Visualization of Particle Swarm Optimization Process

Maurice.Clerc@WriteMe.com
2001-08-16

Using a demo version PSOx ($x \geq 7$), you can generate a coordinate data file (see save_visu( )). Then, you can use the program leonardo_visu on this file, specially written for the free Leonardo C compiler (http://www.dis.uniroma1.it/~demetres/Leonardo/). Just for Mac for the moment, but a PC version might arrive.

By running this program on the data file, you obtain a four windows animation, showing the search process.
To illustrate this, here I am attaching three animated GIF files (if possible, open them with QuickTime, so that you can visualize them step by step), for the Sphere function on $[0,10]^{18}$. The convergence process (admissible error = 0.001) needs 207 objective function evaluations for 30 time steps.

Note : I used an experimental PSO8, with adaptive neighbourhood size. With PSO7 and a neighbourhood size of 3 or 5, you need about 500 evaluations.

Swarm topology

(file topology.gif)

At each time step, there is an arrow from the particle *i* to the particle *j* if :

- *j* belongs to the neighbourhood of *i*
- *j* is a better particle than *i*, that is to say its best position ever found is better than the one of *i*.

If you are seeing this in colour, please note the colour coding :

Position	**Color**
Bad	Violet
Average	Dark magenta
Quite good	Red
Good	Orange
Very good	Yellow
the winner particle (see below)	Light green

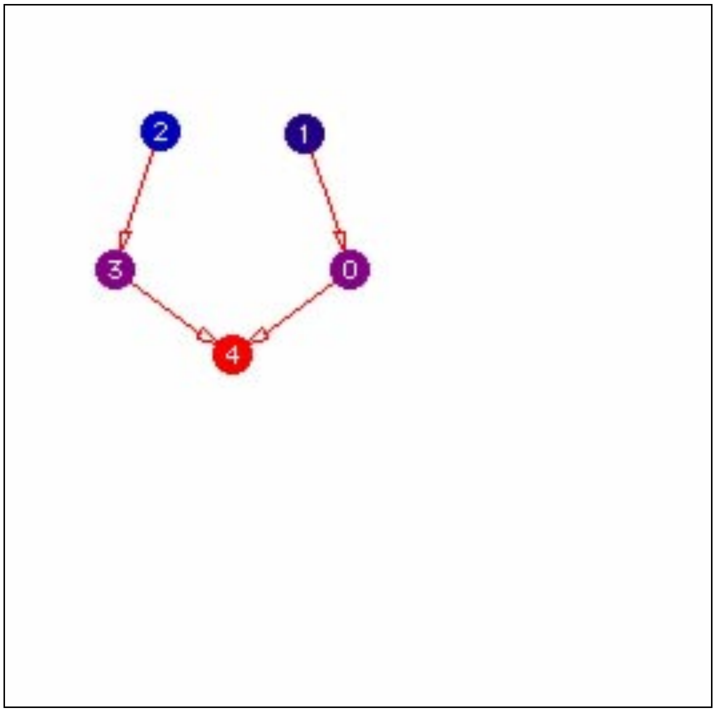

Figure 1. Initial topology (neighbourhood size = 3).

Note that the winner particle, that is to say the one that will reach the solution area first, does not necessarily appear at the very beginning. In this example, it appears after ten time steps.

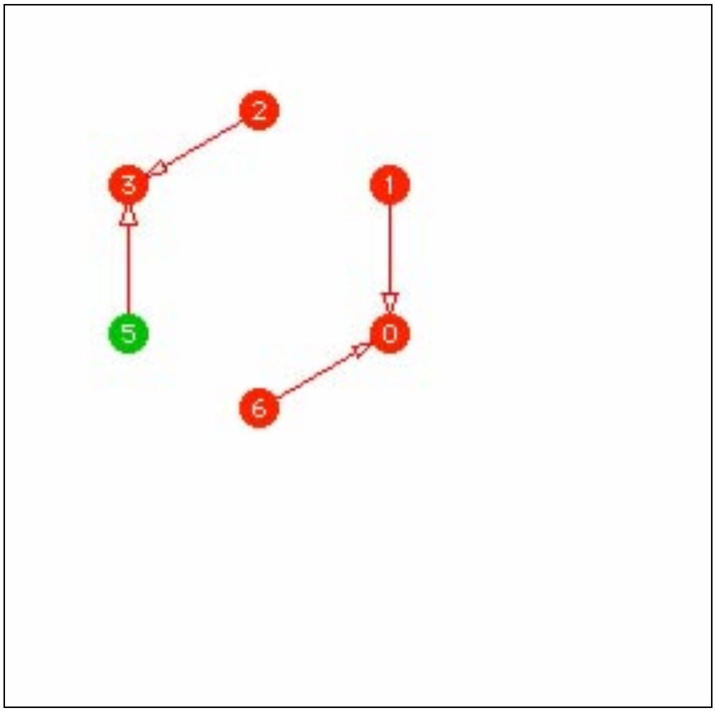

Figure 2. After ten time steps, the bad particle 4 is removed, and particle 5 and 6 are generated. Particle 5 will be the winner. Neighbourhood size is still 3.

