Particle Swarm Optimization

Krzysztof Trojanowski

IPI PAN & UKSW, Warszawa

30 maja 2015 r.

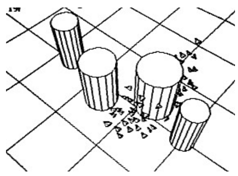
Swarm

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.

Boids, Background and Update by Craig Reynolds

http://www.red3d.com/cwr/boids/



Reynolds, C. W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pages 25-34

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 behaviour 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

- A population of particles (the *swarm*) — each particle represents a location in a multidimensional search space.
- The particles start at random locations and with random velocity.
- The particles search for the minimum (or maximum) of a given objective function by moving through the search space.
- The analogy to reality (in the case of search for a maximum) is: the objective function measures the quality or amount of the food at each place and the particle swarm searches for the place with the best or most food.

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

Initialize location and velocity of each particle $\mathbf{x} \in P_{\text{swarm}}$.
repeat
  evaluate (P_{swarm})
  for all $\mathbf{x}_j$ from P_{swarm} **do**
    update the personal best position
    update the global best position ▷ depends on the neighborhood
  end for
  for all $\mathbf{x}_j$ from P_{swarm} **do**
    update the velocity
    compute the new location of the particle
  end for
until termination condition met

Swarm

Velocity and location update in $\mathbf{R}^n$:

$$\begin{aligned} \mathbf{v}^{t+1} &= \mathbf{v}^t + \mathbf{a}^{t+1} , \\ \mathbf{x}^{t+1} &= \mathbf{x}^t + \mathbf{v}^{t+1} \end{aligned}$$

Each coordinate is evaluated separately:

$$a_j^{t+1} = \varphi_1 \cdot r_1^t (y_j^t - x_j^t) + \varphi_2 \cdot r_2^t (y_j^{*t} - x_j^t) ,$$

[Kennedy and Eberhart, 1995]

where:
$\mathbf{v}^t$ — particle's velocity,
$\mathbf{x}^t$ — particle's location,
a_j^t — particle's acceleration,
y_j^t — the best location the particle $\mathbf{x}^t$ has found so far,
y_j^{*t} — the best location obtained so far by any particle in the neighborhood of $\mathbf{x}^t$.
r_1, r_2 — random values: $U(0, 1)$.

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

Similarities

- 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

- It is necessary to clamp particle velocities in this original algorithm at a maximum value *vmax*:

$$v_j^{t+1} = \begin{cases} v_j^{t+1} & \text{if } v_j^{t+1} < vmax_j \\ vmax_j & \text{otherwise} \end{cases}$$

- 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

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.

Swarm

Disadvantage of the approach from 1995

- The *vmax* parameter drawbacks:
  1. *vmax* is problem dependent,
  2. does not controls the positions, only the step sizes.
- Further development of *vmax* mechanism:
  1. dynamically decrease *vmax* when gbest does not improve over τ iterations:

$$vmax_j^{t+1} = \begin{cases} \beta \cdot vmax_j^t & \text{if } F(\hat{\mathbf{x}}^t) \geq F(\hat{\mathbf{x}}^{t-t'}) \ \forall t' \in \{1, \ldots, \tau\} \\ vmax_j^t & \text{otherwise} \end{cases}$$

  where $0 < \beta < 1$ and β is also decreased by 0.01.
  2. exponentially decreasing *vmax* during the process of search:

$$vmax_j^{t+1} = (1 - (t/n_t)^{\alpha}) vmax_j^t$$

Swarm

Convergence analysis

In [Clerc and Kennedy, 2002] authors presented **convergence analysis for the approach from 1995**. This shed some light on the problem of parameters tuning for the convergent behaviour of a swarm.

- Essential properties:
  1. **stability of particles** — convergence of particles to a point in the search space
  2. **local convergence property** — the PSO algorithm converges to a local optimum
- Aim of analysis:
  define boundaries for the parameters of PSO in such a way that if the parameters are selected in these boundaries, the particles are stable.

Swarm

Convergence analysis

In [Clerc and Kennedy, 2002] authors assumed that:

1. the particle moves in one-dimensional search space,
2. the rules of the particle's movement are deterministic, that is, random vales in the formula are replaced by their expected values (equal 0.5)
3. both the attractors remain in the same place of the search space,
4. we have just one particle to observe (due to the fact that global attractor remains unchanged, there is no any other communication between particles).

