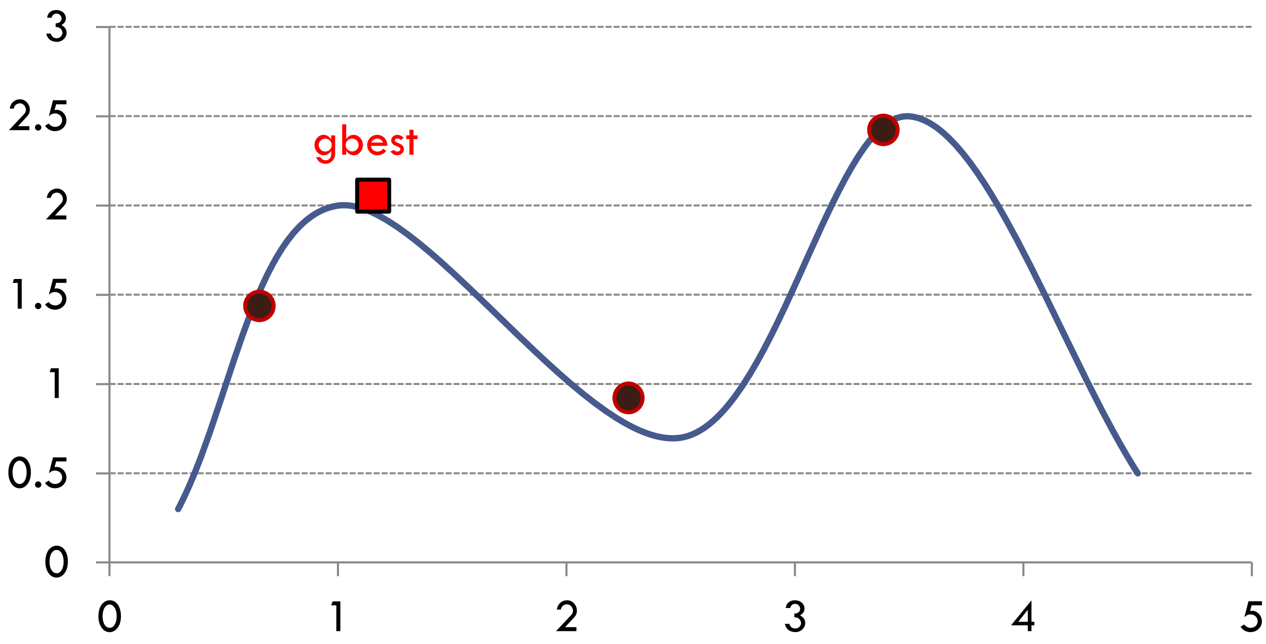
$$V_i^{t+1} = V_i^t + \varphi_1 . r_1 (P_i - X_i^t) + \varphi_2 . r_2 (P_g - X_i^t)$$



Simple 1D Example

Evaluate fitness

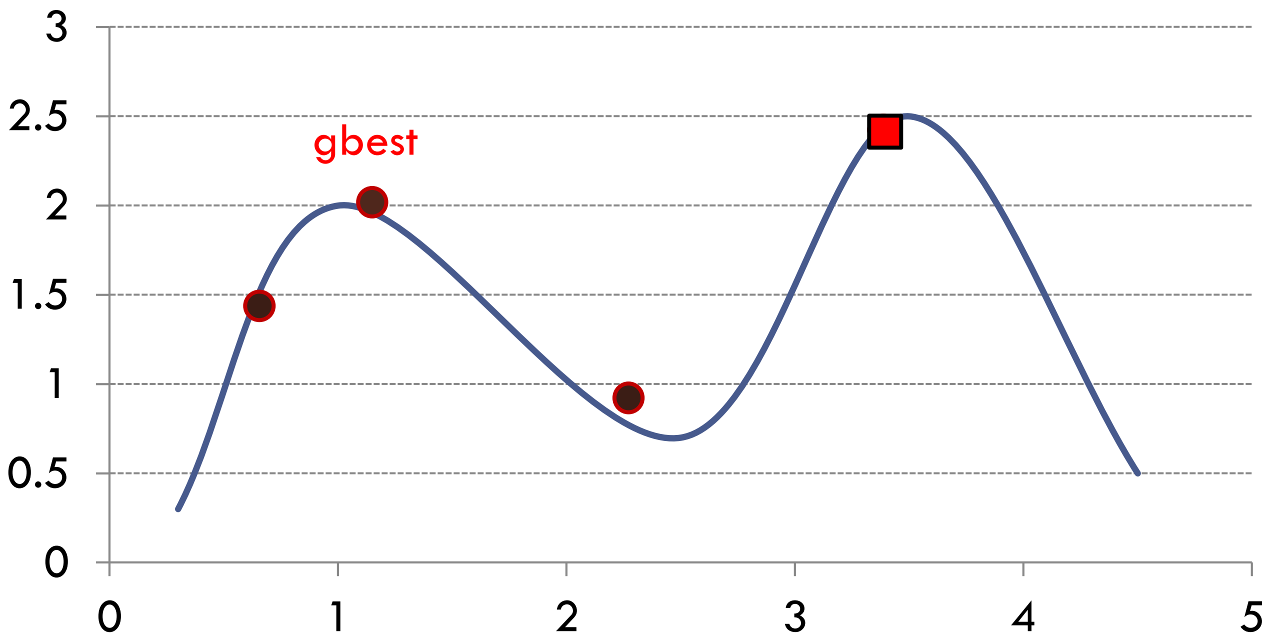
Update personal and global best (t=2)