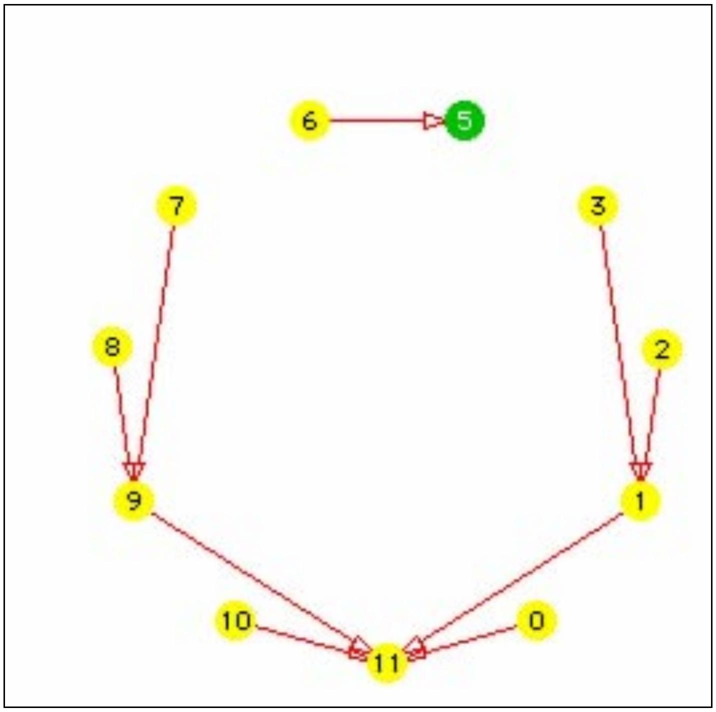

Figure 3. Swarm topology at time 28. Neighbourhood size = 5. Particle 11 is the best, but not in the neighbourhood of 5.

Swarm position, and best swarm position

(file current_swarm.gif)

This representation is done in two dimensions, X1 and X2. If the problem has indeed just two dimensions, the figure can easily show the real position of the swarm. However, for higher number of dimensions (x_1,x_2,...,x_D), the following transformation helps to restrict the representative dimensions to two. First a transformation is done so that the solution is O=(0,...,0) ; then X1 is computed as the distance of the particle to O; and X2 is computed as the shortest Euclidean distance from the position to the axis of symmetry for the coordinate system, i.e. to the line where x_1=x_2=...=x_D.

For Graph Coloring problems, a different set of dimensions is used. Here, X1 is the number of unsatisfied constraints and X2 the number of colors unnecessarily used (in excess compared to the optimum number).

You can easily change the meaning of X1 and X2 in the PSOx C source code (see tot()).

The same representation is used for the best swarm position, that is to say best positions ever found. Of course, this « virtual swarm » usually moves more slowly than the real one.

In both cases, you still have the winner particle, and also its trajectory, in light green.

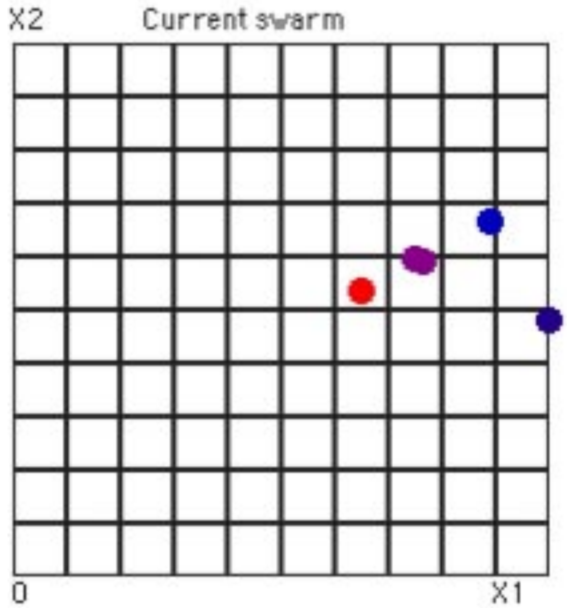

Figure 4. Initial swarm position.

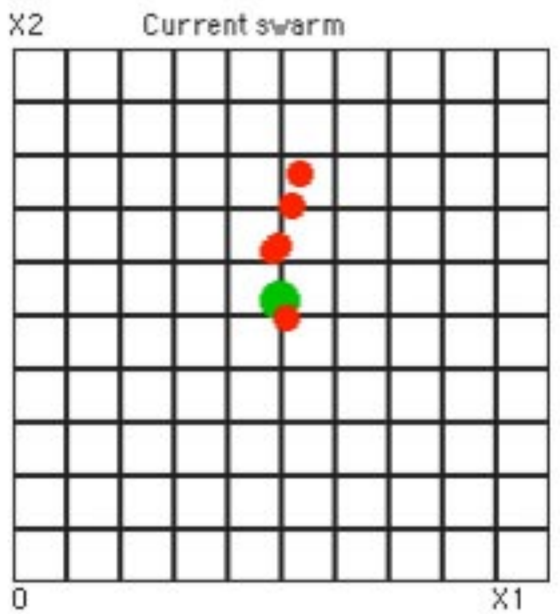

Figure 5. Swarm position at time 10. Swarm size = 6.