Thus, all the further equations consider a value of x instead of a vector $\mathbf{x}$.

Swarm

Convergence analysis – the stable point

The particle reaches equilibrium point when velocity equals zero:

$$\varphi_1(y - x) + \varphi_2(y^* - x) = 0 \tag{1}$$

that is:

$$\varphi_1 y + \varphi_2 y^* = \varphi_1 x + \varphi_2 x. \tag{2}$$

This particular location x where there is no velocity equals:

$$x = \frac{\varphi_1 y + \varphi_2 y^*}{\varphi_1 + \varphi_2} \tag{3}$$

Swarm

Convergence analysis – the stable point

Assuming that equilibrium point is a local attractor:

$$y \leftarrow \frac{\varphi_1 y + \varphi_2 y^*}{\varphi_1 + \varphi_2}. \tag{4}$$

Let's substitute x by y in Eq. (2). This gives:

$$y\varphi_1 + y\varphi_2 = \varphi_1 y + \varphi_2 y^* \ \Rightarrow \ y = y^* \tag{5}$$

that is, the equilibrium state is truly obtained when the local attractor is also a global attractor.

Swarm

Convergence analysis

Reformulation of the velocity equation:
Let's redefine $\varphi = \varphi_1 + \varphi_2$ and $y = \frac{\varphi_1 y + \varphi_2 y^*}{\varphi_1 + \varphi_2}$.
This gives:

$$v^{t+1} = v^t + \varphi(y - x^t), \quad (6)$$
$$x^{t+1} = x^t + v^{t+1}, \quad (7)$$

where y i φ are constant for any t.

Swarm

Convergence analysis

Let z^t represents difference between the current location of a particle and optimum: $z^t = y - x^t$

$$\begin{cases} v^{t+1} = v^t + \varphi z^t, \\ z^{t+1} = -v^t + (1 - \varphi) z^t. \end{cases} \quad (8)$$

This way **a basic simplified dynamic system** can be defined:

$$\mathbf{P}_{t+1} = M \times \mathbf{P}_t, \quad (9)$$

where:

$$M = \begin{bmatrix} 1 & \varphi \\ -1 & 1-\varphi \end{bmatrix}_{2\times 2} \quad \mathbf{P}^t = \begin{bmatrix} v^t \\ z^t \end{bmatrix}_{2\times 1}$$

Swarm

Convergence analysis

In the context of the dynamic system theory:

- $\mathbf{P}^t$ — the particle state made up of its current position and velocity,
- M — the dynamic matrix whose properties determine the time behavior of the particle (asymptotic or cyclic behavior, convergence, etc.),

In general, the initial particle state is not at equilibrium.

It is of highest practical importance to determine:

- whether the particle will eventually settle at the equilibrium (that is if the optimization algorithm will converge) and
- how the particle will move in the state space (that is how the particle will sample the state space in search of better points).

Standard results from dynamic system theory say that the time behavior of the particle depends on the eigenvalues of the dynamic matrix.

Swarm

Convergence analysis

Eigen values of M are the solutions of characteristic polynomial, that is, roots of the determinant $\det(\lambda I - M)$:

$$\det\left(\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 1 & \varphi \\ -1 & 1-\varphi \end{bmatrix}\right) =$$
$$\det\left(\begin{bmatrix} \lambda - 1 & -\varphi \\ 1 & \lambda - 1 + \varphi \end{bmatrix}\right) = \lambda^2 + (\varphi - 1)\lambda + 1$$

Thus:

$$\begin{cases} \lambda_1 = 1 - \frac{\varphi}{2} + \frac{\sqrt{\varphi^2 - 4\varphi}}{2}, \\ \lambda_2 = 1 - \frac{\varphi}{2} - \frac{\sqrt{\varphi^2 - 4\varphi}}{2}. \end{cases} \quad (10)$$

Swarm

Convergence analysis

$$\lambda = \overbrace{\underbrace{1 - \frac{\varphi}{2}}_{\text{real}} \pm \underbrace{\frac{\sqrt{\varphi^2 - 4\varphi}}{2}}_{\text{imaginary or real}}}^{\text{complex number}}$$

Assuming that:
$\varphi_1 > 0$, $\varphi_2 > 0$ and $\varphi = \varphi_1 + \varphi_2$,

one can discuss just three cases:

1. $0 < \varphi < 4$ (the solution is a complex number),
2. $\varphi > 4$ (the solution is a real value),
3. $\varphi = 4$ (the special case).

