MACHINE INTELLIGENCE

Swarm Intelligence

Dr. Arti Arya

Department of Computer Science

artiarya@pes.edu

+080-661896629 Extn 6629

MACHINE INTELLIGENCE

Swarm Intelligence

Dr. Arti Arya

Department of Computer Science and Engineering

What is a Swarm?

- A loosely structured collection of interacting agents
  - **Agents**:
    - Individuals that belong to a group (but are not necessarily identical)
    - They contribute to and benefit from the group
    - They can recognize, communicate, and/or interact with each other
- A swarm is better understood if thought of *as agents exhibiting a collective behavior.*

- An AI technique based on the collective behavior which is decentralized, self-organized systems.

- Self organized systems are robust, reliable and simple.

- Example of Self Organized systems:

- People commonly self-organize without a leader. For example, in many cultures people will naturally form a line to implement a system of fairness when waiting for something.

- Self-organization is the *basis for swarm robotics, a technique that involves small robots that cooperate to complete work as opposed to being centrally controlled.*

- Generally made up of agents who interact with each other and the environment.
- No centralized control structures.
- Based on group behavior found in nature.

Examples of Swarms in Nature

- Classic Example: Swarm of Bees

- Can be extended to other similar systems:
  - Ant colony
    - ✓ Agents: ants
  - Flock of birds
    - ✓ Agents: birds
  - Traffic
    - ✓ Agents: cars
  - Crowd
    - ✓ Agents: humans
  - Immune system
    - ✓ Agents: cells and molecules

Why Insects?

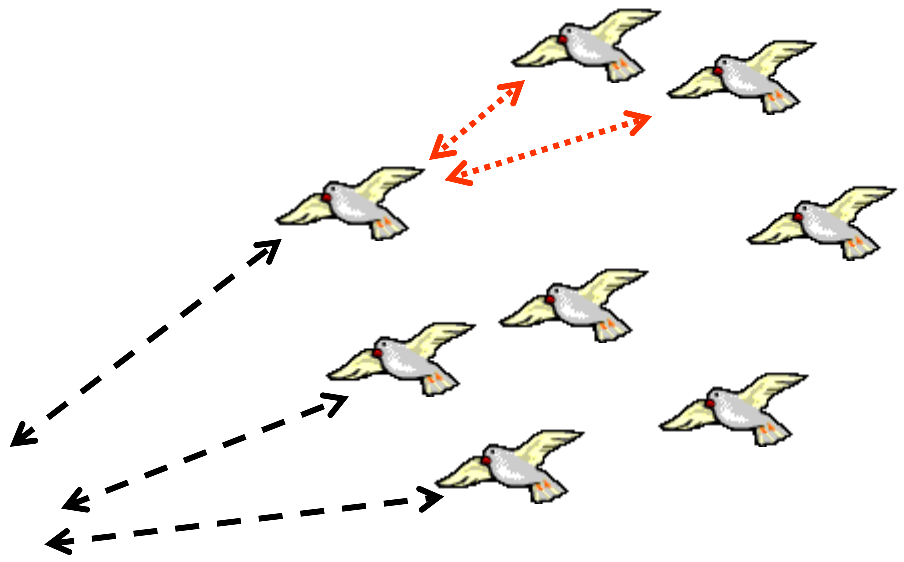
- Insects have a few hundred brain cells.

- However, organized insects have been known for:
  - ✓ Architectural marvels. (*Just check out 9 Wonderful Animals and Insects who are better at Architecture and buildings than humans)*
  - ✓ Complex communication systems.
  - ✓ Resistance to hazards in nature.

- In the 1950’s E.O. Wilson observed:
  - ✓ A single ant acts (almost) randomly – often leading to its own demise
  - ✓ A colony of ants provides food and protection for the entire population

Particle Swarm Optimization- PSO

- As described by the inventers James Kennedy and Russell Eberhart, "particle swarm algorithm imitates human (or insects) social behaviour.
- Individuals interact with one another while learning from their own experience, and gradually the population members move into better regions of the problem space".

