**Swarm**

**Introduction – PSO Precursors**
In 1986 I made a computer model of coordinated animal motion such as bird flocks and fish schools. It was based on three dimensional computational geometry of the sort normally used in computer animation or computer aided design.
I called the generic simulated flocking creatures boids. The basic flocking model consists of three simple steering behaviors which describe how an individual boid maneuvers based on the positions and velocities its nearby flockmates. 


**Swarm**

**Introduction**
- A population-based stochastic optimization technique modelled on the social behaviors observed in animals or insects, e.g., bird flocking, fish schooling, and animal herding. Originally proposed by James Kennedy and Russell Eberhart in 1995.
- Initially they intended to model the emergent behavior (i.e., self-organization) of flocks of birds and schools of fish.
- The coordinated search for food lets a swarm of birds land at a certain place where food can be found.
- The behavior was modeled with simple rules for information sharing between the individuals of the swarm.
- Their model further evolved to handle optimization.
- The term particle was used simply because the notion of velocity was adopted — particle seemed to be the most appropriate term in this context.

**Swarm**

**Introduction**
- The movements of a particle depend only on:
  1. its velocity and
  2. the locations where good solutions have already been found by the particle itself or other (neighboring) particles in the swarm.
- This is in analogy to bird flocking where each individual makes its decisions based on:
  1. cognitive aspects (modeled by the influence of good solutions found by the particle itself) and
  2. social aspects (modeled by the influence of good solutions found by other particles).
- The swarm of particles uses no gradient information.

**Swarm**

**The main idea**
The particle’s move — two attractors:
- Each particle keeps track of the coordinates in the search space which are associated with the best solution it has found so far (the corresponding value of the objective function is also stored).
- Another “best” value that is tracked by each particle is the best value obtained so far by any particle in its topological neighborhood (when a particle takes the whole population as its neighbors, the best value is a global best).
- At each iteration the velocity of each particle is changed towards the above-mentioned two attractors: (1) personal and (2) global best (or neighborhood best) locations.
- Also some random component is incorporated into the velocity update.
Swarm

Particle Swarm
1: Initialize location and velocity of each particle \( x \in P_{\text{swarm}} \).
2: repeat
3: evaluate \( (P_{\text{swarm}}) \)
4: for all \( x_j \) from \( P_{\text{swarm}} \) do
5: update the personal best position
6: update the global best position \( \land \) depends on the neighborhood
7: end for
8: for all \( x_j \) from \( P_{\text{swarm}} \) do
9: update the velocity
10: compute the new location of the particle
11: end for
12: until termination condition met

Particle Swarm

Swarm

Velocity and location update in \( \mathbb{R}^n \):
\[
\begin{align*}
\mathbf{v}_t^{i+1} &= \mathbf{v}_t^i + \mathbf{a}_t^{i+1}, \\
\mathbf{x}_t^{i+1} &= \mathbf{x}_t^i + \mathbf{v}_t^{i+1}
\end{align*}
\]

Each coordinate is evaluated separately:

\[
d_j^{i+1} = \phi_1 \cdot r_1 (y_j^i - x_j^i) + \phi_2 \cdot r_2 (y_j^i - g_j^i),
\]

where:
- \( \mathbf{v}_t^i \) — particle’s velocity,
- \( x_j^i \) — particle’s location,
- \( a_j^i \) — particle’s acceleration,
- \( y_j^i \) — the best location the particle has found so far,
- \( g_j^i \) — the best location obtained so far by any particle in the neighborhood of \( a_j^i \).
- \( r_1, r_2 \) — random values: \( U(0, 1) \).

Swarm

Disadvantage of the approach from 1995
- It is necessary to clamp particle velocities in this original algorithm at a maximum value \( \text{vmax} \) for all \( t \):

\[
v_t^{i+1} = \begin{cases} 
v_t^{i+1} & \text{if } v_t^{i+1} < \text{vmax} \\ \text{vmax} & \text{otherwise} \end{cases}
\]

- Without this clamping in place the system was prone to entering a state of explosion, wherein the random weighting of the \( r_1 \) and \( r_2 \) values caused velocities and thus particle positions to increase rapidly, approaching infinity.

Swarm

The neighborhood
- A particle’s neighborhood is defined as the subset of particles which it is able to communicate with.
- The first PSO model used an Euclidian neighborhood for particle communication, measuring the actual distance between particles to determine which were close enough to be in communication.
- The Euclidian neighborhood model was abandoned in favor of less computationally intensive models when research focus was shifted from biological modeling to mathematical optimization.
- Topological neighborhoods unrelated to the locality of the particle came into use (including a global neighborhood, or gbest model, where each particle is able to obtain information from every other particle in the swarm).