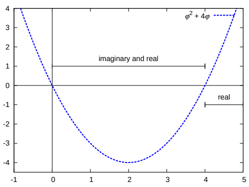


Figure: φ intervals for λ_1 and λ_2 being a real or a complex number

Swarm

Convergence analysis

The particle location in k-th step of the algorithm can be obtained from:

$$\mathbf{P}_k = M^k \times \mathbf{P}_0 \quad (11)$$

Thus, in searching for convergent behaviour of a particle we need to find φ i k such that:

$$M^k = I. \quad (12)$$

Swarm

Convergence analysis

$det\left(\begin{bmatrix} 1 & \varphi \\ -1 & 1-\varphi \end{bmatrix}\right) > 0$ (equal to 1, in fact), so it exist P so that:

$$P^{-1}MP = \Lambda \tag{13}$$

where:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{14}$$

Therefore, eventually we have to solve $\Lambda^k = I$:

$$\begin{bmatrix} \lambda_1^k & 0 \\ 0 & \lambda_2^k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ that is, we must have } \lambda_1^k = \lambda_2^k = 1. \tag{15}$$

Swarm

Convergence analysis

Let's remind that we have:

$$\begin{cases} \lambda_1 &= 1 - \frac{\varphi}{2} + \sqrt{\Delta}, \\ \lambda_2 &= 1 - \frac{\varphi}{2} - \sqrt{\Delta} \end{cases} \quad \text{where} \quad \Delta = \frac{\varphi^2 - 4\varphi}{2} = \left(1 - \frac{\varphi}{2}\right)^2 - 1 \tag{16}$$

Thus, $\lambda_1^k = \lambda_2^k = 1$ can be found $\iff$

the solutions of characteristic polynomial are complex numbers, that is, $\frac{\varphi^2 - 4\varphi}{2} < 0$, which means that $\varphi < 4$ must be satisfied.

Swarm

Convergence analysis

Solutions of $\lambda_1^k = \lambda_2^k = 1$ and $\varphi < 4$:

$$(k, \varphi) = (3, 3), (4, 2), (5, \frac{3-\sqrt{5}}{2}), (5, \frac{3+\sqrt{5}}{2}), (6, 1) \tag{17}$$

In these cases, after a number of steps the particle goes back to its starting position.

How to show this?

Print figures with subsequent positions of a particle in 2-dimensional space *speed vs. distance to attractor*, that is, $v \times z$ for $z^t = y - x^t$ and:

$$\begin{cases} v^{t+1} &= v^t + \varphi z^t, \\ z^{t+1} &= -v^t + (1-\varphi) z^t. \end{cases} \tag{18}$$

Swarm

Convergence analysis

The deterministic model of a particle movement

```
Initialize location, attractor and velocity of a particle, for example, x = 1; y = 1, v = 1.
Initialize φ                         ▷ for example, φ ∈ 3, 2, (3−√5)/2, (3+√5)/2, 1
z = y − x                            ▷ update the reference variable z
repeat
    v = v + φz                       ▷ update the speed
    x = x + v                        ▷ update the location
    z = y − x                        ▷ update z
    cout << "v:  " << v << "z:  " << z << "x:  " << x << endl;
until termination condition met
```


Swarm

Convergence analysis

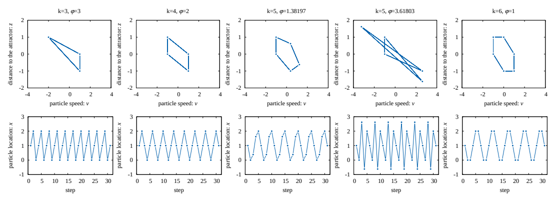


Figure: Cyclic movement of a particle for different values of $(k, \varphi) = (3, 3), (4, 2), (5, \frac{3-\sqrt{5}}{2}), (5, \frac{3+\sqrt{5}}{2}), (6, 1)$.

Swarm

Convergence analysis

For other values of φ but satisfying also $\varphi < 4$:

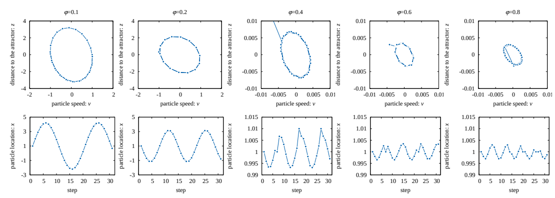


Figure: Quasicyclic movement of a particle for different values of φ.