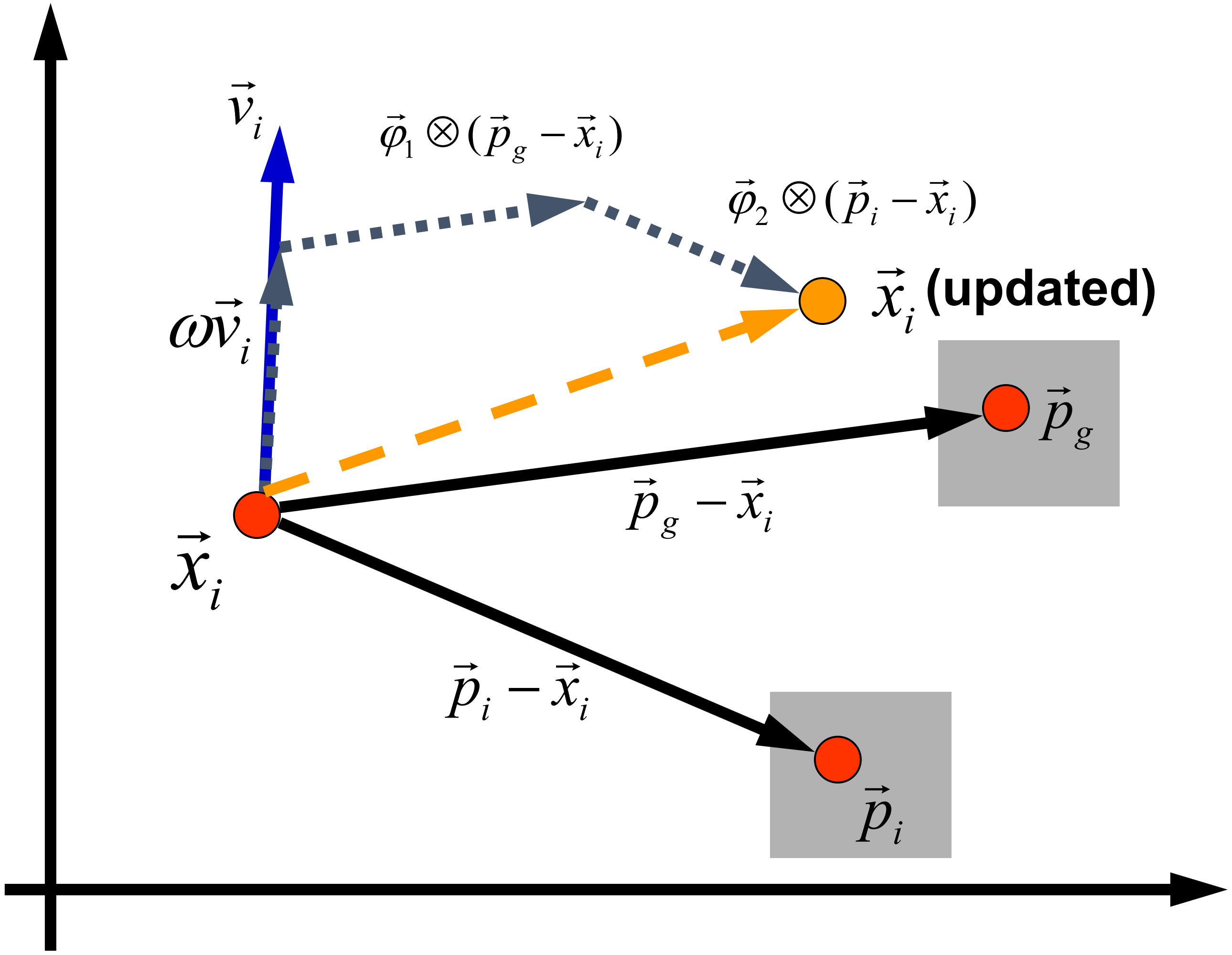


Why named as "particle", not "points"? Both Kennedy and Eberhart felt that velocities and accelerations are more appropriately applied to particles.

Particle Swarm Optimization- PSO

- A population based stochastic optimization technique.
- Searches for an optimal solution in the computable search space.
- Developed in 1995 by Dr. Eberhart and Dr. Kennedy.
- Inspiration: Swarms of Bees, Flocks of Birds, Schools of Fish

Visualizing- PSO



The Basic Idea

- ✓ Each particle is searching for the optimum.
- ✓ Each particle is *moving* and hence has a *velocity*.
- ✓ Each particle remembers the position it was in where it had its best result so far (its *personal best*)
- ✓ *But this would not be much good on its own; particles need help in figuring out where to search.*

The Basic Idea

- The particles in the swarm *co-operate*.
- They exchange information about what they've discovered in the places they have visited
- The co-operation is very simple. In basic PSO it is like this:
  - ✓ A particle has a *nbd* associated with it.
  - ✓ A particle knows the fitness of those in its nbd, and uses the *position* of the one with best fitness.
  - ✓ This position is simply used to adjust the particle's velocity.

