**Genetic programming / memetic algorithms – REVIEW**

**Harri Juutilainen**
**haeejuut@student.jyu.fi**

## Genetic programming

The basic idea of genetic programming is to solve a problem via a high level function call without telling the computer how to do it. This is done via a method similar to evolutionary algorithms but the population that are evaluated and bred are programs instead of binary strings like in GA:s. The population of programs are bred into fitter ones and evaluated again, adding the functionality of parents into the offspring. The mutation and crossover operations differ from regular GA solutions due to the structure of the handled individuals being different.

Crossover is done commonly with a method called *subtree crossover*. In the method, two crossover points are selected for each parent. The offspring are generated from the subtrees ”under” these crossover points and swapping them over between parents. Mutation is done by choosing a random point in the tree and substituting the subtree with a randomly generated subtree of similar form (*subtree mutation*).

Additional values are needed to run the GP system. These include external inputs, constants, how fitness is measured and termination criterion. Since the algorithm is running mostly functions instead of values, it is important to have safeties for type consistency and evaluation. For example the fittest programs are only selected when it is certain that all individuals are evaluated. These evaluations are done only when the function has gotten the values it requires, which might be asynchronous in these kind of solutions.

Sections which covers the basics from the book: Langdon W.B., Poli R., McPhee N.F., Koza J.R. (2008) Genetic Programming: An Introduction and Tutorial, with a Survey of Techniques and Applications. In: Fulcher J., Jain L.C. (eds) Computational Intelligence: A Compendium. Studies in Computational Intelligence, vol 115. Springer

## Memetic algorithms

Memetic computation is defined as a hybrid between population-based evolutionary algorithms and cultural refinement procedures. The term 'meme' is coined by Richard Dawkins as an idea that spreads between individuals in a culture. Memetic computation paradigm uses these ideas for solving problems.

Memetic computation started with simple hybrids between population-based search and refinement procedures. It later evolved into using adaptive strategies because issues were raised in determining parameters and operators that fit the structure of the problem. In the simple hybrids, these parameters were determined by human experts during the design process. The aim of the evolution of memetic computation has been to counter the need for solving increasingly complex problems.

The next step in this evolution is 'memetic automatons'. These are defined as self-contained adaptive entities that use memes to store information. The high-level idea is to have a framework of these entities (memetic automatons) that can automatically adapt to different dynamic environments. The main aspects of memes in memetic automatons according to the paper are representation and evolution. Representation means the ability of the meme to represent the knowledge it has to others. As according to the definition of meme, this knowledge can be expressed to others in an attempt to influence their behaviour. The evolution is done via this expression and also with the ability to self-generate and adapt to situations.

A Multi-Facet Survey on Memetic Computation, Xianshun Chen, Yew-Soon Ong, Meng-Hiot Lim, and Kay Chen Tan, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 15, NO. 5, OCTOBER 2011