Multiobjective Grammatically-based Genetic Programming

Author: Jiří Petrlík

The aim of the thesis is to describe an approach for automatic generating simple programs based on evolutionary computation techniques and grammars. Automatically generated programs can be optimized in few objectives, mainly correctness and program size. The first part of the thesis describes a principle of evolutionary computation techniques, especially a genetic programming and a grammar based genetic programming. The second part describes principles of a multiobjective optimization and the third part presents few examples of usage of a grammatically based genetic programming and achieved results.

Evolutionary computation techniques are widely used in the field of optimization. They are inspired in nature, especially in biological evolution. These techniques are used for searching in huge state spaces and finding the best solutions of current task. Usually, at the beginning, they start with randomly generated solutions. For correct work of algorithms based on evolutionary computation techniques it is necessary to be able to evaluate a quality of a candidate solution. Then, few best generated solutions are used in processes of crossover and mutation which are able to generate new candidate solutions. A quality of these newly generated solutions is compared with previously generated solutions. Solutions with a better quality are preferred in the new generation. This principle is called elitism.

Comparison of two solutions is usually done by a fitness function. As a better solution is considered a solution with the highest value of the fitness function. But sometimes there could be a problem to evaluate quality of a solution in only one fitness function. Especially, when objectives that should be optimized are contrary. In this situation usually one fitness function per an objective could be obtained. One of the often used approach how to solve this problem is to use a weighted sum of all objectives fitness functions as a resulted fitness function. There could be a problem with how to weigh objective fitness functions. Another more sophisticated approach is to use a non-dominated sorting, which is based on principle of Pareto optimality. One of the most used algorithm for this approach is NSGAII.

The genetic programming is an evolutionary based method for automatic generating of computer programs. The method is in the principle the same as genetic algorithms, however runnable programs are used as candidate solutions. Candidate solutions (programs) are usually represented as trees. Nodes of these trees are chosen from two sets. Non-leaf nodes are chosen from the set of functions and leaves are chosen from the set of terminals. A set of terminals usually consists of constants or functions with a side effect.

In basic genetic programming, every member of the set of functions must be well defined for any input. This is necessary, because there are no restrictions on connecting tree nodes. In contrast with a basic genetic programming, in grammar based genetic programming, there is a defined grammar (typically context free grammar) which restricts how could be tree nodes connected. A multiobjective optimization is typically used for optimization of the programs size.

A grammatically based genetic programming has been used for solving many tasks. For example, it was used for generating rule induction algorithms, a topology of neural networks and a symbolic function regression.