Simple 1D Example

Evaluate fitness

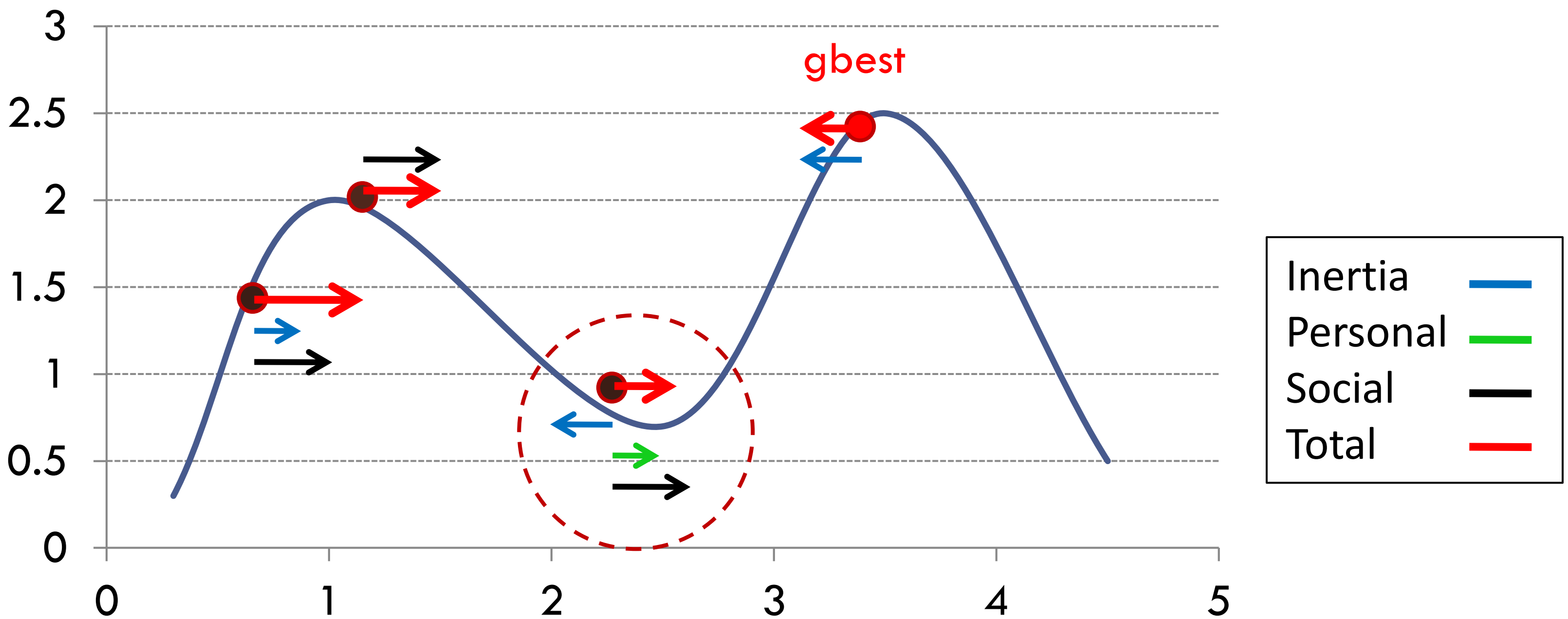
Update personal and global best (t=2)



