Particle Swarm Optimization: Part 2

Advanced Topologies

Neighborhoods are usually formed on the basis of particle indices, however others exist. We will go over

- Spatial Social Networks
- Fitness-Based Spatial Neighborhoods
- Growing Neighborhoods

Spatial Social Networks

Suganthan proposed that neighborhoods be formed on the basis of the Euclidean distance between particles. The neighborhood, $\mathcal{N}_i$, of particle i of size $\|\mathcal{N}_i\|$, is defined to consist of the $\|\mathcal{N}_i\|$ particles closest to particle i.

- Any norm can be used, but generally the euclidean norm is used.
- The approach is computability expensive.
  - There are $\binom{\|\Omega\|}{2} = \frac{n(n-1)}{2}$ distances to calculate on each iteration.
- Determining neighborhoods based on distances has the advantage that neighborhoods change dynamically with each iteration.

Fitness-Based Spatial Neighborhoods

Braendler and Hendtlass proposed a variation on the spatial neighborhoods, which takes into account both

- relative distance between particles
- fitness of particles you choose to have in your neighborhood.

Specifically, $\mathcal{N}_i$ particles are chosen with the smallest:

$$\|\mathbf{x}_i - \mathbf{x}_{i'}\| f(\mathbf{x}_{i'}) \tag{1}$$

where $i' \neq i$ is the index of another particle in the swarm.

Fitness-Based Spatial Neighborhoods

$$\|\mathbf{x}_i - \mathbf{x}_{i'}\| f(\mathbf{x}_{i'}) \tag{2}$$

What could be a problem with this approach?

Growing Neighborhoods

Social networks with a low interconnection are thought to converge slower, which allows larger parts of the search space to be explored. Convergence of the fully interconnected star topology is faster, but at the cost of neglecting parts of the search space.
Suganthan, proposed that

- The topologies should start off with a low level of connectivity
- but approach the star (fully connected topology) towards the end of the run.

Growing Neighborhoods

Specifically, Suganthan proposed that if

$$\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{d_{max}} < \epsilon(t) \tag{3}$$

then particle $\mathbf{x}_j$ should be added to the neighborhood of particle i. Note that if equation (3) was not true in a subsequent iteration, the particle would not be removed. The neighborhood only grows.

- $\epsilon(t)$ would start off small and approach or exceed 1 towards the end of the run.
- why?

Variants of PSO

There have been numerous PSO variant proposed that alter the way the PSO update equations work while preserving the core ideas of the technique. We will discuss

- Fully Informed PSO
- Unified PSO
- Barebones PSO
- SPSO2011
- EA inspired PSOs
- DE inspired PSO

Fully Informed PSO

The FIPS algorithm was inspired by the observation made by Kennedy and Mendes, that in human society individuals are not influenced by only a single individual, but rather by a statistical summary of the state of their neighborhood.

- Based on this observation, the velocity equation of PSO is altered such that each particle is influenced by the successes of all its neighbors, and not on the performance of only one other particle in the neighborhood.

Fully Informed PSO

The velocity and position update equations of FIPS are defined as follows:

$$\boldsymbol{v}_i(t+1) = w\boldsymbol{v}_i(t) + \sum_{m=1}^{|\mathcal{N}_i|} \gamma_m(t) \frac{(\boldsymbol{y}_m(t) - \boldsymbol{x}_i(t))}{|\mathcal{N}_i|}, \tag{4}$$