Swarm

Topological neighborhoods
- Local topology — any swarm model without global communication.
- One of the simplest form of a local topology is the ring model. The lbest ring model connects each particle to only two other particles in the swarm.
- The lbest swarm model showed lower performance, that is, slower convergence rate relative to the gbest model.
- The much faster convergence of the gbest model seems to indicate that it produces superior performance, but this is misleading — risk of premature convergence.

Swarm

PSO and EC: Comparison
Simulation
- Both PSO and EC are population-based.
- Both PSO and EC use fitness concept.

Differences
- In PSO less-fit particles do not die (no “survival of the fittest” mechanism).
- In PSO there is no evolutionary operators like crossover or mutation but each particle is varied according to its past experience and relationship with other particles in the population (swarm).
Swarm

Disadvantage of the approach from 1995

- vmax method — viewed as both artificial and difficult to balance:
  1. very large spaces required larger values to ensure adequate exploration, while
  2. smaller spaces required very small values to prevent explosion-like behavior on their scale.
- a poorly-chosen vmax could result in extremely poor performance, yet there was no simple, reliable method for choosing this value beyond trial and error.

Inertia weight parameter

Obtaining convergent behaviour of a swarm was a real pain. Therefore...

- A few years after the initial PSO publications, a velocity equation with a new parameter was introduced — the inertia weight parameter w:

\[ v_{t+1}^{j} = w \cdot v_{t}^{j} + c_{1} \cdot r_{1}(y_{t}^{j} - x_{t}^{j}) + c_{2} \cdot r_{2}(y^{*t}_{t} - x_{t}^{j}) \]

where
- \( v_{t}^{j} \): previous velocity
- \( y_{t}^{j} \): cognitive component
- \( y^{*t}_{t} \): social component
- \( w \): inertia weight parameter
- \( c_{1} \) and \( c_{2} \): cognitive and social components
- \( r_{1} \) and \( r_{2} \): random numbers

Further development of vmax method — viewed as both artificial and difficult to balance:

- 1. dynamically decrease vmax when gbest does not improve over \( r \) iterations:
  \[
  v_{t+1}^{j} = \begin{cases} \beta \cdot v_{t}^{j} \text{ if } F(x') \geq F(x^{t-1}) \forall x' \in \{1, \ldots, r\} \\ v_{max}^{j} \text{ otherwise} \end{cases}
  \]
  where \( 0 < \beta < 1 \) and \( \beta \) is also decreased by 0.01.
- 2. exponentially decreasing vmax during the process of search:
  \[
  v_{t+1}^{j} = (1 - (t/n)^{p}) \cdot v_{max}^{j}
  \]

Velocity components

1. previous velocity: \( w \cdot v_{t}^{j} \)
- 1.1 inertia component
- 1.2 memory of previous flight direction
- 1.3 prevents particle from drastically changing direction
2. cognitive component: \( c_{1} \cdot r_{1}(y_{t}^{j} - x_{t}^{j}) \)
- 2.1 quantifies performance relative to past performances
- 2.2 memory of previous best position
- 2.3 nostalgia
3. social component: \( c_{2} \cdot r_{2}(y^{*t}_{t} - x_{t}^{j}) \)
- 3.1 quantifies performance relative to neighbors
- 3.2 envy

In 2000, Eberhart and Shi proposed values \( w = 0.7298 \) and \( c_{1} = c_{2} = 1.49618 \) as quite effective for a set of 1-dimensional optimization functions.
Swarm

Inertia weight parameter

Dynamically changing inertia weights:

- linear decreasing:
  \[ w(t + 1) = w(t) - \alpha \cdot w(t) \]
- non-linear decreasing:
  \[ w(t + 1) = w(t) - \alpha \cdot \sqrt{w(t)} \]
- based on relative improvement for \( i \)-th particle:
  \[ w(t + 1) = w(t) \cdot \frac{m_i(t)}{m_i(t) + 1} \]

The characteristic polynomial of a 3-dimensional system:

\[
\lambda^3 - (1 + \omega - \phi_1 - \phi_2)\lambda^2 + (1 + \omega - \phi_2 + 1)\lambda - \omega + \phi_1 - \phi_2 = 0
\]

From a system of equations:

\[
\begin{align*}
\dot{x}^i &= x^i - \phi_1 x^i + \phi_2 y^* - x^i, \\
\dot{y}^i &= y^i - \phi_1 x^i + \phi_2 y^* - y^i
\end{align*}
\]

A recursive formula for particle coordinates can be derived:

\[
x^i(t+1) = (1 - \omega - \phi_1 - \phi_2) x^i - \omega y^i + \phi_1 y + \phi_2 y^*
\]

Swarm

Theoretical analysis

The stability analysis for the model with the inertia weight parameter (\( w(t) \)):

\[
\begin{align*}
\dot{x}^i &= w^i \cdot x^i \lambda_1 - x^i, \\
\dot{y}^i &= w^i \cdot y^i \lambda_2 - y^i
\end{align*}
\]

is presented in [9].

