Automatic invention

I think therefore I CGP

Cartesian Genetic Programming

Genetic Programming (GP)

Genetic Programming is concerned with the automatic evolution (as in Darwinian evolution) of computational structures (such as mathematical equations, computer programs, digital circuits, etc.). The best known figure in this field is

John Koza .

The form of GP pioneered by John Koza used a tree representation of computer programs. This was inspired by the artificial intelligence computer language, LISP.

Cartesian Genetic Programming (CGP)

CGP is a **highly efficient and flexible** form of Genetic Programming that encodes a graph representation of a computer program. It was invented by Julian Miller in 1999 and was developed from a representation of electronic circuits devised by Julian Miller and Peter Thomson developed a few years earlier.

CGP represents computational structures (mathematical equations, circuits, computer programs etc) as a string of integers. These integers, known as genes determine the functions of nodes in the graph, the connections between nodes, the connections to inputs and the locations in the graph where outputs are taken from.

Using a graph representation is very flexible as many computational structures can be represented as a graph. A good example of this is artificial neural networks (ANNs). These can be easily encoded in CGP. Recent published results of Khan and Miller and turner and Miller(see the **publications** linked from this site) show that using CGP to encode and evolve ANNs (CGPANNs) is highly efficient and competitive with other methods of evolving ANNs. **Here is a ten-slide introduction to CGP** .

What can CGP be used for?

Machine Learning, Neural Networks, Artificial Intelligence, Data Mining, Financial prediction, Function optimization, Classification, Electronic circuit design, medical diagnostics, evolutionary art and music,..., the list is endless.

To learn more

Buy the CGP book edited by Julian Miller (cheapest at amazon).

Cartesian Genetic Programming

Download a tutorial by Julian Miller and Simon Harding

GECCO 2012 Tutorial on CGP .

News highlights

Self-modifying Cartesian Genetic Programming software is available

Julian Miller has just finished working on a C implementation of SMCGP that can solve boolean and mathematical problems. See the resources section of this website.

CGP Image Filters

CGP is being used in a European robotics project. Simon Harding has developed a neat way of using CGP to evolve object recognition image filters. Once the filters are evolved, they can operate extremely fast and can pick out objects of interest in images. The filters are orientation independent and still work very well under different lighting conditions. There is a nice video at IM-CLeVeR European project Here is an video of an evolved object recognition program. It detects a packet of tea!. It can detect the packet in any orientation, proximity and lighting condition.

Machine Intelligence Ltd.

Simon Harding has founded his own company Machine Intelligence Ltd to exploit the power of CGP for automatic visual defect detection, and process control and optimization. Here is a interesting video showing how software detects faults in a metallic surface.

Google Lunar X prize

CGP is being used by a team who are aiming to win the Google Lunar X prize. This is a prize awarded to the first private venture to land a rover vehicle on the moon. Wes Faler, who is a member of a team called "Part-time scientists" are building a rover called "Asimov 1". Wes gave talks at a conference called 28C3 in Berlin, Germany about two ways he is using CGP in the project. His talks are available on his blog