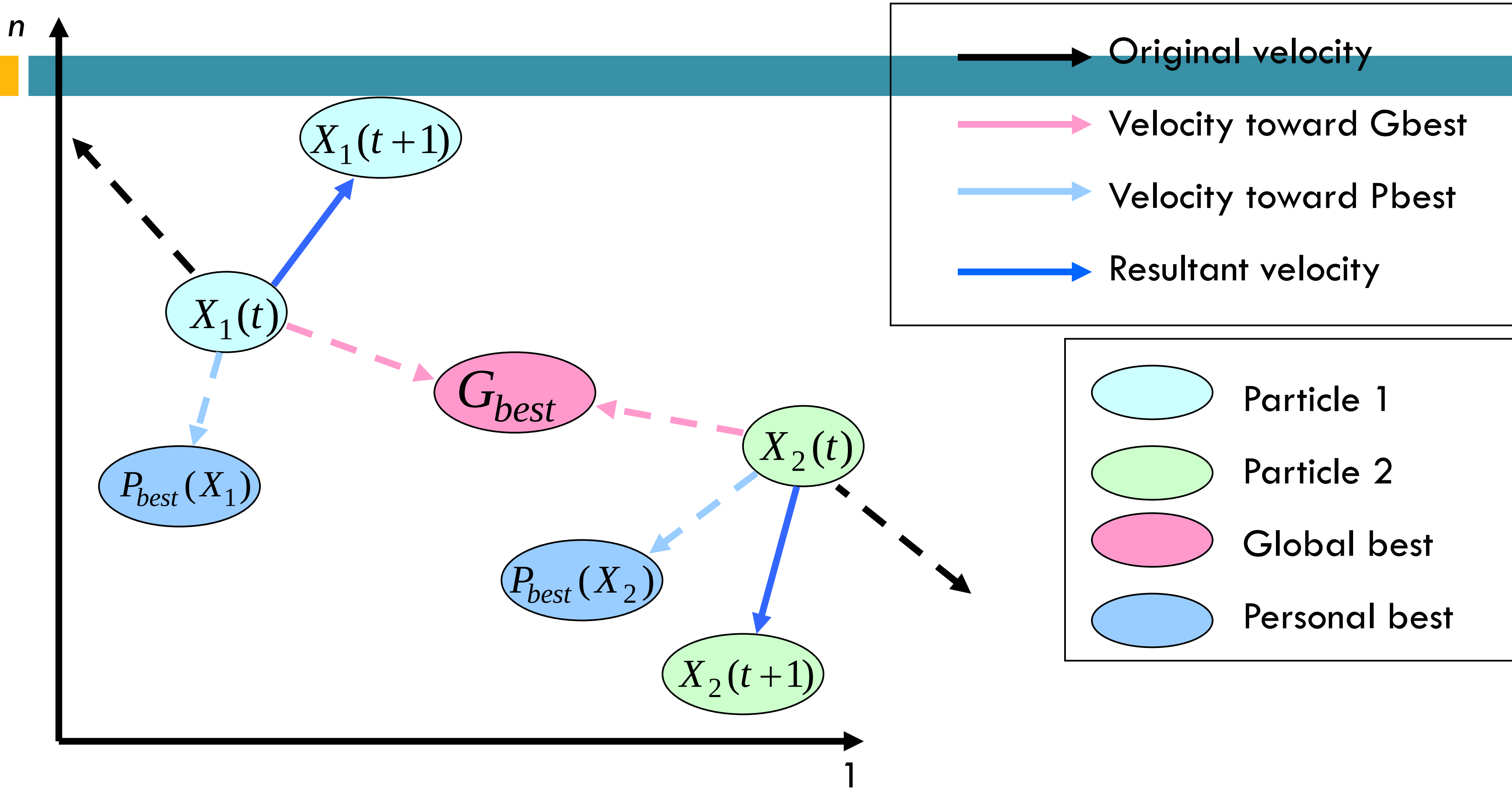
Simple 1D Example

Update velocity and position (t=2)

$$V_i^{t+1} = V_i^t + \varphi_1.r_1(P_i - X_i^t) + \varphi_2.r_2(P_g - X_i^t)$$



Movement of Particles



Individual particles (1 and 2) are accelerated toward the location of the global best solution (Gbest) and the location of their own personal best (Pbest) in the *n*-dimensional space.