Swarm

Convergence analysis

For $\varphi > 4$ the values of λ_1 and λ_2 are real:

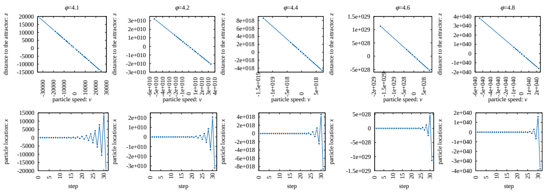


Figure: Non-cyclic movement of a particle for different values of φ.

There is no chance for even quasicyclic behaviour..

Swarm

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:

$$v_j^{t+1} \quad = \quad w \cdot v_j^t + c_1 r_1^t (y_j^t - x_j^t) + c_2 r_2^t (y_j^{*t} - x_j^t) \ ,$$

[Shi and Eberhart, 1998]

Swarm

Inertia weight parameter

- w — designed to replace *vmax* by adjusting the influence of the previous particle velocities on the optimization process.
- By adjusting the value of w, $w > 0$, the swarm has a greater tendency to eventually constrict itself down to the area containing the best fitness and explore that area in detail.

Swarm

Velocity components

$$v^{t+1} \quad = \quad w \cdot v_j^t + c_1 \cdot r_1^t (y_j^t - x_j^t) + c_2 \cdot r_2^t (y_j^{*t} - x_j^t) \ ,$$

1. previous velocity: $w \cdot v_j^t$
   1.1 inertia component
   1.2 memory of previous flight direction
   1.3 prevents particle from drastically changing direction
2. cognitive component: $c_1 \cdot r_1^t (y_j^t - x_j^t)$
   2.1 quantifies performance relative to past performances
   2.2 memory of previous best position
   2.3 nostalgia
3. social component: $c_2 \cdot r_2^t (y_j^{*t} - x_j^t)$
   3.1 quantifies performance relative to neighbors
   3.2 envy

Swarm

Inertia weight parameter

- For $w \geq 1$
  1. velocities increase over time
  2. swarm diverges
  3. particles fail to change direction towards more promising regions
- For $0 < w < 1$
  1. particles decelerate
  2. convergence also dependent on values c_1 and c_2
- The authors suggested using w as a dynamic value over the optimization process:
  1. starting with a value greater than 1.0 to encourage exploration, and
  2. decreasing eventually to a value less than 1.0 to focus the efforts of the swarm on the best area found in the exploration.

Swarm

Inertia weight parameter

Dynamically changing inertia weights:

- $w \sim N(0.72, \sigma)$
- linear decreasing:

$$w(t+1) = (w(0) - w(n_t)) \cdot \frac{n_t - t}{n_t} + w(n_t)$$

- non-linear decreasing:

$$w(t+1) = \alpha \cdot w(t)$$

  where $\alpha = 0.975$, $w(0) = 1.4$ and $w(n_t) = 0.35$.
- based on relative improvement for i-th particle:

$$w_i(t+1) = w(0) + (w(n_t) - w(0)) \cdot \frac{e^{m_i+1} - 1}{e^{m_i+1} + 1}$$

  where the relative improvement m_i is estimated as

$$m_i(t) = \frac{F(\mathbf{y}^{*t}) - F(\mathbf{x}_i^t)}{F(\mathbf{y}^{*t}) + F(\mathbf{x}_i^t)} \quad \text{where } \mathbf{y}^{*t} \text{ is the global attractor}$$

Swarm

Convergence analysis

The convergence analysis for the model with the inertia weight parameter ([Shi and Eberhart, 1998]):

$$
\begin{aligned}
v_j^{t+1} &= w \cdot v_j^t + c_1 \cdot r_1^t (y_j^t - x_j^t) + c_2 \cdot r_2^t (y_j^{*t} - x_j^t)\,, \qquad (19)\\
x_j^{t+1} &= x_j^t + v_j^{t+1} \qquad (20)
\end{aligned}
$$

is presented in [van den Bergh and Engelbrecht, 2006].

