CS 3445 – Spring 2015
Project 2

A Comparison of Neighborhood Topologies in Particle Swarm Optimization

Due Date: Friday, 6 March 2015

I would like you to test the impact of neighborhood topology on the performance of the Particle Swarm Optimization (PSO) algorithm on a set of three common benchmark functions.

Neighborhood Topologies in PSO

There are a number of common neighborhood topologies, specifying which other particles are in a particle's neighborhood, that people have devised. I would like you to test three of these plus one version of a random topology:

Written in Majercik's Code (Phoebe writes)

- *global*, in which every particle's neighborhood is the entire swarm,
- *ring*, in which the particles are imagined to be in a ring and the neighbors of each particle are the particles to the left and right of it, (Ryan writes)
- *von Neumann*, in which the particles are imagined to be in a grid (that wraps around in both directions), and the neighbors of each particle are the particles above, below, and to the left and right of it, and (Sawyer writes)
- a version of *random* neighborhoods, in which a random neighborhood of size k is chosen for each particle on each iteration by choosing $k-1$ other particles randomly without repetition. (David writes)

Note that, for all of these, the particle itself can be included in the neighborhood or not, but I am asking you to just test the versions in which the particle *is* included.

Test Functions for the Experiments

Your tests should include these functions: Rosenbrock (Ryan), Ackley (Sawyer), and Rastrigin (David). You are welcome to use my code for these functions from Lab 2, but you will need to modify the code so that it can compute the function in $d > 2$ dimensions.

Typically, in tests of these functions, each element of each particle's initial position vector is initialized to a random value in a specified range that does *not* include the location of the minimum (to negate any algorithmic bias toward the center of the search space). So, for example, the global minimum for the sphere function is at the origin (in however many dimensions we are testing), but particles are initialized such that the value of each element in the position vector is in the interval [50,100]. A list of the initialization ranges (which is the same for each dimension for a given function) is given here:

- Rosenbrock: [15.0, 30.0]
- Ackley: [16.0, 32.0]
- Rastrigin: [2.56, 5.12]

Each element of each particle's initial velocity vector is initialize to a value in a specified range. You can use the following:

- Rosenbrock: [-2.0, **2.0**]
- Ackley: [-2.0, 4.0]
- Rastrigin: [-2.0, 4.0]

Code Requirements

Your code needs to implement the basic constriction factor PSO algorithm using a constriction factor of $\chi = 0.7298$. The user should be able to indicate on the command line the following, *in this order and using the indicated labels*:

- which neighborhood topology to test (`gl`, `ri`, `vn`, `ra`)
- the size of the swarm,
- the number of iterations,
- which function to optimize (`rok`, `ack`, `ras`), and
- the dimensionality of the function.

The required experiments fix the number of iterations at 50,000 and the dimensionality at 30, but I want you to write your code so that these could be changed, if desired.

Experiments

Your tests will focus on the effect of different neighborhood topologies on performance. The performance can change, however, depending on other characteristics, such as the size of the swarm, the number of iterations, the function being optimized, and the dimensionality of the function. For example, given 20 particles and 30 dimensions, it may be that one neighborhood generally produces better results, but needs more iterations to do so. In fact, this is generally the case with the global topology and the ring topology. The reduced information flow in the ring topology often leads to better performance but only after a greater number of iterations.

I would like you to run tests with ~~10~~ 12, 20, and 50 particles. For your experiments with the random topology, you should set k, the number of particles in the neighborhood to 5. Each function should be tested in 30 dimensions. Allow 50,000 iterations for each run and do 50 runs.

The normal inclination would be to measure the average and standard deviation of the best function value obtained in each run over the 50 runs, and you should do this

(although you can just report the mean and not the standard deviation). But, since good performance here means a function value of 0.0, it could be that a topology and swarm size you are testing does extremely well in 49 of the 50 runs, but very badly in one run, so the average would look bad. The median would give a much better indication of the performance. So, for each of the 36 cases (4 neighborhood topologies $\times$ 3 swarm sizes $\times$ 3 functions) you should measure the median of the best function value obtained in each run over all 50 runs. Also, I want you to measure this for each iteration (or each 10 iterations, if that's easier to deal with). The reason is that even if two cases end up with approximately the same median, it might be that one of them produces better results earlier in the allowed number of iterations, and we want to be able to see that.

Report

The structure of your report should be the same as in Project 1, but note the following, in particular:

You should be sure to include graphs showing the median at each iteration (or each 10 iterations) for each of the 36 cases. Of course, you're not going to want to put 36 plots on one graph. Since we are interested in how these cases compare for a given function, a good way to break this up would be to have three graphs (one for each test function) comparing the performance of the 12 cases (4 neighborhood topologies $\times$ 3 swarm sizes).

In addition to these graphs, it would be a good idea to split things up further by topology or swarm size. In other words, for a given function and a given topology, compare the plots for the three different swarm sizes. Or, for a given function and a given swarm size, compare the plots for the four different topologies. Or, better yet, do both (on separate graphs) so the reader can see the impact of swarm size in one case, and topology in the other case. Be careful to keep the scale of the y-axis the same on graphs that you are comparing to each other; otherwise, a visual comparison is extremely difficult.

You should also have a table (or tables) presenting some of your data. Table 1 in the GR-PSO paper I gave you as an example for the first project is an excellent guide here. In that paper, for each function, I showed the results for each of 8 algorithms I tested. I reported the mean and standard deviation, and then the median error at intervals of 2,000 function evaluations. In your case, for each function, you will have results for 12 topology and swarm size combinations. You could report the mean function value, and then report the median function value (which is the same as median error, since the optimum is 0.0) at intervals of 10,000 iterations. You are welcome to use the Latex code for the table in my paper as the basis for your table.

What to Hand In

Please submit a folder containing a copy of your code and your report on BlackBoard. All the files I need to run your program should be in a directory that includes a `README` file containing clear instructions for running your program. Your code should be documented well enough that I can easily see what you are doing. *Also, please submit a hard copy of*

your report.

Grading

Since I have prescribed the tests you should run, the grading will be split between your code and your report:

50%: the correctness and clarity of your code,

50%: the content, organization, and clarity of your report.