PSO with Inertia Weight (w)

- **Inertia weight:**

$$V_i^{t+1} = w.V_i^t + \varphi_1.r_1(P_i - X_i^t) + \varphi_2.r_2(P_g - X_i^t)$$

- Scaling the previous velocity
- **Rate of Convergence Improvement**
- Control search behavior
  - High values → exploration
  - Low values → exploitation

PSO with Inertia Weight (w) (Cont.)

- Can be decreased over time:
  - Linear [0.9 to 0.4]
  - Nonlinear

$$\boxed{w(t) = \frac{A}{e^{t}}}$$

- Main disadvantage:
  - Once the inertia weight is decreased, the swarm loses its ability to search new areas (can not recover its exploration mode).

Rate of Convergence Improvement (Cont.)

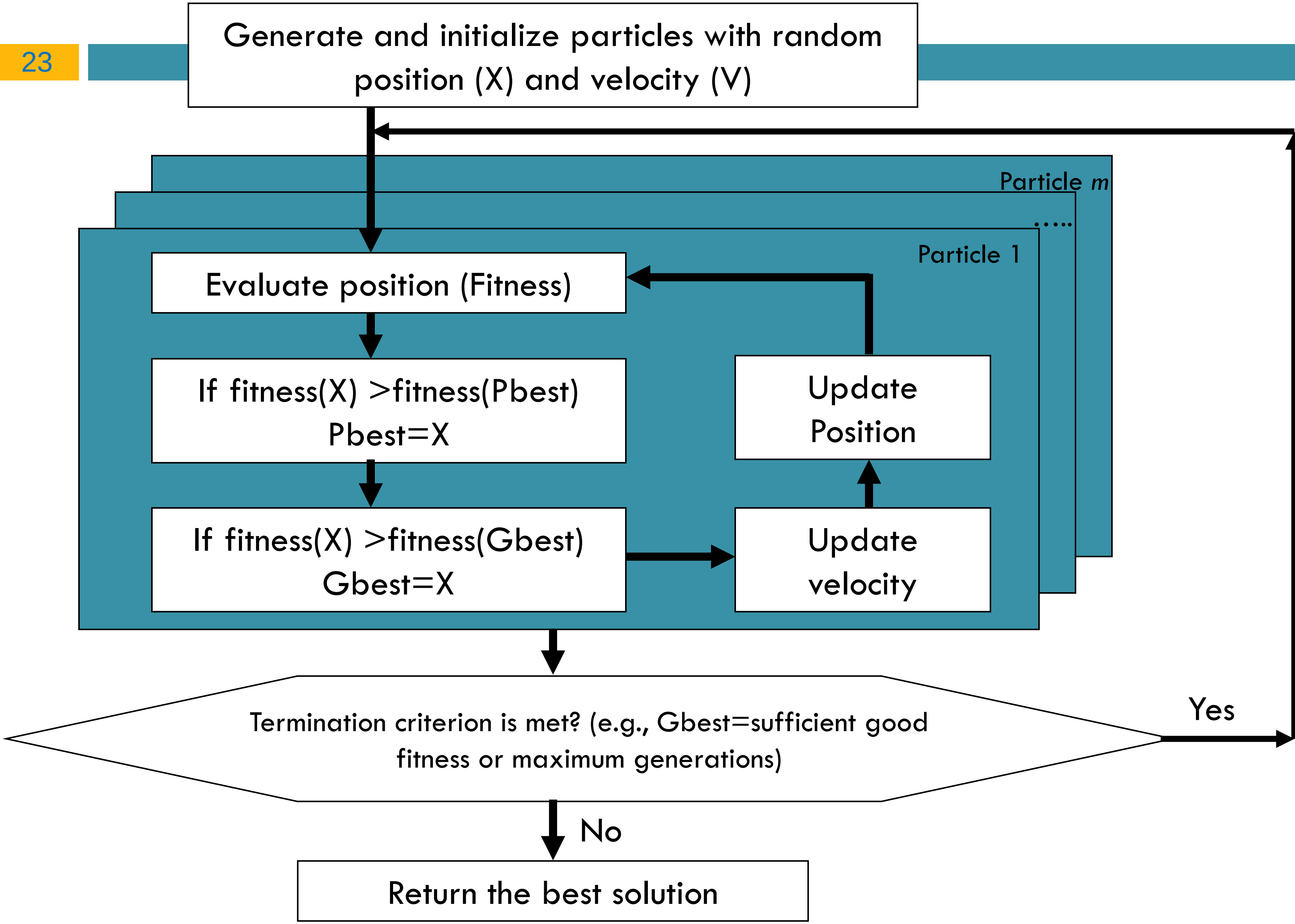
- **Constriction Factor:**
  - Canonical PSO

