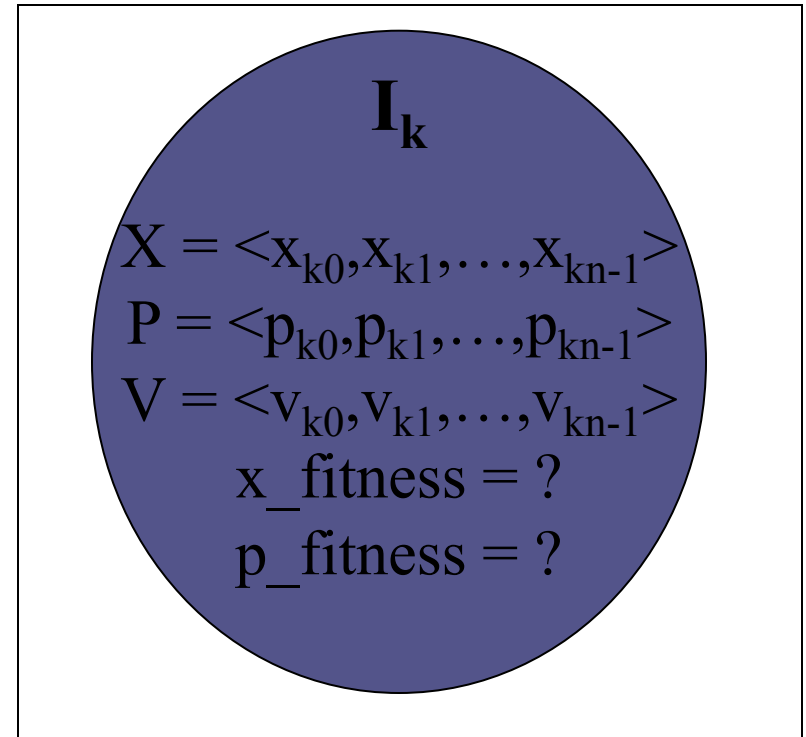


Particle Swarm Optimization

- Particle Swarm Optimization (PSO) applies the concept of social interaction to problem solving.
- It was developed in 1995 by James Kennedy and Russ Eberhart [Kennedy, J. and Eberhart, R. (1995). "Particle Swarm Optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942-1948, IEEE Press.] (<http://dsp.jpl.nasa.gov/members/payman/swarm/kennedy95-ijcnn.pdf>)
- It has been applied successfully to a wide variety of search and optimization problems.
- In PSO, a swarm of n individuals communicate either directly or indirectly with one another search directions (gradients).
- PSO is a simple but powerful search technique.

Particle Swarm Optimization: The Anatomy of a Particle

- A particle (individual) is composed of:
 - Three vectors:
 - The **x-vector** records the current position (location) of the particle in the search space,
 - The **p-vector** records the location of the best solution found so far by the particle, and
 - The **v-vector** contains a gradient (direction) for which particle will travel in if undisturbed.
 - Two fitness values:
 - The **x-fitness** records the fitness of the x-vector, and
 - The **p-fitness** records the fitness of the p-vector.



Particle Swarm Optimization: Swarm Search

- In PSO, particles never die!
- Particles can be seen as simple agents that fly through the search space and record (and possibly communicate) the best solution that they have discovered.
- So the question now is, “How does a particle move from one location in the search space to another?”
- This is done by simply adding the v-vector to the x-vector to get another x-vector ($X_i = X_i + V_i$).
- Once the particle computes the new X_i it then evaluates its new location. If x-fitness is better than p-fitness, then $P_i = X_i$ and p-fitness = x-fitness.

Particle Swarm Optimization: Swarm Search

- Actually, we must adjust the v-vector before adding it to the x-vector as follows:
 - $v_{id} = v_{id} + \phi_1 * \text{rnd}() * (p_{id} - x_{id}) + \phi_2 * \text{rnd}() * (p_{gd} - x_{id}) ;$
 - $x_{id} = x_{id} + v_{id};$
- Where **i** is the particle,
- ϕ_1, ϕ_2 are learning rates governing the **cognition** and **social** components
- Where **g** represents the index of the particle with the best p-fitness, and
- Where **d** is the d^{th} dimension.

Particle Swarm Optimization: Swarm Search

- Initially the values of the velocity vectors are randomly generated with the range $[-V_{max}, V_{max}]$ where V_{max} is the maximum value that can be assigned to any v_{id} .

Particle Swarm Optimization: Swarm Types

- In his paper, [Kennedy, J. (1997), "The Particle Swarm: Social Adaptation of Knowledge", Proceedings of the 1997 International Conference on Evolutionary Computation, pp. 303-308, IEEE Press.]
- Kennedy identifies 4 types of PSO based on ϕ_1 and ϕ_2 .
- Given:
$$v_{id} = v_{id} + \phi_1 * \text{rnd}() * (p_{id} - x_{id}) + \phi_2 * \text{rnd}() * (p_{gd} - x_{id}) ;$$
$$x_{id} = x_{id} + v_{id};$$
 - Full Model ($\phi_1, \phi_2 > 0$)
 - Cognition Only ($\phi_1 > 0$ and $\phi_2 = 0$),
 - Social Only ($\phi_1 = 0$ and $\phi_2 > 0$)
 - Selfless ($\phi_1 = 0, \phi_2 > 0$, and $g \neq i$)