- 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*.

- In PSO individuals strive to improve themselves and often achieve this by observing and imitating their neighbors.
- Each PSO individual has the ability to remember.
- PSO has simple algorithms and low overhead
  - ✓ Making it more popular in some circumstances than Genetic/Evolutionary Algorithms.
  - ✓ Has only one operation calculation:
    - Velocity: *a vector of numbers that are added to the position coordinates to move an individual*

Procedure of the Global Version

1. An array of population of particles with random positions and velocities on *d* dimensions in the problem space are initialized

2. Evaluate the fitness function in *d* variables for each particle.

3. Compare particle's fitness evaluation with particle's "pbest". If the current value is better than "pbest", then the current value is saved as the "pbest" and the "pbest" location corresponds to the current location in D−dimensional space.

4. Compare fitness evaluation with the population's overall previous best. If the current value is better than the "gbest", then current value is saved as "gbest" to the current particle's array index and value.

Suppose that the search space is D−dimensional, then the **i**

Suppose that the search space is D−dimensional, then the **i^{th} particle of the swarm** can be represented by a D−dimensional vector

$$X_{id} = (x_{i1}, x_{i2}, \ldots\ldots x_{iD})^T$$

The **velocity (position change) of this particle**, can be represented by another D−dimensional vector

$$V_{id} = (v_{i1}, v_{i2}, \ldots\ldots v_{iD})^T.$$

The **best previously visited position** of the i^{th} particle is denoted as

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

Procedure of the Global Version

Personal influence

5. Modify the **velocity and position** of the particle according to the following equations

$$v_{id}^{t+1} = v_{id}^{t} + c_1 rand_1 * (p_{id}^{t} - x_{id}^{t}) + c_2 rand_2 * (p_{gd}^{t} - x_{id}^{t}),$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}$$

momentum

Social influence

6. If the desired criterion is not met, go to step 2 otherwise stop the process. Usually the desired criterion may be a good fitness function or a maximum number of iterations.

Defining g as the index of the best particle in the swarm (i.e., the g^{th} particle is the best), and let the superscripts denote the iteration number, then the swarm is manipulated according to the above stated two equations,

- where d = 1, 2, …., D; i= 1, 2. . . N, and N is the size of the swarm
- c_1 & c_2 is a positive constants, called cognitive and social parameters.
- $rand_1$ and $rand_2$ are random numbers, uniformly distributed in [0, 1]
- t determines the iteration number

PES UNIVERSITY ONLINE

Mechanism to control the velocity

$$if \quad v_{id} > V_{\max}, \quad then \quad v_{id} = V_{\max} \qquad if \quad v_{id} < -V_{\max}, \quad then \quad v_{id} = -V_{\max}$$

PSO is able to locate the optimum area faster than the EC techniques, but fails in adjusting its velocity step size to continue the search for a finer grain. The problem is addressed by incorporating a weight parameter for the previous velocity of the particle

$$v_{id}^{t+1} = \varphi(wv_{id}^{t} + c_1 rand()^t * (p_{id}^t - x_{id}^t) + c_2 Rand()^t * (p_{gd}^t - x_{id}^t)),$$

where *w* is called inertia weight and ϕ is a constriction factor, which is used, alternatively to *w* to limit velocity, c_1 and c_2 are two positive constants, called cognitive and social parameter respectively.

*The inertia weighted PSO can converge under certain conditions without using **Vmax**.*

Parameters of PSO

- **pbest (p_{id}):** 'pbest' is the best position of the particle attained so far and can be considered as the particles memory and one memory slot is allotted to each particle.

- **'nbest' (p_{nd}) and 'gbest' (p_{gd}):** The best position that neighbours of a particle achieved so far is the 'nbest', where 'gbest' is the extreme of 'nbest', where it takes whole population as the neighbours of each particle.
- The selection of the 'nbest' consists of two phases. In the first phase the determination of the neighbourhood come into picture and in the second phase the selection of the 'nbest' is done.

- **Learning factors:** The constants c_1 and c_2 are the learning factors, which represent the weighting of the stochastic acceleration terms that pull each particle towards 'pbest' and 'nbest' positions.