$$V_i^{t+1} = \chi.\left(V_i^t + \varphi_1.r_1(P_i - X_i^t) + \varphi_2.r_2(P_g - X_i^t)\right)$$

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \quad , \quad \varphi = \varphi_1 + \varphi_2 \, , \varphi > 4$$

  - Typically $\varphi = 4.1 \quad \chi = 0.729$
  - Can converge without using Vmax (velocity clamping)
  - Improve the convergence by damping the oscillations

The Flowchart of PSO



Overview of PSO with Pseudocode

```
For each particle
    Initialize particle
END

Do
    For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness value (pBest) in history
            set current value as the new pBest
    End

    Choose the particle with the best fitness value of all the particles as the gBest
    For each particle
        Calculate particle velocity according equation of updating velocity
        Update particle position according equation of updating position
    End
While maximum iterations or minimum error criteria is not attained
```

Pseudo-code for PSO

Aspects of PSO

$$V_i^{t+1} = w.V_i^t + \varphi_1.r_1(P_i - X_i^t) + \varphi_2.r_2(P_g - X_i^t)$$

- **Previous Velocity, V_i^t**
  - ✓Inertia component
  - ✓Memory of previous flight direction
  - ✓Prevents particle from drastically changing direction
- **Cognitive Component, $\varphi_1.r_1(P_i - X_i^t)$**
  - ✓Quantifies performance relative to past performances
  - ✓Memory of previous best position
  - ✓Nostalgia
- **Social Component, $\varphi_2.r_2(P_g - X_i^t)$**
  - ✓Quantifies performance relative to neighbors
  - ✓Envy

Aspects of PSO (Cont.)

$$V_i^{t+1} = w.V_i^t + \varphi_1.r_1(P_i - X_i^t) + \varphi_2.r_2(P_g - X_i^t)$$

Inertia weight (w) Controls the tendency of particles to keep searching in the same direction

- **For w ≥ 1**
  - ✓ Velocities increase over time
  - ✓ Swarm diverges
  - ✓ Particles fail to change direction towards more promising regions
- **For 0 < w < 1**
  - ✓ Particles decelerate
  - ✓ Convergence also depend on values of φ_1 and φ_2
- **Exploration–Exploitation**
  - Large values – Favor exploration
  - Small values – Promote exploitation
- **Problem-dependent**

Aspects of PSO (Cont.)

$$V_i^{t+1} = w.V_i^t + \varphi_1.r_1(P_i - X_i^t) + \varphi_2.r_2(P_g - X_i^t)$$

Exploration–exploitation tradeoff

- exploration – the ability to explore regions of the search space
- exploitation – the ability to concentrate the search around a promising area to refine a candidate solution
- c_1 vs c_2 (φ_1 vs φ_2) influence on the exploration–exploitation tradeoff

Aspects of PSO (Cont.)

$$V_i^{t+1} = w.V_i^t + \varphi_1.r_1(P_i - X_i^t) + \varphi_2.r_2(P_g - X_i^t)$$

- φ_1 and φ_2, together with r_1 and r_2, control the stochastic influence of the cognitive and social components on the overall velocity of a particle.
- The acceleration constants φ_1 and φ_2 are also referred to as trust parameters, where φ_1 expresses how much confidence a particle has in itself, while φ_2 expresses how much confidence a particle has in its neighbors.
- If $\varphi_1 > 0$ and $\varphi_2 = 0$, all particles are independent hill-climbers. Each particle finds the best position in its neighborhood by replacing the current best position if the new position is better. Particles perform a local search.
- If $\varphi_2 > 0$ and $\varphi_1 = 0$, the entire swarm is attracted to a single point P_g

Aspects of PSO (Comparing with GA)

- PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA).
- The system is initialized with a population of random solutions and searches for optima by updating generations.
- However, unlike GA, PSO has no evolution operators such as crossover and mutation.
- In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.
- Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust.
- PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.