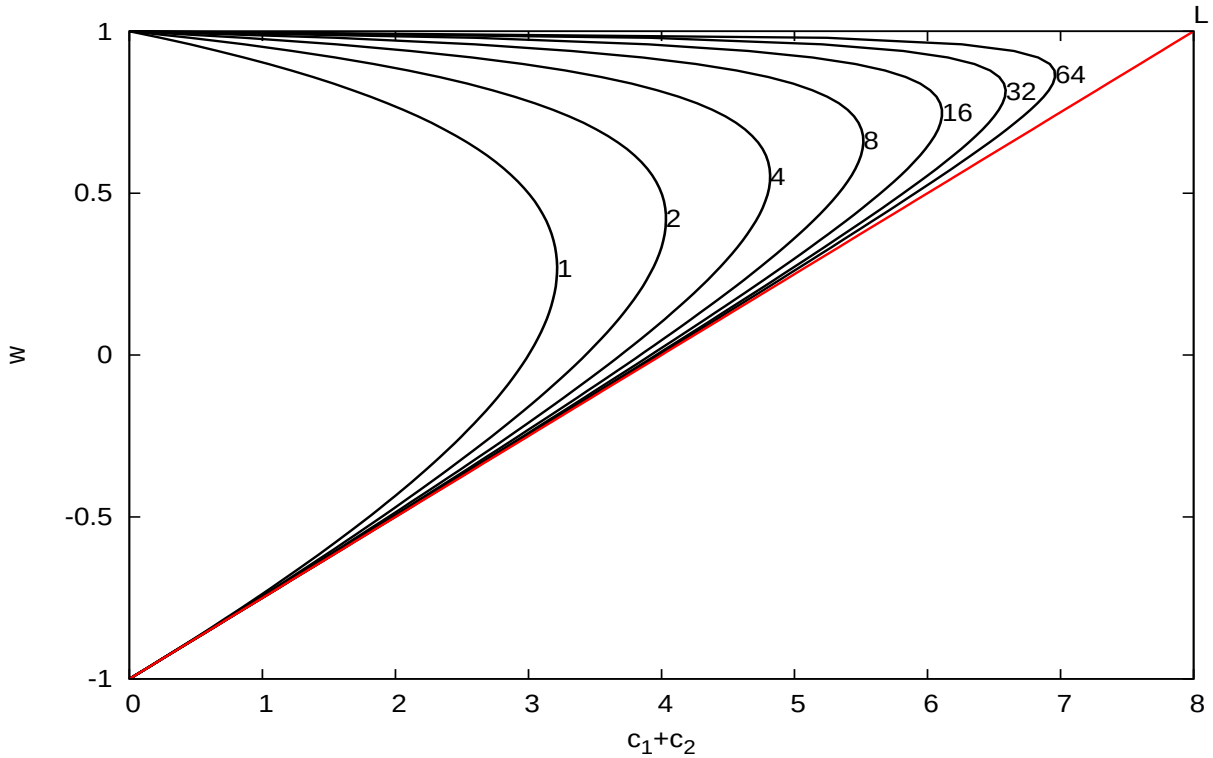
$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1), \tag{5}$$

- where $\mathcal{N}_i$ is the set of particles in particle i's neighborhood,
- and $\gamma_m(t) \in U(0, c_1 + c_2)^d$, where d is the dimension.
- The position $\boldsymbol{y}_m(t)$ represents the "best" position that particle m has visited

Fully Informed PSO

Criteria for order-1 and order-2 stability (with $c_1 = c_2$).

$$0 < c_1 + c_2 < \frac{12|\mathcal{N}_i|\left(1 - w^2\right)}{3|\mathcal{N}_i| + 1 + w\left(1 - 3|\mathcal{N}_i|\right)} \text{ and } |w| < 1 \tag{6}$$



Unified PSO

The UPSO algorithm was developed by Parsopoulos and Vrehatis. The authors argued that it would be beneficial to be able to utilize the exploitative nature of global best PSO (Gbest PSO) and the exploratory nature of local best PSO (Lbest PSO) in one unified scheme.

- Based on this idea, the velocity update equations of PSO was altered to be a combination of the Gbest and Lbest velocity update equations.
- The weighting of Lbest and Gbest is controlled by an additional control parameter, $u \in [0, 1]$, called the unification factor.

Unified PSO

The velocity and position update equation of UPSO are defined as follows:

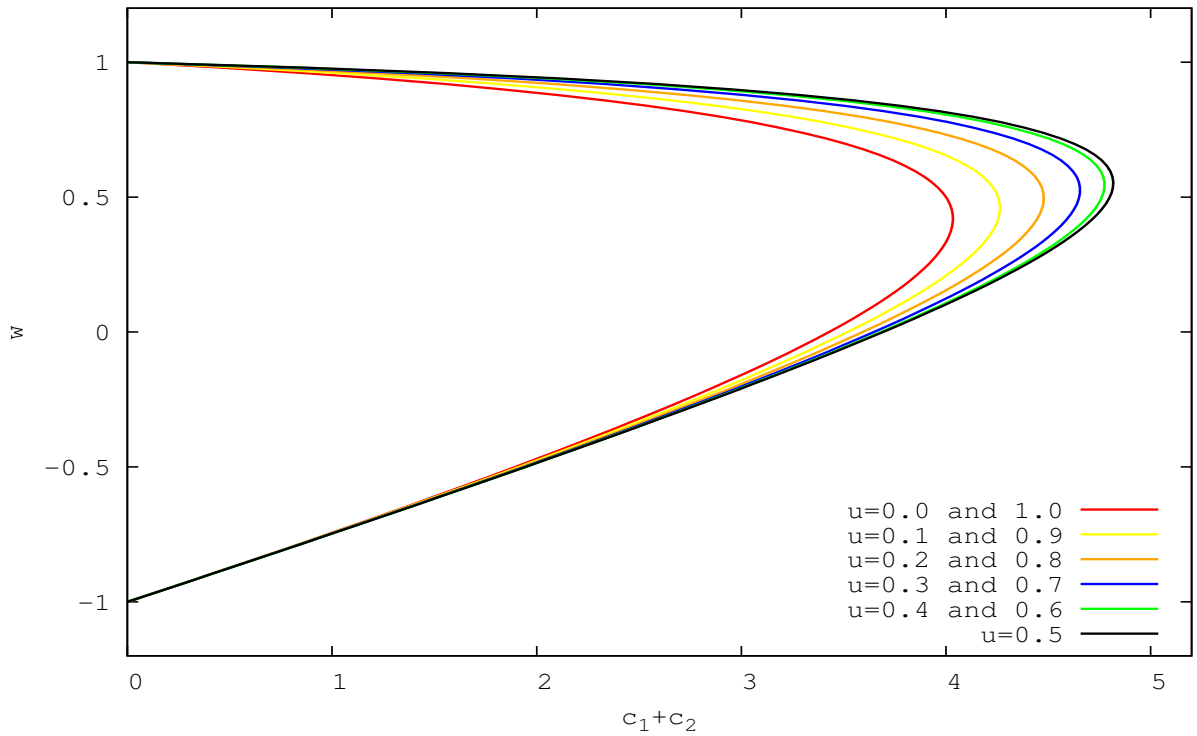
$$\begin{aligned}
\mathbf{g}_i(t+1) &= w\mathbf{v}_i(t) + c_1\mathbf{r_1}(\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2\mathbf{r_2}(\mathbf{g}(t) - \mathbf{x}_i(t)) \\
\mathbf{l}_i(t+1) &= w\mathbf{v}_i(t) + c_1\mathbf{r}'_1(\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2\mathbf{r}'_2(\mathbf{n}_i(t) - \mathbf{x}_i(t)) \\
\mathbf{v}_i(t+1) &= u\mathbf{g}_i(t+1) + (1-u)\,\mathbf{l}_i(t+1) \qquad (7) \\
\mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t+1), \qquad (8)
\end{aligned}$$

- where $\mathbf{r_1}, \mathbf{r_2}, \mathbf{r}'_1, \mathbf{r}'_2 \sim U(0,1)^d$,
- The positions $\mathbf{p}_i$ and $\mathbf{n}_i$ are the "best" positions that particle i and particle i's neighborhood of particles have visited.
- The position $\mathbf{g}$ is the "best" position found by the swarm so far.

Unified PSO

Criteria for order-1 and order-2 stability (with $c_1 = c_2$).

$$0 < c_1 + c_2 < \frac{24(1 - w^2)}{7 - 5w + 2(u^2 - u)(1 + w)} \text{ and } |w| < 1 \tag{9}$$



Barebones PSO

Kennedy proposed the Barebones PSO (BPSO) algorithm based on the empirical observation that the distribution of particle positions are centered around the weighted average between the personal best and neighborhood best positions, specifically

$$\zeta_i = \frac{c_1 \boldsymbol{p}_i(t) + c_2 \boldsymbol{n}_i(t)}{c_1 + c_2}. \tag{10}$$

Barebones PSO

For BPSO, the velocity update equation changes to

$$v_{i,j}(t+1) \sim N(\zeta_{i,j}, \phi_{i,j}(t)), \tag{11}$$

where $\phi_{i,j}(t) = |p_{i,j}(t) - n_{i,j}(t)|$.

The position update equation is changed to

$$\boldsymbol{x}_i(t+1) = \boldsymbol{v}_i(t+1). \tag{12}$$

Barebones PSO

In the standard implementation of BPSO , c_1 and c_2 are assumed to be equal resulting in the point of convergence being

$$\frac{\boldsymbol{p}_i(t) + \boldsymbol{n}_i(t)}{2}. \tag{13}$$

Barebones PSO

Kennedy also proposed an alternative to the barebones PSO velocity:

$$v_{i,j}(t+1) \quad \begin{cases} = p_{i,j} & \text{if } U(0,1) < 0.5 \\ \sim N(\zeta_{i,j}, \phi_{i,j}(t)) & \textit{else} \end{cases}, \tag{14}$$

where $p_{i,j}$ is the jth component of ith personal best position.