From a system of equations:

$$
\begin{aligned}
v^{t+1} &= w \cdot v^t + \varphi_1 (y^t - x^t) + \varphi_2 (y^{*t} - x^t), \qquad (21)\\
x^{t+1} &= x^t + v^{t+1} \qquad (22)
\end{aligned}
$$

a recursive formula for particle coordinates can be derived:

$$x^{t+1} = (1 - w - \varphi_1 - \varphi_2) x^t - w x^{t-1} + \varphi_1 y + \varphi_2 y^* \qquad (23)$$

Swarm

A model of a particle

In [van den Bergh and Engelbrecht, 2006] authors also assumed that:

1. the particle moves in one-dimensional search space,
2. the rules of the particle's movement are deterministic, that is, random vales in the formula are replaced by their expected values (equal 0.5)
3. both the attractors remain in the same place of the search space,
4. we have just one particle to observe (due to the fact that global attractor remains unchanged, there is no any other communication between particles).

Thus, all the further equations consider a value of x instead of a vector $\mathbf{x}$.

Swarm

The formula

$$x^{t+1} = (1 - w - \varphi_1 - \varphi_2) x^t - w x^{t-1} + \varphi_1 y + \varphi_2 y^* \qquad (24)$$

can be expressed as a product:

$$
\begin{bmatrix} x^{t+1} \\ x^t \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + w - \varphi_1 - \varphi_2 & -w & \varphi_1 y + \varphi_2 y^* \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^t \\ x^{t-1} \\ 1 \end{bmatrix}
$$

The characteristic polynomial of a 3×3 matrix is:

$$(1 - \lambda)(w - \lambda(1 + w - \varphi_1 - \varphi_2) + \lambda^2). \qquad (25)$$

which has a trivial root of $\lambda = 1$ and two other solutions:

$$
\left\{ \begin{array}{lcl} \lambda_1 & = & \frac{1+w-\varphi_1-\varphi_2+\Delta}{2}, \\ \lambda_2 & = & \frac{1+w-\varphi_1-\varphi_2+\Delta}{2}. \end{array} \right. \quad \text{where: } \Delta = \sqrt{(1 + w - \varphi_1 - \varphi_2)^2 - 4w}. \qquad (26)
$$

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:

$$x^t = k_1 + k_2 \lambda_1^t + k_3 \lambda_2^t \qquad (27)$$

where:

$$
\begin{aligned}
k_1 &= \frac{\varphi_1 y + \varphi_2 y^*}{\varphi_1 + \varphi_2}\\
k_2 &= \frac{\lambda_2 (x_0 - x_1) - x_1 + x_2}{\Delta(\lambda_1 - 1)} \qquad (28)\\
k_3 &= \frac{\lambda_1 (x_1 - x_0) + x_1 - x_2}{\Delta(\lambda_2 - 1)}
\end{aligned}
$$

for a given x_0, x_1 and $x_2 = (1 + w - \varphi_1 - \varphi_2) x_1 - w x_0 + \varphi_1 y + \varphi_2 y^*$.

Eq. (27) is valid as far as y i y^* remain unchanged.
If any better solution is found, y i y^* should be updated and k_1, k_2 i k_3 should be recalculated.

Swarm

In [van den Bergh and Engelbrecht, 2006] authors prove that:

x^t converges (more or less rapidly) to

$$\lim_{t \to +\infty} x^t = k_1 = \frac{\varphi_1 y + \varphi_2 y^*}{\varphi_1 + \varphi_2}, \qquad (29)$$

as long as the following condition is met:

$$\max\{||\lambda_1||, ||\lambda_2||\} < 1. \qquad (30)$$

Swarm

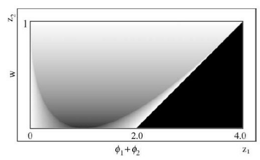


Fig. 2. The black triangle to the bottom right represents the values for which the particle strictly diverges, i.e. $\max(\|\lambda_1\|, \|\lambda_2\|) > 1$. This is the region for which $w < 0.5(\phi_1 + \phi_2) - 1$. The lighter regions represent the magnitude of $\max(\|\lambda_1\|, \|\lambda_2\|)$, with white representing magnitude 1. The darker regions (outside of the divergent region) represent values leading to more rapid convergence.

Figure: An experimentally obtained map.
Source: [van den Bergh and Engelbrecht, 2006].

The intensity of each point on the grid represents the magnitude $\max\{||\lambda_1||, ||\lambda_2||\}$, with lighter shades representing larger magnitudes.

Swarm

The only problem is . . .

. . . how to tune the PSO control parameters w, φ_1 i φ_2?

