CS 3445 – Fall, 2018
Project 2

A Comparison of Neighborhood Topologies in Particle Swarm Optimization

Due Dates: October 5 and October 17

In this project, you will test the impact of neighborhood topology on the performance of the Particle Swarm Optimization (PSO) algorithm.

Neighborhood Topologies in PSO

There are a number of common neighborhood topologies. I would like you to test three of them plus one version of a random topology:

- *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,
- *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
- a version of *random* neighborhoods, in which a random neighborhood of size k is initially created for each particle by choosing $k-1$ other particles randomly without repetition and, in subsequent iterations, a particle's neighborhood is recreated in the same way, **with a probability of 0.2**.

Note that a particle is included in its own neighborhood.

Test Functions for the Experiments

Your tests should include these functions: Rosenbrock, Ackley, and Rastrigin. 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. I am providing a paper that should be helpful here: "Defining a Standard for Particle Swarm Optimization." Table 1 gives the equations in d (D in the paper) dimensions for many benchmark functions, including the three I want you to test on.

Typically, in tests of these functions, each element of each particle's initial position vector is initialized to a *different* 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]. The "Feasible Bounds" in Table 1 of the paper I gave you include the location of the minimum, so ignore those. Here are the initialization ranges you should use:

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

In addition, each element of each particle's initial velocity vector is initialized to a different 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

Please use Java or C++.

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 easily specify the following when running your program:

- 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 10,000 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 16, 30, and 49 particles. For 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 10,000 iterations for each run and do 20 runs.

The normal inclination would be to measure the average and standard deviation of the best function value obtained in each run over the 20 runs, and you should do this (although you can calculate and report just 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 19 of the 20 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 20 runs. Also, I want you to measure this for each 1,000 iterations. 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 (although there is only one basic algorithm, instead of two, and there's not much to say about the test problems, except that they are widely used benchmarks for PSO algorithms), but note the following, in particular:

You should be sure to include graphs showing the median at each interval of 1,000 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, you should 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. And, for a given function and a given swarm size, compare the plots for the four different topologies. 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 with 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, and graphed these for every 1,000 function evaluations. In your case, you will be showing data at intervals of 1,000 *iteratons*, not function evaluations. 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 1,000 iterations. You are welcome to use the Latex code for the table in my paper as the basis for your table.

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.

Deadlines

I'm requiring the same intermediate deadlines as before.

Friday, October 5:

As before, submit a hardcopy of your report, *except for*:

- the few sentences in the Abstract (not optional this time) that will summarize your results,
- the few paragraphs in the Introduction that will briefly explain your results, and
- Sections 5 through 8 (refer back to the paper description in the Project 1 handout, if necessary).

I will review your paper and provide feedback.

Wednesday, October 17:

- Submit a folder containing a copy of your code and your finished 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.
- Submit a hard copy of your report.
- Submit, **individually**, a confidential report assessing each group member's contribution to the project.

The Week of October 22:

- A project review session with me. These sessions should take between half an hour and an hour. I will ask you questions about your code, e.g. where the code is that does a particular thing, why you did things in a particular way, etc. and your report. I will expect anyone on the team to be able to answer any questions, so even if, for example, someone did not work on the code, they should still understand it well enough to be able to answer questions about it. And everyone should be able to talk about what's in the paper. I will also ask you to run your program for me on some test problems.