SPSO2011

Clerc developed SPSO2011 in an attempt to define a new baseline for future PSO improvements. The two primary benefits of the SPSO2011 are stated to be

- rotational invariance
- adaptive topology.

The particle velocity update equation is defined as follows:

$$\boldsymbol{v}_i(t+1) = w\boldsymbol{v}_i(t) + \mathcal{H}_i(\boldsymbol{g}_i(t), ||\boldsymbol{g}_i - \boldsymbol{x}_i||_2) - \boldsymbol{x}_i(t), \quad (15)$$

where $\boldsymbol{g}_i$ is defined as

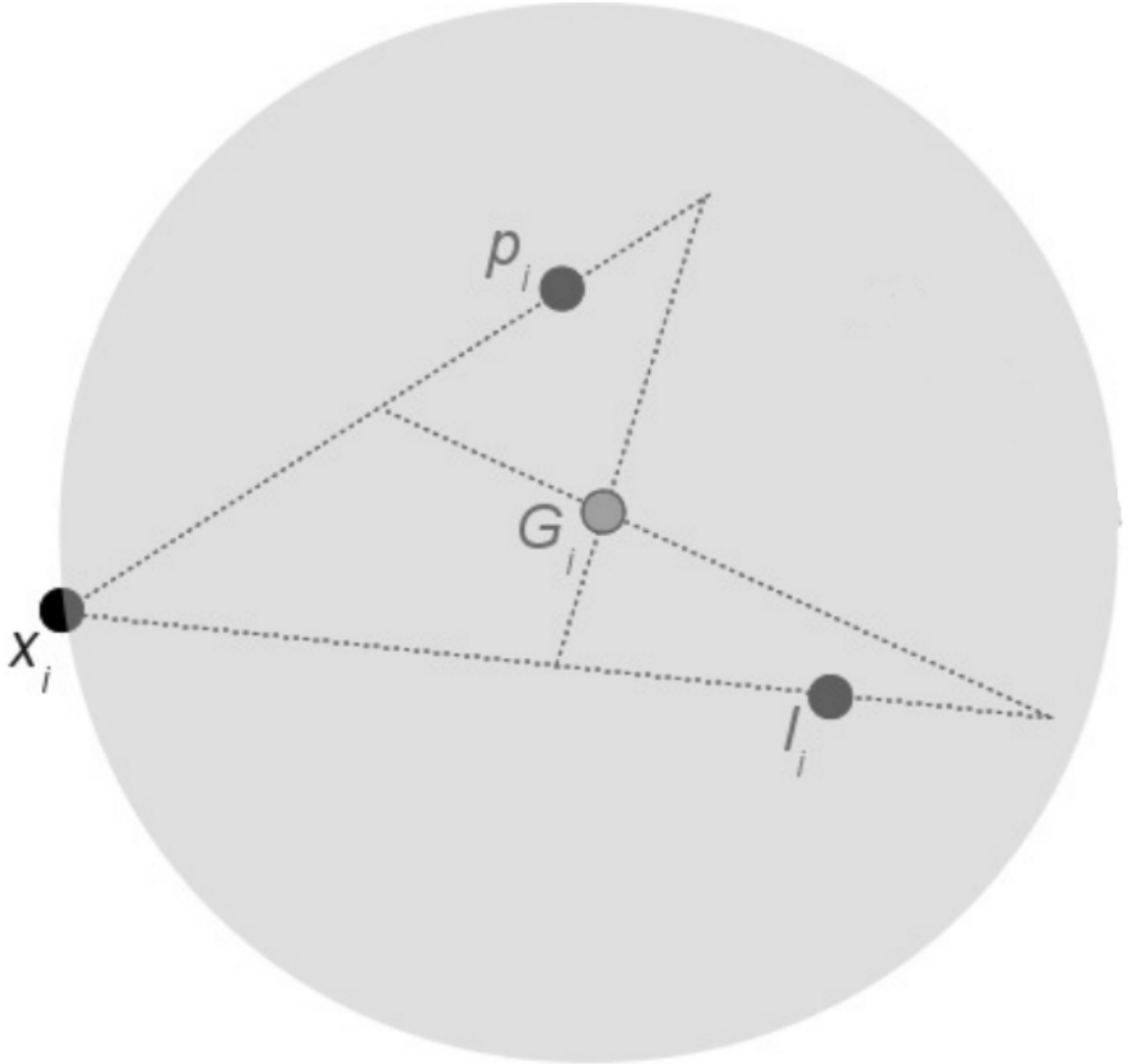
$$\boldsymbol{g}_i(t) = \frac{\boldsymbol{x}_i(t) + \boldsymbol{\alpha}_i(t) + \boldsymbol{\beta}_i(t)}{3}, \quad (16)$$

