

ECJ 23

A Java-based Evolutionary Computation Research System

By Sean Luke, Liviu Panait, Gabriel Balan, Sean Paus, Zbigniew Skolicki, Rafal Kicinger, Elena Popovici, Keith Sullivan, Joseph Harrison, Jeff Bassett, Robert Hubley, Ankur Desai, Alexander Chircop, Jack Compton, William Haddon, Stephen Donnelly, Beenish Jamil, Joseph Zelibor, Eric Kangas, Faisal Abidi, Houston Mooers, James O'Beirne, Khaled Ahsan Talukder, Sam McKay, and James McDermott.

ECJ is a research EC system written in Java. It was designed to be highly flexible, with nearly all classes (and all of their settings) dynamically determined at runtime by a user-provided parameter file. All structures in the system are arranged to be easily modifiable. Even so, the system was designed with an eye toward efficiency.

ECJ is developed at George Mason University's [ECLab](#) Evolutionary Computation Laboratory. The software has nothing to do with its initials' [namesake](#), *Evolutionary Computation Journal*. ECJ's sister project is [MASON](#), a multi-agent simulation system which dovetails with ECJ nicely.

New Book!

Sean's got a free online text, [Essentials of Metaheuristics](#). Over 200 pages of goodness. Check it out!

Features

General Features

- GUI with charting
- Platform-independent checkpointing and logging
- Hierarchical parameter files
- Multithreading
- Mersenne Twister Random Number Generators
- Abstractions for implementing a variety of EC forms.

EC Features

- Asynchronous island models over TCP/IP
- Master/Slave evaluation over multiple processors, with support for generational, asynchronous steady-state, and coevolutionary distribution
- Genetic Algorithms/Programming style Steady State and Generational evolution, with or without elitism

GP Tree Representations

- Set-based Strongly-Typed Genetic Programming
- Ephemeral Random Constants
- Automatically-Defined Functions and Automatically Defined Macros
- Multiple tree forests
- Six tree-creation algorithms
- Extensive set of GP breeding operators
- Grammatical Encoding
- Push
- Many pre-done GP application problem domains, including ant, regression, multiplexer, lawnmower, parity, two-box, edge

Vector (GA/ES) Representations

The following table lists the available representations for the various EC algorithms.