From a system of equations:

\[
\begin{align*}
\dot{x}^i &= x^i - \phi_1 x^i + \phi_2 (y^* - x^i), \\
\dot{y}^i &= y^i - \phi_1 x^i + \phi_2 (y^* - y^i)
\end{align*}
\]

A recursive formula for particle coordinates can be derived:

\[
x^i(t+1) = (1 - \omega - \phi_1 - \phi_2) x^i - \omega y^i + \phi_1 y + \phi_2 y^*
\]

Swarm

When we know eigenvalues, we can switch from the recursive formula to the formula without recursion. For the proposed deterministic model a coordinate of the solution can be evaluated for any time \( t \):

\[
x^i(t) = k_1 y^i + k_2 y^{i-1} + k_3 y^{i-2}
\]

where:

\[
\begin{align*}
k_1 &= \frac{\alpha \cdot \phi_1 \cdot y^*}{\phi_2 \cdot (\phi_2 + \phi_1)} \\
k_2 &= \frac{\alpha \cdot \phi_1 \cdot (y^* - y)}{\phi_2 \cdot (\phi_2 + \phi_1)} \\
k_3 &= \frac{\alpha \cdot \phi_1 \cdot (y^* - y)}{\phi_2 \cdot (\phi_2 + \phi_1)}
\end{align*}
\]

for a given \( y \) and \( y^* \) remain unchanged.

Eq. (9) is valid as far as \( y \) and \( y^* \) remain unchanged.

If any better solution is found, \( y \) and \( y^* \) should be updated and \( k_1 \), \( k_2 \), \( k_3 \) should be recalculated.
In [6] authors prove that:

\[ x^t \] converges (more or less rapidly) to

\[ x = k = \frac{\phi_1 y + \phi_2 y^*}{\phi_1 + \phi_2} \]  

(11)
as long as the following condition is met:

\[ \max\{||\lambda_1||, ||\lambda_2||\} < 1. \]  

(12)

Another method of balancing global and local searches known as constriction was being explored simultaneously with the inertia weight method and was occasionally referenced in PSO literature, though the actual research proposing its use was not published until 2002.

The only problem is . . .

... how to tune the PSO control parameters \( w, \phi_1, \phi_2 \)?

Precisely: the number of possible configurations satisfying system of inequalities:

\[
\begin{cases}
    w > 0 \land w < 1, \\
    \phi_1 + \phi_2 > 0, \\
    w > 0,5(\phi_1 + \phi_2) - 1
\end{cases}
\]

(13)
is infinitely large. ☹️

In [8] authors prove that:

\[ x_t \] converges (more or less rapidly) to

\[
\lim_{t \to +\infty} x_t = k_1 = \phi_1 y + \phi_2 y^*.
\]

(11)
as long as the following condition is met:

\[
\max\{||\lambda_1||, ||\lambda_2||\} < 1. \]

(12)

Application of the convergence rules

1. Select a point in the region for which the particle strictly converges; \( \psi, \omega \).

2. Evaluate a new velocity of a particle with the formula, for example:

\[
\psi^{t+1} = \omega \psi_t + \psi^* (y - x^t) + \psi_\text{conv} \cdot (1 - r)(\psi_t - \psi^t)
\]

(14)

instead of:

\[
\psi^{t+1} = \omega \psi_t + \gamma \psi^* (y - x^t) + \omega \psi_\text{conv} \cdot (y - x^t)
\]

(15)

But it is still not clear . . .

- which point \( \psi_\text{conv} \) and \( \omega_\text{conv} \) should be selected?
- do we have to keep this point through the entire search process?
- do all the particles in the swarm should have the same values of \( \psi_\text{conv} \) and \( \omega_\text{conv} \)?
- . . .

General representation

In [3] a more general representation is produced by adding five coefficients \( \alpha, \beta, \gamma, \delta, \eta \):

\[
\begin{cases}
    \psi^{t+1} = \alpha \psi_t + \beta \psi^* (y - x^t), \\
    x^{t+1} = \psi^{t+1} + \gamma \psi^* (y - x^t)
\end{cases}
\]

(16)

Version from [6] is obtained for \( \alpha = 1, \beta = 1, \gamma = 1, \delta = 1, \eta = 1 \).