where $\boldsymbol{\alpha}_i(t)$ and $\boldsymbol{\beta}_i(t)$ are defined as

$$\boldsymbol{\alpha}_i(t) = \boldsymbol{x}_i(t) + c_1\boldsymbol{r}_1 \otimes (\boldsymbol{p}_i(t) - \boldsymbol{x}_i(t)), \quad (17)$$

$$\boldsymbol{\beta}_i(t) = \boldsymbol{x}_i(t) + c_2\boldsymbol{r}_2 \otimes (\boldsymbol{n}_i(t) - \boldsymbol{x}_i(t)). \quad (18)$$

SPSO2011



SPSO2011

The function $\mathcal{H}_i\left(\boldsymbol{g}_i\left(t\right), ||\boldsymbol{g}_i - \boldsymbol{x}_i||_2\right)$ returns a uniformly sampled random position from a hyper-sphere centered at $\boldsymbol{g}_i\left(t\right)$ with a radius of $||\boldsymbol{g}_i - \boldsymbol{x}_i||_2$.

Constructing a point in d-dimension hyper-sphere distribution

- Given center $\boldsymbol{ce}$ and radius ra of the spherical distribution, $\mathcal{H}(\boldsymbol{ce}, ra)$ is calculated as follows.
- Generate d scalars from the normal distribution $N\left(0, 1\right)$.
- Construct the vector $\boldsymbol{re}$ from the d scalars.
- Normalize $\boldsymbol{re}$
- Generate the scalar s from $U\left(0, ra\right)$.
- Set $\boldsymbol{re}$ to $s(\boldsymbol{re}) + \boldsymbol{ce}$.
- Return $\boldsymbol{re}$

SPSO2011

The topology used by SPSO2011 is a particular case of the stochastic star topology proposed by Mirinda et al.

- On initialization, each particle's neighborhood is constructed by selecting three particles randomly from the swarm and the particle itself (the same particle is allowed to be chosen several times).
- If an unsuccessful iteration occurs, the neighborhoods are reconstructed.
  - An unsuccessful iteration is defined as an iteration where no new position was found that improved the previous best objective evaluation obtained by the whole swarm.

Hybrid Algorithms

The are numerous PSO variants that are constructed as hybrids of PSO and evolutionary computation techniques. We will discuss the following ideas that have been merged into PSO

- Selection-Based PSO
- Reproduction
- Mutation
- Differential Evolution Based PSO

Selection-Based PSO

A common mechanism utilized within EA is that of selection. Specifically:

- Select which individuals of the population live or dies.
- Select which individuals of the population can procreate.

Angeline provided the first approach to combine GA inspired selection concepts with PSO.

Selection-Based PSO

The approach proposed by Angeline was, after the position updates where applied

- Calculate the fitness of all particles;
- for each particle i
  - Select α random distinct particles
  - For each selected particle with a "worse" fitness than i award particle i a point.
- Sort the swarm based on there scores
- Replace the bottom half with the top half, without changing the personal best positions.

Selection-Based PSO

Disadvantage:

- Diversity rapidly decrease

A simple improvement is to

- Rather replace the worse individuals with the mutated copies of the best individuals.

Reproduction in PSO

One of the first PSO's to use the concept of reproduction is Cheap-PSO by Clerc, where a particle can preform one of three actions

- to generate a new particle (asexual reproduction),
- kill itself,
- modify the inertia and acceleration coefficient, on the basis of environment conditions.

Reproduction in PSO

Koay and Srinivasan utilized the concept of reproduction, in the Global Best Spawning algorithm. The idea being to allow the best particle to spawn multiple offspring until an improved offspring was found to replace the parent

- Offspring where created by adding a normally distributed vector to the current position.