Parameters of PSO

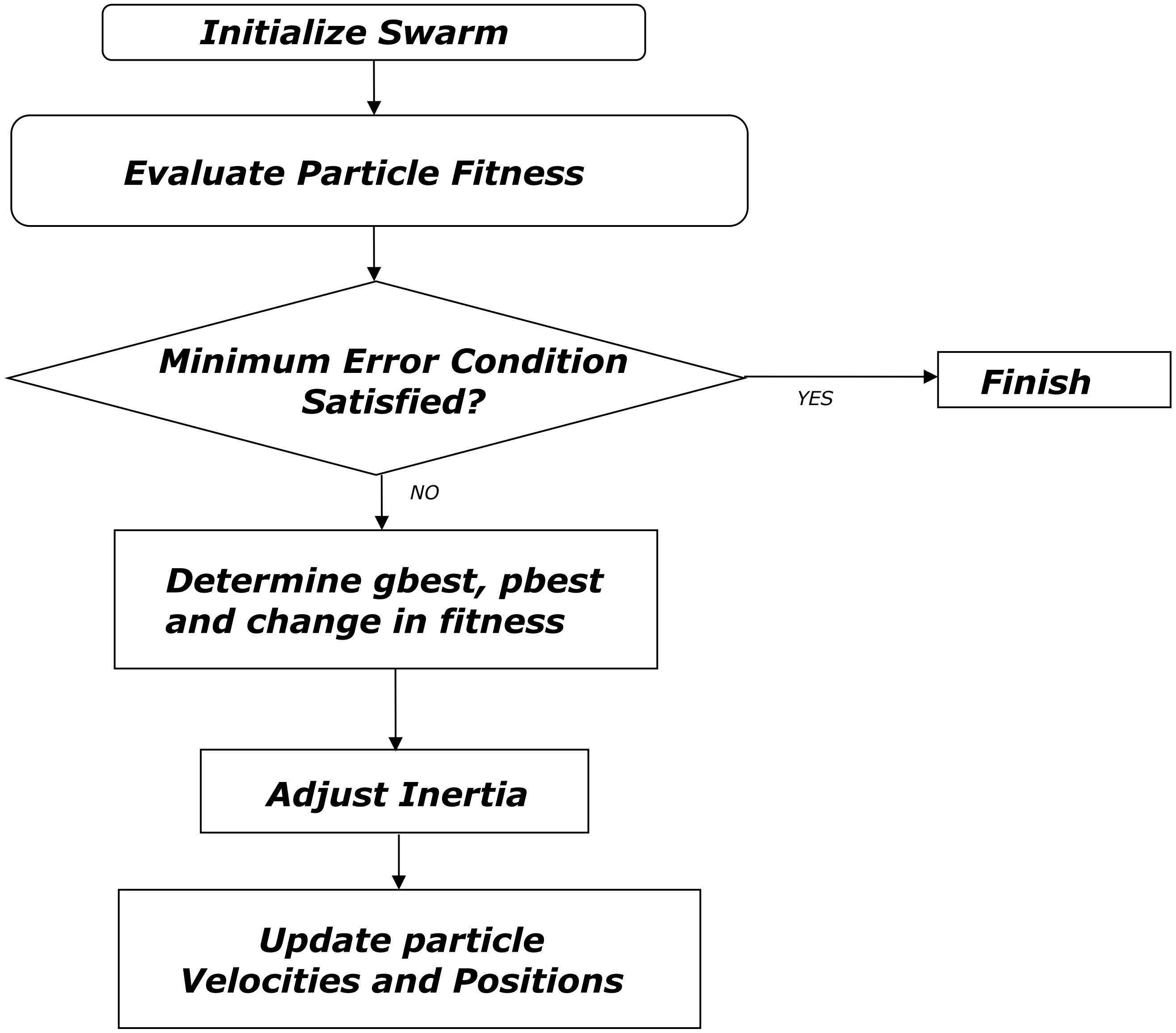
- **Inertia weight:** The inertia weight is initially set to a constant, but later experimental results suggested having a larger value initially, in order to promote global exploration of the search space, and gradually decrease it to get more refined solutions.

- **Constriction factor (ϕ):** Use of the constriction factor ϕ may be necessary to *ensure convergence of PSO.*

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

where $\psi = c_1 + c_2$, $\psi > 4$

Working of PSO



References

1. Some material of these slides are from Prof. Stephany Coffman-Wolph slides on PSO.

THANK YOU

Dr. Arti Arya
Department of Computer Science
artiarya@pes.edu
+080-66186629 Extn 6629