Step back to classic equations (where \( z = y - x^t \)) looks like here:

\[
\begin{cases}
    \psi^{t+1} = \phi^t + \lambda \psi^* (y - x^t), \\
    x^{t+1} = \psi^{t+1} + \lambda \psi^* (y - x^t)
\end{cases}
\]

(17)
It was found that when $\phi < 4$, the swarm would slowly “spiral” toward and around the best found solution in the search space with no guarantee of convergence, while for $\phi > 4$ and $\kappa \in [0, 1]$ convergence would be quick and guaranteed.

Constriction was being explored simultaneously with the inertia weight method and was occasionally referenced in PSO literature, though the actual research proposing its use was not published until 2002.

The last model made a successful career.

**Observation:**

It was found that when $\phi < 4$, the swarm would slowly “spiral” toward and around the best found solution in the search space with no guarantee of convergence, while for $\phi > 4$ and $\kappa \in [0, 1]$ convergence would be quick and guaranteed.

Constriction was being explored simultaneously with the inertia weight method and was occasionally referenced in PSO literature, though the actual research proposing its use was not published until 2002.

The last model made a successful career.

The parameter values noted above are preferred in most cases when using constriction for modern PSOs due to the proof of stability.
Swarm

Synchronous vs asynchronous updates
- **synchronous** — personal best and neighborhood bests updated separately from position and velocity vectors
  1. slower feedback
  2. better for \( gbest \)
- **asynchronous** — new best positions updated after each particle position update
  1. immediate feedback about best regions of the search space
  2. better for \( lbest \)

Swarm

Communication topologies are expressed in the velocity update procedure:
- \( gbest \) — each particle is influenced by the best found from the entire swarm.
- \( lbest \) — each particle is influenced only by particles in local neighbourhood.

Swarm

Adaptive acceleration coefficients \( c_1 \) and \( c_2 \)
1. \( c_1 = c_2 = 0 \) — swarm is one stochastic hill-climber.
2. \( c_1 > 0 \) \( c_2 = 0 \) — particles are independent hill climbers performing own local search processes.
3. \( c_1 = c_2 > 0 \) — swarm is one stochastic hill-climber.
4. \( c_1 = c_2 = 0 \) — particles are attracted towards the average of \( y^* \) and \( y \).
5. \( c_1 > c_2 \) — more beneficial for unimodal problems,
6. \( c_1 < c_2 \) — more beneficial for multimodal problems,
7. low \( c_1 \) and \( c_2 \) — smooth particle trajectories,
8. high \( c_1 \) and \( c_2 \) — more acceleration, abrupt movements.

Swarm

Bare Bones PSO
In [5] authors propose a PSO variant, which drops the velocity term from the PSO equation and introduces a Gaussian sampling, based on the swarm best (\( gbest \) or \( lbest \)) and personal best (\( pbest \)) information.

- Motivation:
  1. The observed distribution of new location samples for a particle is a bell curve centered midway between \( y^* \) and \( y^* \) and extending symmetrically beyond them.
  2. So, we should simply generate normally distributed random numbers around the mean \( (y^* + y^*)/2 \).

- In BBPSO the canonical update equations are replaced by:

\[
\begin{align*}
\delta^i \left( t \right) &= \mu \delta^i \left( t \right) \\
\delta^i \left( t \right) &= \sigma \delta^i \left( t \right)
\end{align*}
\]

In experimental research the canonical version performed competitively but not outstandingly [5].

Swarm

Communication topologies

Communication topologies are expressed in the velocity update procedure:
- \( gbest \) — each particle is influenced by the best found from the entire swarm.
- \( lbest \) — each particle is influenced only by particles in local neighbourhood.
Communication topologies

**Balance between exploration and exploitation**

- *gbest* model propagate information the fastest in the population; while the *lbest* model using a ring structure the slowest.
- For complex multimodal functions, propagating information the fastest might not be desirable.
- However, if this is too slow, then it might incur higher computational cost.
- Mendes and Kennedy (2002) found that von Neumann topology seems to be an overall winner among many different communication topologies.

For complex multimodal functions, propagating information the fastest might not be desirable.

- However, if this is too slow, then it might incur higher computational cost.
- Mendes and Kennedy (2002) found that von Neumann topology seems to be an overall winner among many different communication topologies.

The adaptive random topology

- At the very beginning, and after each unsuccessful iteration (no improvement of the best known fitness value), the graph of the information links is modified.
- Each particle informs at random K particles (the same particle may be chosen several times), and informs itself.
- The parameter \( K \) is usually set to 3:
  - each particle informs at least one particle (itself), and at most \( K+1 \) particles (including itself)
  - each particle can be informed by any number of particles between 1 and \( |S| \).
- On average, a particle is often informed by about \( K \) others but the distribution of the possible number of informants is not uniform.

References I