The advantage of this approach is that it will inheriting have local convergence properties, but it is very computationally expensive.

- Often times a spawn cycle limit is imposed

Reproduction in PSO

Lvberg *et al* used an arithmetic crossover operator to produce offspring from two randomly selected particles.

- Select particles with indices a and b to be used for crossover
- Replace of both particles, the positions with

$$\mathbf{x}_a(t+1) = \mathbf{r} \otimes \mathbf{x}_a(t) + (\mathbf{r} - 1) \otimes \mathbf{x}_b(t) \tag{19}$$

$$\mathbf{x}_b(t+1) = \mathbf{r} \otimes \mathbf{x}_b(t) + (\mathbf{r} - 1) \otimes \mathbf{x}_a(t) \tag{20}$$

- Replace of both particles, the velocities with

$$\mathbf{v}_a(t+1) = \frac{\mathbf{v}_a(t) + \mathbf{v}_b(t)}{\|\mathbf{v}_a(t) + \mathbf{v}_b(t)\|} \|\mathbf{v}_a(t)\| \tag{21}$$

$$\mathbf{v}_b(t+1) = \frac{\mathbf{v}_a(t) + \mathbf{v}_b(t)}{\|\mathbf{v}_a(t) + \mathbf{v}_b(t)\|} \|\mathbf{v}_b(t)\| \tag{22}$$

The personal best position of an offspring is initialized to its current position.

Reproduction in PSO

- There are numerous ways you can select whether two particles should be bread together when using Lvberg *et al* approach.
- The breeding process is done for each iteration after the velocity and position updates have been done.

Mutation

There are many ways of utilizing mutation in PSO. We will go over some of the early approaches:

- Miranda and Fonseca mutated only the global best position as follows,

$$\mathbf{g}(t+t) = \mathbf{g}^{'}(t+1) + \eta N(0, \mathbf{I}) \tag{23}$$

  η is referred to as a learning parameter, which can be a fixed value, or adapted as a strategy parameter as in evolutionary strategies.
- Higashi and Iba mutate the components of particle position vectors at a specified probability
  - For each dimension j if $U(0,1) < P$ (P a parameter)
  - update the component with

$$x_{i,j}(t+t) = x^{'}_{i,j}(t+1) + N(0, r)x^{'}_{i,j}(t+1) \tag{24}$$

    where $r = 0.1 * (s_j)$, and s_j is the diameter of the search space in dimension j.

Mutation

Secrest and Lamont proposed a substantially change to PSO by using the update equations

$$\mathbf{x}_i = \begin{cases} \mathbf{p}_i(t) + \mathbf{v}_i(t+1) & \text{if } U(0,1) > c_1 \\ \mathbf{g}(t) + \mathbf{v}_i(t+1) & \textit{else} \end{cases} \tag{25}$$

where $\mathbf{v}_i(t+1) = m(t+1)\mathcal{H}_i(\mathbf{0}, 1)$, and

$$m(t+1) = \begin{cases} N(0, (1-c_2)\|\mathbf{p}_i(t) - \mathbf{g}_i(t)\|) & \text{if } U(0,1) > c_2 \\ N(0, c_2\|\mathbf{p}_i(t) - \mathbf{g}_i(t)\|) & \text{else} \end{cases} \tag{26}$$

- Coefficient $c_1 \in (0, 1)$ specifies the trust between the global and personal best positions.
  - The larger c_1, the more particles are placed around the global best position;
- $c_2 \in (0, 1)$ establishes the point between the global best and the personal best to which the corresponding particle will converge

DE Inspired PSO

Differential evolution (DE), is a popular and effective evolutionary algorithm. Full details will be discussed in the DE lecture.

- For each particle i
  - Selected three particles, $a \neq b \neq c$.
  - replace PSO update equations with

$$x'_{i,j}(t+1) = \begin{cases} x_{a,j} + \beta(x_{b,j} - x_{c,j}) & \text{if } U(0,1) \leq P \\ x_{i,j} & else \end{cases} \tag{27}$$

    p the probability of crossover, and β the scaling factor.
  - set $x_{i,j}(t+1) = x'_{i,j}(t+1)$ only if $f(x'_{i,j}(t+1)) < f(x_{i,j}(t))$ (assuming minimization)

The DE step was used intermittently along with the normal PSO updated by Hendtlass

PSO Theory

Content from GECCO 2018 tutorial slides 49-72