Particle Swarm Optimization

Marco A. Montes de Oca

IRIDIA-ULB
Optimization Group Meeting
November 30th, 2005

Agenda

- Definition
- Antecedents
- The Canonical PSO algorithm
- Variations
  - Local PSO
  - Inertia Weight
  - Constriction Factor
  - Fully Informed Particles
  - Probabilistic Particles
- Other Research Issues
- Past Research with Promising Future
- Future Research

Definition

Particle Swarm Optimization (PSO) is a population-based optimization technique in which individuals (called particles) represent a solution in an n-dimensional search space. Each particle has a velocity that allows it to “fly” over the search space.

Velocity vectors are updated every iteration considering particles' best past performance and that of their topological neighbors.

Antecedents

- Particle Systems

Particle systems have been used in computer graphics to model dynamic objects (e.g. smoke,fire, water). Particles behavior is controlled by a set of attributes such as initial position, initial velocity, lifetime, etc. Usually, the end result is the result of the interaction of particles' attributes and the environment's simulated physics. Each particle may also be governed by scripts so that different condictions in the environment lead to different behaviors.

Antecedents

- Simulated Bird Flocks

Scripting within particles was exploited by Reynolds to simulate the flocking behavior of birds. Each bird in a flock follows a set of behavioral rules that collectively allows the simulated birds to flock. There are three categories of behavioral rules in Reynolds's model :

- Collision avoidance
- Velocity matching*
- Flock centering*

* These two were considered in the initial development of PSO

Antecedents

- Social Learning Theory

Social sciences also inspired the design of PSO, in particular, Bandura's social learning theory.

In that theory, an individual observing a behavior (in another individual) being rewarded, will tend to imitate that behavior. Example: TV commercials.

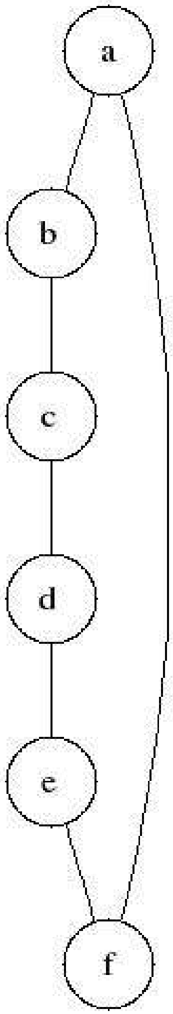
Canonical PSO algorithm

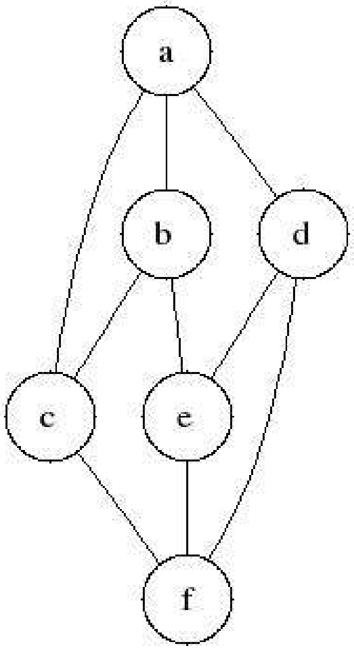
Algorithm 1 Pseudocode version of the original PSO algorithm

/* Initialization */
for $i = 1$ to k **do**
  Create particle i with random position and velocity
end for
Initialize $gbest$ and all $pbest_i$ to some sensible values

/* Main Loop */
$t = 0$
while $gbest$ is not good enough or $t < t_{max}$ **do**
  /* Evaluation Loop */
  for $i = 1$ to k **do**
    if $f(\vec{x}_i)$ is better than $pbest_i$ **then**
      $\vec{p}_i = \vec{x}_i$
      $pbest_i = f(\vec{x}_i)$
    end if
    if $pbest_i$ is better than $gbest$ **then**
      $gbest = pbest_i$
      $\vec{s} = \vec{p}_i$
    end if
  end for
  /* Update Loop */
  for $i = 1$ to k **do**
    Generate $\vec{\varphi}_1$ and $\vec{\varphi}_2$
    $\vec{v}_i = \vec{v}_i + \vec{\varphi}_1 * (\vec{p}_i - \vec{x}_i) + \vec{\varphi}_2 * (\vec{s} - \vec{x}_i)$
    $\vec{x}_i = \vec{x}_i + \vec{v}_i$
  end for
  $t = t + 1$
end while

Variations: Local PSO





In the local version of PSO, particles have a topological neighborhood. Topologies can be represented as graphs.

In this version, each particle only considers its neighborhood to update its velocity. Instead of considering gbest, a particle considers an lbest in the velocity update equation.

The global version is just a special case in which all particles are connected to each other.

