EVCO Practical 3

Learning Objectives

- To gain familiarity with the ECJ - a Java-based evolutionary computation toolkit, and an alternative to using the MATLAB Global Optimisation toolbox. This toolkit contains 'out-of-the-box' GA and GP algorithms, is very flexible, and can be extended easily. For these reasons, it is widely used by researchers in evolutionary computation, particularly for Genetic Programming (GP).
- To solve problems using Genetic Programming.

Prerequisites

- An understanding of Genetic Programming concepts (EVCO Lecture 6).
- Some (basic) knowledge of Java will help, but is not essential.

  ECJ is based on Java, both in terms of its general structure and concepts, and the custom code that is used to define the specific problem you intend to solve and extend the system if the built-in functionality is insufficient. However, for this practical, all the Java code that is required is already supplied as part of the ECJ tutorials. If you are new to Java, don't be concerned about the actual code used, but concentrate instead on the general concepts and treat the Java code as a "black box" that implements some of these concepts. Of course, please be sure to ask the demonstrators if you have any questions about the Java code.

Linux

Please use **Linux** for the practical. (The Java configuration in the teaching lab version of Windows requires you to perform some additional setup, and we suggest you avoid this complication.)

Disk Quota

You should ensure that you have at least 20 MB free in your home directory for this practical.

Simon Poulding and John Clark, October 2011

Part 1 - ECJ Installation

The first task is to download and install ECJ version 20 in your home directory.

1. Go to the ECJ website at: http://cs.gmu.edu/~eclab/projects/ecj/
2. Midway down the page, there is a link to download the ECJ version 20 compressed as a zip file. Download and save this zip file to your home directory.
3. Start a terminal session and change to the directory where you downloaded the file.
4. Unzip the file. You should see now see a `ecj` subdirectory containing files such as `README`, and directories such as `ec`.
5. Add this `ecj` directory to your Java CLASSPATH. If you are **in** the `ecj` directory, you can use the following command to achieve this:

   ```
   export CLASSPATH=${CLASSPATH}:${PWD}
   ```

   (You will need to set the CLASSPATH whenever you start a new terminal session to use ECJ. Therefore, if you know how to, you may want to edit your profile to set the classpath automatically. Be sure to specify it absolutely rather than rely on the shortcut of `${PWD}` as used above.)
6. Change to the `ec` subdirectory beneath `ecj`, and delete the `display` subdirectory. (Classes in the display directory will not compile correctly, and are not required in this practical.)
7. In the `ec` directory, run the following command to compile all the Java classes:

   ```
   javac -O *.java */*.java */*/*.java */*/*/*.java
   ```

   Do not worry about the 11 error messages, most of which refer to `app/gui`. If you have more than 11 error messages and 1 warning, check with one of the demonstrators.
8. To check if ECJ is installed correctly, run an example using the following command:

   ```
   java ec.Evolve -file app/sum/sum.params
   ```

   You should get output that begins as follows and proceeds to report fitnesses at each generation. If not, please let one of the demonstrators know.

```
| ECJ
| An evolutionary computation system (version 20)
| By Sean Luke
| Contributors: L. Panait, G. Balan, S. Paus, Z. Skolicki, R. Kicinger, E. Popovici,
|               K. Sullivan, J. Harrison, J. Bassett, R. Hubley, A. Desai, A. Chircop,
|               J. Compton, W. Haddon, S. Donnelly, B. Jamil, J. Zelibor, E. Kangas,
|               F. Abidi, H. Mooers, and J. O'Beirne
| URL: http://cs.gmu.edu/~eclab/projects/ecj/
| Mail: ecj-help@cs.gmu.edu
|       (better: join ECJ-INTEREST at URL above)
| Date: December 15, 2010
| Current Java: 1.6.0_26 / Java HotSpot(TM) 64-Bit Server VM-20.1-b02-383
| Required Minimum Java: 1.4


Threads:  breed/1 eval/1
Seed: 1505709091
Job: 0
Setting up
```

Part 2 - ECJ Tutorial 1

We will learn about ECJ during this practical using the tutorials that are shipped with the tool.

To access the tutorials, open the file `docs/index.html` (the `docs` directory is under the main `ecj` directory) in a browser. The first section lists some reference documentation that you might find useful, such as the manual and an overview of ECJ. However, we will start with the tutorials that are listed in the next section of the web page. (The tutorials and manual are also available from the main ECJ website.)

In this part of the practical, follow Tutorial 1. The key points to understand are:

1. The use of parameter files that control every aspect of ECJ (equivalent in purpose to the settings of optimtool in MATLAB). If you're not familiar with Java-style parameter syntax, you could also read the "parameter and parameter files" primer accessed from the link in the first section of the webpage.
2. The basic concepts of ECJ such as populations, species, evaluators, breeding pipelines etc.
3. How to define the problem/fitness evaluation using a Problem class written in Java.

The suggested directory `tutorial1` already exists under the `ec/app` directory, and this directory already contains all the parameter files and code that you need for tutorial. Therefore you don't need to write these files yourself (as the tutorial suggests), but you **must** make sure you understand the contents of the files.

You should also follow the links to some of the other ECJ documentation (e.g. the ECJ Javadoc) for more details. (I've found that a few of these links are broken, in which case access the Javadoc from a working link, and navigate to the class/package you want.)

There is one discrepancy between the tutorial and the supplied files: the genome-size in the latter is 20, and you should stick with this value. You will probably also find that your results are not identical to the ones shown in the tutorials for various reasons, but they should be qualitatively similar.

Part 3 - ECJ Tutorial 2

This tutorial extends the basics covered in Tutorial 1.

The key points to understand are:

1. The flexibility of default and parent parameters, and passing parameters on the command line.
2. How to build custom breeding pipelines.
3. Specifying random seeds.
4. Multi-threaded operation.

Part 4 - ECJ Tutorial 4

We **skip** Tutorial 3 on building Evolutionary Strategies algorithms, and move straight on to Tutorial 4 that describes how to use ECJ for Genetic Programming.

Tutorial 4 is a little more involved that the previous tutorials, with substantially more Java code. The key points to understand are:

1. The additional concepts and classes required for GP: such as nodes, trees, function sets, and the GPData object.
2. How to specify your own node functions and terminals as Java classes.
3. How to specify a Problem class for a regression problem.

Don't worry too much about the more advanced stats (which may not be working correctly in any case) and printing out trees in LaTeX etc., unless you are interested in these neat features.

Part 5 - ECJ GP Examples (Optional)

If you have some time left after completing Tutorial 4, you should explore the configuration of the built-in examples described at the bottom of the main documentation page (`docs/index.html`), particularly some of the 'classic' GP problems.