Particle Swarm Optimization: Related Issues

- There are a number of related issues concerning PSO:
 - Controlling velocities (determining the best value for V_{max}),
 - Swarm Size,
 - Neighborhood Size,
 - Updating X and Velocity Vectors,
 - Robust Settings for (ϕ_1 and ϕ_2),
 - An Off-The-Shelf PSO
- Carlisle, A. and Dozier, G. (2001). “An Off-The-Shelf PSO”, *Proceedings of the 2001 Workshop on Particle Swarm Optimization*, pp. 1-6, Indianapolis, IN. (http://antho.huntingdon.edu/publications/Off-The-Shelf_PSO.pdf)

Particle Swarm: Controlling Velocities

- When using PSO, it is possible for the magnitude of the velocities to become very large.
- Performance can suffer if V_{max} is inappropriately set.
- Two methods were developed for controlling the growth of velocities:
 - A dynamically adjusted inertia factor, and
 - A constriction coefficient.

Particle Swarm Optimization: The Inertia Factor

- When the inertia factor is used, the equation for updating velocities is changed to:

$$- v_{id} = \omega * v_{id} + \phi_1 * \text{rnd}() * (p_{id} - x_{id}) \\ + \phi_2 * \text{rnd}() * (p_{gd} - x_{id}) ;$$

- Where ω is initialized to 1.0 and is gradually reduced over time (measured by cycles through the algorithm).

Particle Swarm Optimization: The Constriction Coefficient

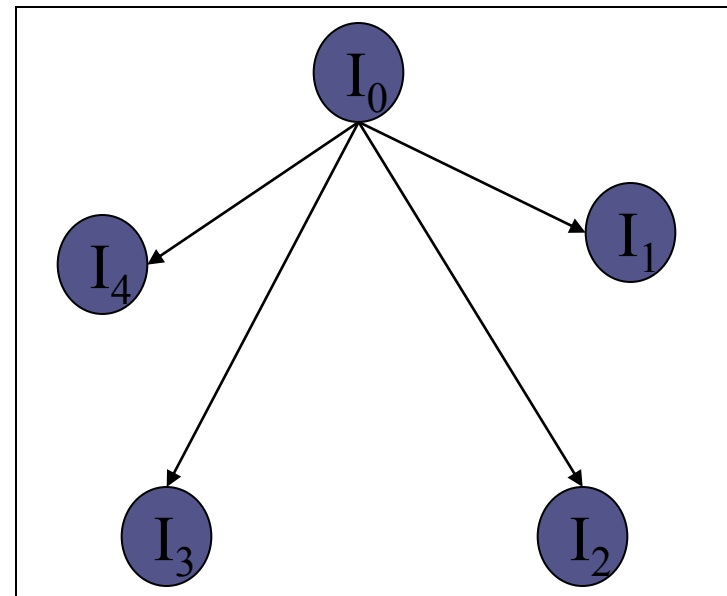
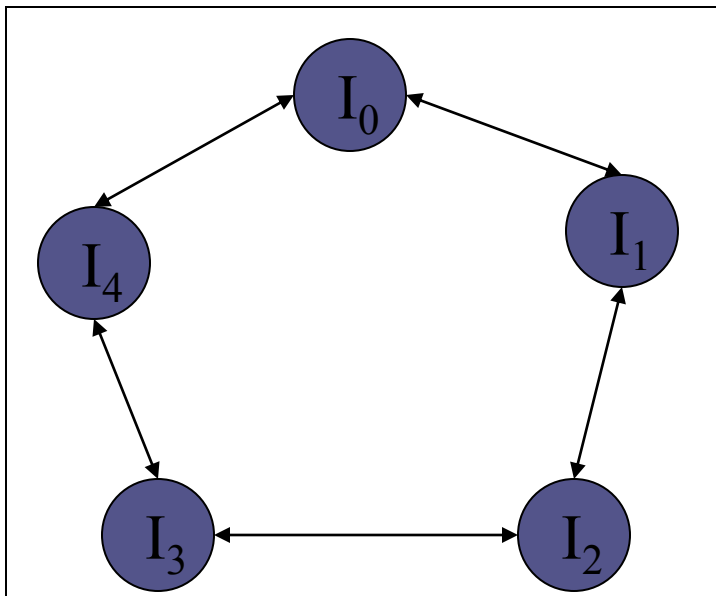
- In 1999, Maurice Clerc developed a constriction Coefficient for PSO.
 - $v_{id} = K [v_{id} + \varphi_1 * \text{rnd}() * (p_{id} - x_{id}) + \varphi_2 * \text{rnd}() * (p_{gd} - x_{id})] ;$
 - Where $K = 2 / |2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|,$
 - $\varphi = \varphi_1 + \varphi_2,$ and
 - $\varphi > 4.$

Particle Swarm Optimization: Swarm and Neighborhood Size

- Concerning the swarm size for PSO, as with other ECs there is a trade-off between solution quality and cost (in terms of function evaluations).
- Global neighborhoods seem to be better in terms of computational costs. The performance is similar to the ring topology (or neighborhoods greater than 3).
- There has been a lot of research on the effects of swarm topology on the search behavior of PSO/
 - Varies with the problem

Particle Swarm Optimization: Swarm Topology

- In PSO, there have been two basic topologies used in the literature
 - Ring Topology (neighborhood of 3)
 - Star Topology (global neighborhood)



Particle Swarm Optimization: Particle Update Methods

- There are two ways that particles can be updated:
 - Synchronously
 - *Asynchronously*
- Asynchronous update allows for newly discovered solutions to be used more quickly