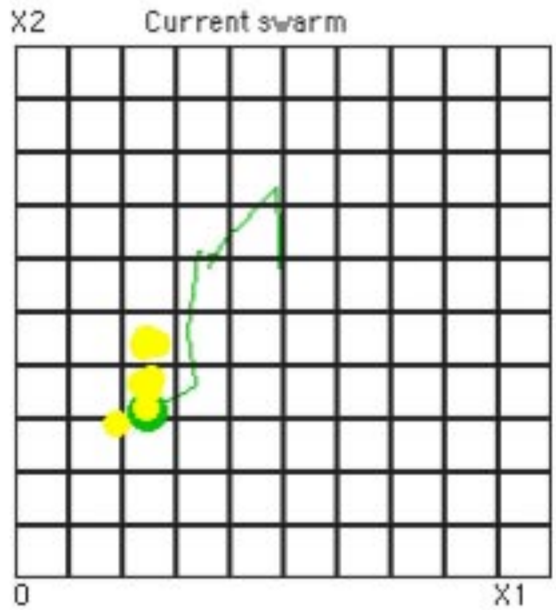

Figure 6. Swarm position and winner's trajectory at time 28. Swarm size = 11, neighbourhood size = 5.

Winner's coordinates

I use here a « star representation ». Each coordinate is shown on its own axis, eventually after a transformation so that all coordinates are positive or null. This representation shows how the winner goes toward the solution.

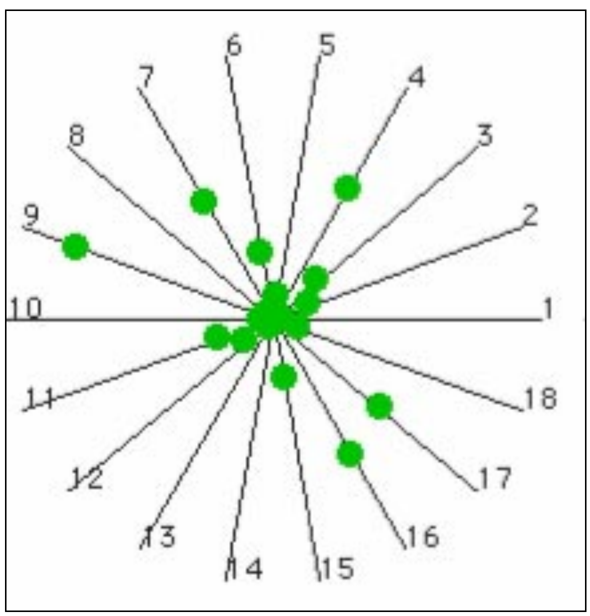

Figure 7. Winner's coordinates at time 10, when it appears. Note the bad 9th one.

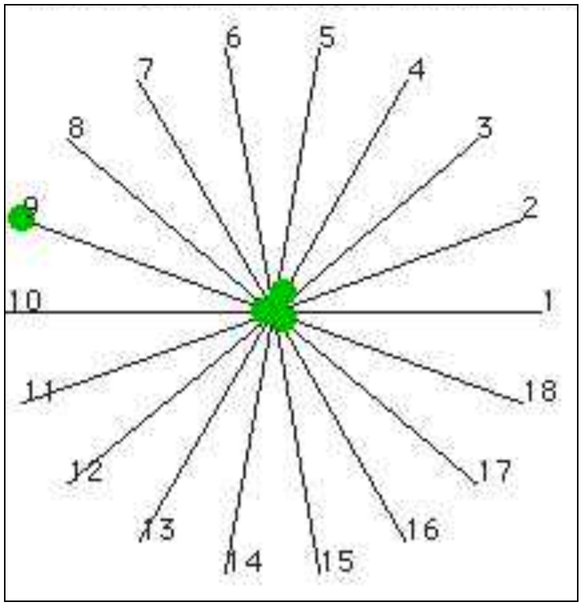

Figure 8. Winner's coordinates at time 25.

Analyzing the example (Sphere function, 18 dimensions)

(Note : better to look at the animations)

Time step	**Comment**
0-1	The initial five particles improve their positions.
2-9	They still try to find better positions, but, finally, there is not enough improvement, so
10-21	a bad particle is removed (4) and two are generated (5 and 6). For the moment 5 is not in the best position, but it finally will be the winner. It has been generated with a very bad 9th coordinate, so it has some difficulty to improve it. At first, there is some global improvement, but after that
22-24	almost no move and even the topology doesn't change, so
25-31	a lot of particles are generated (7, 8, 9, 10, 11). Thanks to their contribution the swarm nicely moves towards the solution.