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

$$\begin{cases} w > 0 \wedge w < 1, \\ \varphi_1 + \varphi_2 > 0, \\ w > 0.5(\varphi_1 + \varphi_2) - 1 \end{cases} \tag{31}$$

is infinitely large. ☹

Swarm

Application of the convergence rules

1. Select a point in the region for which the particle strictly converges; φ_{conv} and w_{conv}.
2. Evaluate a new velocity of a particle with the formula, for example:

$$v_j^{t+1} = w_{\text{conv}} \cdot v_j^t + \varphi_{\text{conv}} \cdot r^t (y_j^t - x_j^t) + \varphi_{\text{conv}} \cdot (1 - r^t)(y_j^{*t} - x_j^t) \tag{32}$$

instead of:

$$v_j^{t+1} = w \cdot v_j^t + c_1 \cdot r_1^t (y_j^t - x_j^t) + c_2 \cdot r_2^t (y_j^{*t} - x_j^t) \tag{33}$$

But it is still not clear . . .

- which point φ_{conv} and w_{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 ϕ_{conv} and w_{conv}?
- . . .

Swarm

1 JAN

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.

D. Bratton, J. Kennedy, Defining a Standard for Particle Swarm Optimization, 2007 IEEE Swarm Intelligence Symposium

Swarm

General representation

In [Clerc and Kennedy, 2002] a more general representation is produced by adding five coefficients $\alpha, \beta, \gamma, \delta, \eta$:

$$\begin{cases} v^{t+1} &= \alpha v^t + \beta \varphi z^t, \\ z^{t+1} &= -\gamma v^t + (\delta - \eta \varphi) z^t. \end{cases} \tag{34}$$

Version from [Kennedy and Eberhart, 1995] is obtained for $\alpha = 1, \beta = 1, \gamma = 1, \delta = 1$ i $\eta = 1$.

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

$$\begin{cases} v^{t+1} &= \alpha v^t + \beta \varphi (y - x^t), \\ x^{t+1} &= y + \gamma v^t - (\delta - \eta \varphi)(y - x^t). \end{cases} \tag{35}$$

Swarm

General particle swarm algorithm

Assign κ and $\varphi_{\max}$
Calculate $\chi, \alpha, \beta, \gamma, \delta, \eta$
Initialize population, i.e., locations and velocities of particles, for example, random: $\mathbf{x}_i$, $\mathbf{v}_i$, and $\mathbf{p}_i = \mathbf{x}_i$.
repeat
  for $i = 1$ to *popsize* **do**
    if $F(\mathbf{x}_i) < F(\mathbf{p}_i)$ **then**
      $\mathbf{p}_i = \mathbf{x}_i$ ▷ update the particle attractor
    end if
  end for
  for $i = 1$ to *popsize* **do**
    $\mathbf{p}^* = \forall_{\mathbf{x} \in \{\mathcal{N}(\mathbf{x}_i) \cup \mathbf{x}_i\}} \arg\min F(\mathbf{x})$ ▷ update the neighborhood attractor
    for $d = 1$ to *dimensions* **do**
      $\varphi_1 = U(0,1) \cdot \varphi_{\max}/2$
      $\varphi_2 = U(0,1) \cdot \varphi_{\max}/2$
      $\varphi = \varphi_1 + \varphi_2$
      $y = ((\varphi_1 p_{id}) + (\varphi_2 p_d^*))/\varphi$
      $v_{id} = \alpha v_{id} + \beta \varphi (y - x_{id})$ ▷ update the speed based on the old v_{id}
      $x_{id} = y + \gamma v_{id} - (\delta - \eta \varphi)(y - x_{id})$ ▷ update the location based on x_{id} and the updated v_{id}
    end for
  end for
until termination condition met

Swarm

Particular classes of Swarm

Proposed in [Clerc and Kennedy, 2002]:

1. Model Type 1:

$$\begin{cases} v^{t+1} &= \chi (v^t + \varphi z^t), \\ z^{t+1} &= -\chi (v^t + (1 - \varphi) z^t). \end{cases} \tag{36}$$

2. Model Type 1':

$$\begin{cases} v^{t+1} &= \chi (v^t + \varphi z^t), \\ z^{t+1} &= -v^t + (1 - \varphi) z^t. \end{cases} \tag{37}$$

3. Model Type 1'':

$$\begin{cases} v^{t+1} &= \chi (v^t + \varphi z^t), \\ z^{t+1} &= -\chi v^t + (1 - \chi \varphi) z^t. \end{cases} \tag{38}$$

The last model made a successful career.