PSO Variants

- Hybrid PSO
  - Incorporate the capabilities of other evolutionary computation techniques.
- Adaptive PSO
  - Adaptation of PSO parameters for a better performance.
- PSO in complex environments
  - Multiobjective or constrained optimization problems or tracking dynamic systems.
- Other variants
  - variations to the original formulation to improve its performance.

Hybrid PSO

- GA-PSO:
  - combines the advantages of swarm intelligence and a natural selection mechanism.
  - jump from one area to another by the selection mechanism → accelerating the convergence speed.
  - capability of "breeding".
  - replacing agent positions with low fitness values, with those with high fitness, according to a selection rate

Hybrid PSO

- EPSO:
  - Evolutionary PSO
  - Incorporates a selection procedure
  - Self-adapting of parameters

- The particle movement is defined as:

$$V_i^t = w'_{i1} . V_i^{t-1} + w'_{i2} . r_1 (P_i - X_i^{t-1}) + w'_{i3} . r_2 (P'_g - X_i^{t-1})$$

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

Hybrid PSO : EPSO

- Mutation of weights and global best:

$$w'_{ik} = w_{ik} + \tau . N(0,1)$$

$$P'_g = P_g + \tau' . N(0,1)$$

  - Learning parameters τ' و τ can be either fixed or dynamically changing as strategic parameters.

- Survival Selection:
  - Stochastic tournament.

Simplicity in PSO and Applications

- **Simplicity make PSO be applied in more and more fields**
  - Convenience of realization
  - Properties of low constraint on the continuity of objective function
  - Joint of search space
  - Ability of adapting to dynamic environment
  - ---------

- **Some PSO applications:**
  - Electronics and electromagnetic
  - Signal, Image and video processing
  - Neural networks
  - Communication networks
  - …

Year	IEEE Xplore
1995	(0)
1996	(0)
1997	(2)
1998	(3)
1999	(6)
2000	(10)
2001	(13)
2002	(36)
2003	(86)
2004	(270)
2005	(425)
2006	(687)

PSO Applications

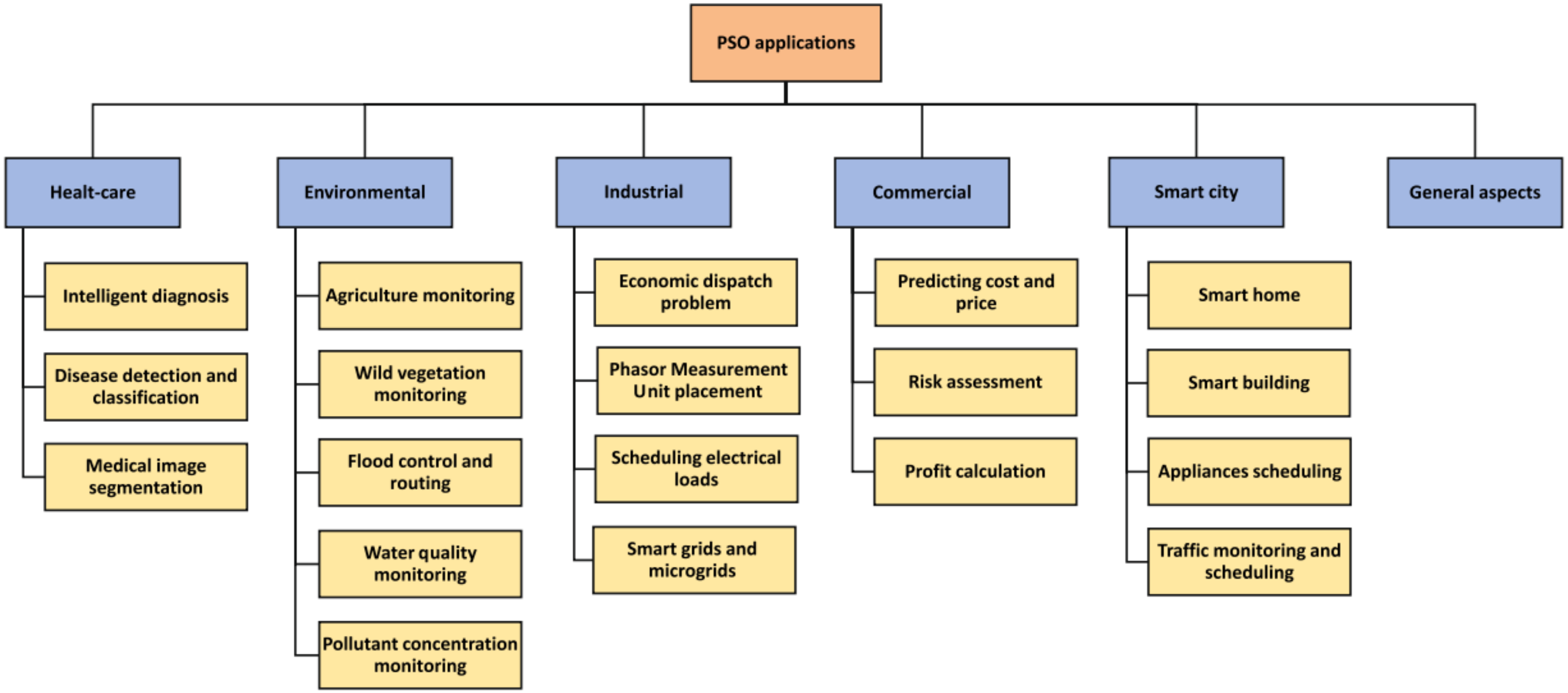
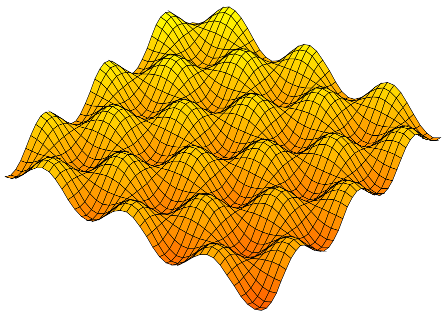


