Python Bytecode Evolution using Stack-based Genetic Programming for Regression Problems

Hyunsoo Park, Kyungjoong Kim
Department of Computer Engineering, Sejong University
E-mail: hspark@sju.ac.kr, kimkj@sejong.ac.kr

Abstract:

Traditional genetic programming uses tree-like data structure to represent a program. It should be converted into a Lisp code, or needs a custom-made virtual machine or interpreter to execute the program generated. Therefore, these methods need extra efforts and/or endure slow execution speed. Recently, there is a study on genetic programming directly using Java bytecode, a practical intermediate language. It evolves a series of commands that manipulate stack and registers in a virtual machine and represents them with a simple list data structure instead of tree. Evolving the intermediate language is promising because 1) it is easy to combine an existing program with an automatically generated program, 2) there are several available development tools and environments for the language including virtual machine, decompiler, optimizer and so on, and 3) incorporating the list data structure into the evolutionary algorithm is simple and straightforward. In this research, we propose to evolve bytecode of Python programming language by stack-based genetic programming. Python is a flexible and popular programming language boosted by a lot research tools. In detail, PyPy is used as a Python interpreter to verify the correctness of bytecode and byteplay module can manipulate the bytecode in runtime. We show this method can solve simple regression problems.

Keywords: Genetic programming, Python programming language, Bytecode