Variations: Inertia Weight

In the early implementations, particles' velocities sometimes grew so much that particles moved away from the range in which they were supposed to search. A V_{max} was introduced as an extra parameter. It was observed that V_{max} could impose restrictions in the exploration-exploitation capabilities of the algorithm.

To handle this problem, V_{max} was (and is) usually set to X_{max}, the maximum of the dynamic range in each dimension, and an inertia weight factor was introduced.

The new velocity update rule becomes

$$\vec{v_i} = w\vec{v_i} + \vec{\varphi}_1 * (\vec{p_i} - \vec{x_i}) + \vec{\varphi}_2 * (\vec{s} - \vec{x_i})$$

Variations: Constriction Factor

To avoid explosion and ensure convergence, a constriction factor was included. The new velocity update rule became

$$\vec{v_i} = K[\vec{v_i} + \vec{\varphi_1} * (\vec{p_i} - \vec{x_i}) + \vec{\varphi_2} * (\vec{s} - \vec{x_i})]$$

where

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

$$k \in [0,1],\ \varphi = \varphi_1^{max} + \varphi_2^{max} \text{ and } \varphi > 4$$

Variations: Fully Informed Particles

The social component in the velocity update equation can be splitted among all topological neighbors. In this way, particles are fully informed since they use all the information available from its neighborhood.

The new velocity update equation is

$$\vec{v_i} = K\left[\vec{v_i} + \sum_{k \in \mathcal{N}} \vec{\varphi_k} * (\vec{p_k} - \vec{x_i})\right]$$

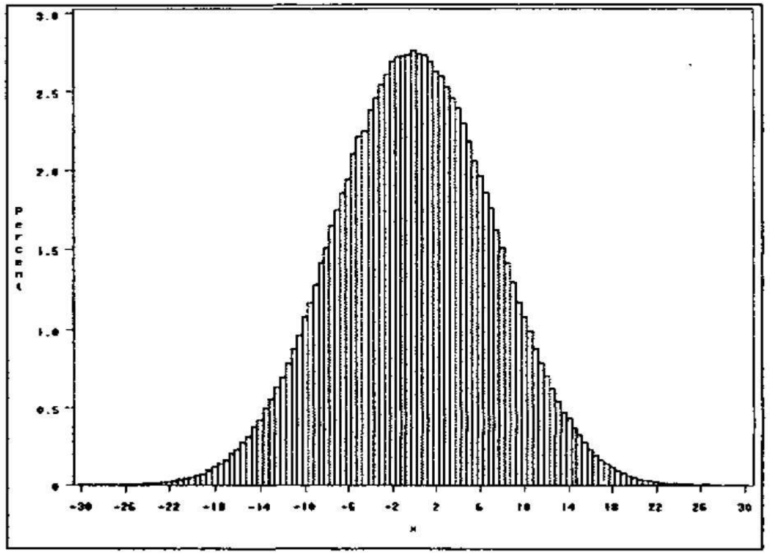
where

$$\vec{\varphi_k} = \vec{U}\left[0, \frac{\varphi_{max}}{|\mathcal{N}|}\right] \forall k \in \mathcal{N}$$

Variations: Probabilistic Particles

The way particles sample the search space is not clear from their behavioral rules. Experiments show that particles use some probability (not exactly normal) distribution centered around

$$\frac{\vec{p} + \vec{p_i}}{2}$$



Variations: Probabilistic Particles (continued)

This led to the design of a new PSO algorithm without velocity. In the new algorithm, particles move according to

$$\vec{x_i} = \vec{G}\left(\frac{\vec{p_g} + \vec{p_i}}{2}, |\vec{p_g} - \vec{p_i}|\right)$$

when trying to approximate the sampling through a Gaussian RNG. Other RNGs have been used. In general, they are as good (or as bad) as the canonical PSO algorithm.

Other Research Issues

We only discussed some variations of the PSO algorithm, but there has been much work on other aspects of the technique.

There is work on:

- Convergence analysis and parameter selection
- Neighborhood topologies
- Hybridizations
- Applications
- Other aspects (DEs, MOO, Communication, etc.)

Past Research with Promising Future

There is no real consensus on which is the best PSO version. Many researchers still use the canonical version.

Comparisons against other evolutionary techniques date back to the mid 90's.

Experiments using a small set of benchmark problems. Baggage from the past.

Undertaking a thorough re-study of PSO could help researchers and practitioners to select or even design their own PSO algorithms according to the characteristics of a given problem.

Future Research

Social learning in the real world happens through direct interactions (e.g. when we know someone personally), but also through indirect communication (e.g. when we read a book). In the PSO, only direct influences are present (through the topological arrangement).

In the real world, indirect communication allows us to reach more people in space and time than we could in person.

Could introducing indirect communication capabilities in PSO be beneficial? In which circumstances?

Thank you