Fig. 5 The taxonomy of PSO applications

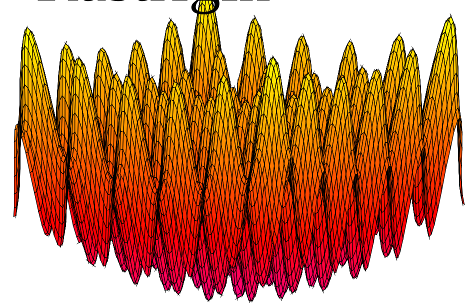
Gad, A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Arch Computat Methods Eng* **29**, 2531–2561 (2022). https://doi.org/10.1007/s11831-021-09694-4

Some functions …

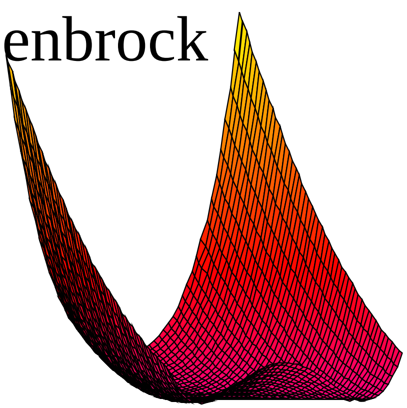
Griewank

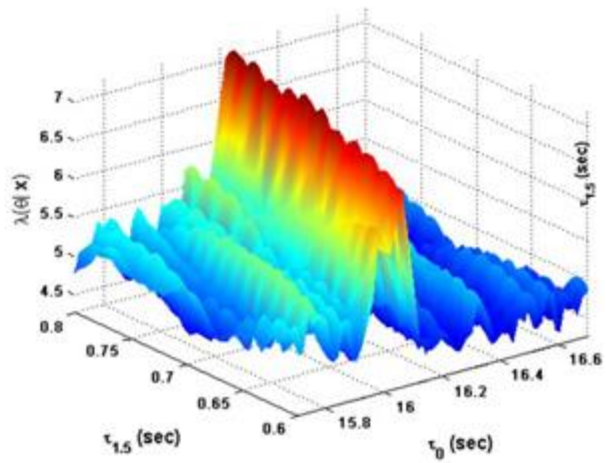


Rastrigin



Rosenbrock





A Detailed Problem Solving

- [TextBook] 2021 - Evolutionary Optimization Algorithms-CRC Press
- 5.7 Example -> Page 98

Thanks for your attention