Swarm

Model Type 1"

- χ is derived from the existing constants in the velocity update equation:

$$\chi = \frac{2 \cdot \kappa}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad \text{where } \varphi = c_1 + c_2 \text{ and } \varphi > 4$$

- The factor κ controls balance between exploration and exploitation:
  1. $\kappa \approx 0$: fast convergence, local exploitation,
  2. $\kappa \approx 1$: slow convergence, high degree of exploration.

Swarm

Observation:
It was found that when $\varphi < 4$, the swarm would slowly "spiral" toward and around the best found solution in the search space with no guarantee of convergence, while for $\varphi > 4$ 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.

Swarm

Velocity update in $\mathbf{R}^n$:

$$v_j^{t+1} \quad = \quad \chi[v_j^t + c \cdot r_1^t \cdot (y_j^t - x_j^t) + c \cdot r_2^t \cdot (y_j^{*t} - x_j^t)] \ ,$$

[Kennedy & Clerc, 2002]

r_1^t i r_2^t: uniform random values in $\langle 0, 1 \rangle$.
Using the constant $\varphi = 4.1$ to ensure convergence, the values $c = 2.05$ $\chi = 0.729843788$ are obtained.

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

Swarm

Particle swarm algorithm Type 1"

Assign κ and $\varphi_{\max}$
Initialize population, i.e., locations and velocities of particles, for example, random: $\mathbf{x}_i$, $\mathbf{v}_i$, and $\mathbf{p}_i = \mathbf{x}_i$.
repeat
  for $i = 1$ to *popsize* **do**
    if $F(\mathbf{x}_i) < F(\mathbf{p}_i)$ **then**
      $\mathbf{p}_i = \mathbf{x}_i$ ▷ update the particle attractor
    end if
  end for
  for $i = 1$ to *popsize* **do**
    $\mathbf{p}^* = \forall_{\mathbf{x} \in \{\mathcal{N}(\mathbf{x}_i) \cup \mathbf{x}_i\}} \arg\min F(\mathbf{x})$ ▷ update the neighborhood attractor
    for $d = 1$ to *dimensions* **do**
      $\varphi_1 = U(0,1) \cdot \varphi_{\max,1}/2$
      $\varphi_2 = U(0,1) \cdot \varphi_{\max,2}/2$
      $v_{id} = \chi(v_{id} + \varphi_1(p_{id} - x_{id}) + \varphi_2(p_d^* - x_{id}))$ ▷ update the speed
      $x_{id} = x_{id} + v_{id}$ ▷ update the location based on x_{id} and the updated v_{id}
    end for
  end for
until termination condition met

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

Acceleration coefficients c_1 and c_2

1. $c_1 = c_2 = 0$ ...? ☺
2. $c_1 > 0$ $c_2 = 0$ — particles are independent hill climbers performing own local search processes,
3. $c_1 = 0$ $c_2 > 0$ — swarm is one stochastic hill-climber,
4. $c_1 = c_2 > 0$ — particles are attracted towards the average of $\mathbf{y}*$ and $\mathbf{y}$,
5. $c_2 > c_1$ — 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

Adaptive acceleration coefficients c_1 and c_2

$$c_1(t) = (c_{1,\min} - c_{1,\max}) \cdot \frac{t}{n_t} + c_{1,\max} \,,$$
$$c_2(t) = (c_{2,\min} - c_{2,\max}) \cdot \frac{t}{n_t} + c_{2,\max} \,.$$

An improved optimum solution for most of the benchmarks was observed when changing c_1 from 2.5 to 0.5 and changing c_2 from 0.5 to 2.5, over the full range of the search.

[A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients, IEEE TEVC, 2004]

Swarm

Bare Bones PSO

- In [Kennedy, 2003] 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^t and y^{*t} and extending symmetrically beyond them.
  2. So, we should simply generate normally distributed random numbers around the mean $(y^t + y^{*t})/2$.
- In BBPSO the canonical update equations are replaced by:

$$x_i^{t+1} = N(\mu^t, \sigma^t) \text{ where: } \quad \mu^t = (y^t + y^{*t})/2 \quad \text{and} \quad \sigma^t = |y^{*t} - y^t| \qquad (39)$$

In experimental research the canonical version performed competitively but not outstandingly [Kennedy, 2003].

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.

