Summer Solstice 2018 Conference on Discrete Models of Complex Systems

Gdańsk, June 25-27, 2018

Cartesian Genetic Programming with memory

Maciej J. Mrowinski, *Faculty of Physics, Warsaw University of Technology*

Cartesian Genetic Programming (CGP) is an evolutionary programming algorithm whose purpose is to evolve computer programs using concepts inspired by natural selection. It's range of applications is wide and includes problems like optimisation or image processing. Programs in CGP are encoded as graphs. The structure of an encoded CGP program is very similar to a multilayer perceptron network with different activation functions assigned to nodes and with equal weights of all connections. A CGP program is evolved using the so called 4+1 algorithm, which tries to maximise a user-provided fitness function by creating, via mutation, new generations of programs.

Recurrent Cartesian Genetic Programming (RCGP) is a variant of CGP which allows cycles in the graphs representing programs, which implicitly introduces memory into CGP. The addition of memory broadens the range of applications of CGP and makes it a more viable tool for problems like time series forecasting. In our work, we propose a modification of CGP which results in an explicit inclusion of memory. We achieve this by directly including a shift register in each node of the CGP graph and constantly providing these registers with values processed by nodes. Thanks to this approach (which we call SRMCGP – Shift Register Memory CGP), users gain fine-grained control over the memory in the program, which is not possible in RCGP, and avoid forward, recurrent connections. In order to study the memory capabilities of RCGP and SRMCGP, we performed numerical simulations of programs whose purpose was to memorise and repeat the input signal after a given number of time steps. Our results suggest that SRMCGP is much more efficient than RCGP – usable solutions/programs can be acquired faster through SRMCGP. Additionally, SRMCGP results in a smaller number of active nodes which makes SRMCGP programs less costly (it terms of computation time) to decode. SRMCGP is also more likely to actually create usable solutions/programs.