Racing-Based Genetic Programming

J.-B. Hoock and O. Teytaud

TAO (Inria), LRI, UMR 8623(CNRS - Univ. Paris-Sud),
bat 490 Univ. Paris-Sud 91405 Orsay, France, hoock@lri.fr

Abstract. Genetic Programming (GP) is the automatic building of programs for solving a given task. The main troubles are (i) the price of the evaluation of a mutation (ii) the size of the huge set of possible mutations. In many cases, one more trouble comes from the noisy evaluation; the fitness function, evaluating the quality of a program, is stochastic. When evaluating a program, one spends a time (supposed constant) for running it, and gets a noisy reward: we are looking for

$$\arg \min_{x \in \text{possible programs}} \mathbb{E} fitness(x).$$

When mining a huge set of mutations in an uncertain framework (noisy optimization), there are two main issues:

- **load balancing:** which mutations of the program are to be tested now ?
- **statistical validation:** which mutations of the program should be validated ?

The statistical validation is not trivial; whenever each mutation is tested rigorously, e.g. with confidence 95 % (i.e. 5% of probability of error), we will have erroneous validations very frequently if millions of mutations are tested (with just 100 mutations, each of them being useless, we have a probability 99.4% ($= 1 - (1 - 0.95)^{100}$) of erroneous validation). This is known as the **Multiple Simultaneous Hypothesis Testing (MSHT) effect.**

Bandits have been used for a while, even in GP, for the load balancing problem, but not yet, to the best of our knowledge, for both issues simultaneously. We here investigate the use of **racing algorithms**, a particular form of bandit algorithms covering simultaneously the load balancing and the statistical validation.

Our algorithm, termed RBGP (racing-based GP) depending on parameters δ and c, such that the following two properties hold simultaneously with probability $1 - \delta$:

$$\textbf{Monotonic improvement:} \mathbb{E} fitness(x_n) > \mathbb{E} fitness(x_{n+1});$$

and (**efficiency**) if there exists a possible mutation of x with improvement at least c, then x_{n+1} contains a mutation with positive effect that was not in x_n.

We applied our results to the GP of the bias module of MCTS, a well-known problem which is difficult even for humans; whereas humans often do mistakes due to the multiplication of trials (the MSHT effect above), RBGP provided improvements comparable to humans developers, and more importantly was never convinced of sending a mistake in the CVS archive of the project.