Swarm

Communication topologies

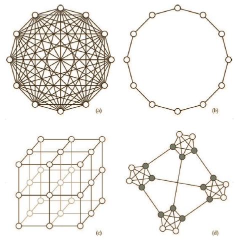


Figure: (a) star topology used in *gbest*, Ring topology used in *lbest*, (c) Von Neumann topology, and (d) Four clusters topology (aka "small world graph")

Swarm

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.

Swarm

Communication topologies

The adaptive random topology [Clerc, 2006]

- 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 less 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.

Swarm

Communication Topologies — FIPS: Fully Informed PSO

In [Mendes et al., 2004] the form of the particle location and velocity formula given in Model 1''([Clerc and Kennedy, 2002]):

$$\begin{cases} v^{t+1} &= \chi(v^t + \varphi(p - x^t)), \\ x^{t+1} &= x^t + v^t. \end{cases} \tag{40}$$

where $\varphi = \varphi_1 + \varphi_2$ and $p = \frac{\varphi_1 y + \varphi_2 y^*}{\varphi_1 + \varphi_2}$

uses an alternate form of calculating φ and p: $\varphi = \sum_{k \in \mathcal{N}} \frac{\varphi}{|\mathcal{N}|}$ and $p = \frac{\sum_{k \in \mathcal{N}} \mathcal{W}(k)\varphi y}{\sum_{k \in \mathcal{N}} \mathcal{W}(k)\varphi}$ where $\mathcal{N}$ is the neighborhood of the evaluated particle and the function $\mathcal{W}(k)$ may describe any aspect of the particle that is hypothesized to be relevant:

- the fitness of the best position found by the particle,
- the distance from that particle to the current individual,
- have return a constant value (eventually).

Swarm

Communication Topologies — FIPS: Fully Informed PSO

For the case where the function $\mathcal{W}(k)$ returns a constant non-zero value:

$$\begin{cases} v^{t+1} &= \chi\left[v^t + \sum_{k \in \mathcal{N}} \left(\frac{\varphi}{|\mathcal{N}|}(y_k - x^t)\right)\right], \\ x^{t+1} &= x^t + v^t. \end{cases} \tag{41}$$

Because all the neighbors contribute to the velocity adjustment, we say that the particle is fully informed.

Swarm

Communication Topologies — FIPS: Fully Informed PSO

Convergence Properties [Montes de Oca and Stützle, 2008]

- In the Model 1'' a particle tends to converge towards a point determined by p, which a weighted average of its previous best y and the neighbourhood's best y^*.
- In FIPS each particle uses the information from all its neighbors to update its velocity, so:
  1. the structure of the population topology has, therefore, a critical impact on the behavior of the algorithm;
  2. when a fully connected topology is used, the performance of FIPS is considerably reduced – the particles explore in a region close to the centroid of the swarm;
  3. the larger the population, the stronger is the bias toward the centroid of the swarm, therefore, increasing the diversity of the population by making it larger, does not work (!);
  4. enhancing the exploratory capabilities of the algorithm by using dynamic restarts provides some benefits but these are problem-dependent.

Clerc, M. (2006).
ISTE (International Scientific and Technical Encyclopedia), chapter Particle Swarm Optimization.

Clerc, M. and Kennedy, J. (2002).
The particle swarm-explosion, stability, and convergence in a multidimensional complex space.
IEEE Transactions on Evolutionary Computation, 6(1):58–73.

Kennedy, J. (2003).
Bare bones particle swarms.
In *SIS*, pages 80–87. IEEE.

Kennedy, J. and Eberhart, R. C. (1995).
Particle swarm optimization.
In *Proc. of the IEEE Int. Conf. on Neural Networks*, pages 1942–1948, Piscataway, NJ. IEEE Service Center.

Mendes, R., Kennedy, J., and Neves, J. (2004).
The fully informed particle swarm: Simpler, maybe better.
IEEE Transactions on Evolutionary Computation, 8(3):204–210.

Montes de Oca, M. A. and Stützle, T. (2008).
Convergence behavior of the fully informed particle swarm optimization algorithm.
In *GECCO '08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pages 71–78, Atlanta, GA, USA. ACM.

Shi, Y. and Eberhart, R. C. (1998).
A modified particle swarm optimizer.
In *Proceedings of the IEEE Congress on Evolutionary Computation 1998*, pages 69–73. IEEE Press.

van den Bergh, F. and Engelbrecht, A. P. (2006).
A study of particle swarm optimization particle trajectories.
Inf. Sci., 176(8):937–971.