On Termination Criteria of Evolutionary Algorithms

Brijnesh J. Jain
Department of Computer Science
Technical University Berlin
Franklinstr. 28/29
D-10587 Berlin, Germany

Hartmut Pohlheim
DaimlerChrysler AG
Research and Technology
Alt-Moabit 96a
D-10559 Berlin, Germany

Joachim Wegener
DaimlerChrysler AG
Research and Technology
Alt-Moabit 96a
D-10559 Berlin, Germany

For the practical use a termination criterion should determine the end of a search process as soon as the EA is not sufficiently efficient, i.e. when the EA is degenerated to a random search or no significant improvements of the best objective value can be expected within a forseeable space of time. The significance of an improvement is highly dependent on the desired precision of the results and on the envisaged goals of the problem solving process. Each improvement corresponds to an economical profit and each iteration causes costs. From this point of view an optimization can run without loss up to a fixed precision. Improvements obtained without loss are considered to be significant.

A suitable point in time to terminate an optimization run prevents premature termination as well as further computations to no avail. Hence the efficiency of a numerical algorithm for optimization is not only dependent on its computational performance but also on its behavior to terminate a run. Furthermore in some real-world applications of EA there are attempts to automate the optimization process as in evolutionary testing. In evolutionary testing, temporal correctness of real-time systems is verified with the assistance of EA whereby the objective functions are discrete. Since real-time systems are often safety-relevant, temporal correctness plays an important role. Consequently, reliable and intelligent termination criteria as well as general recommendations how to employ termination criteria for the automated use of EA are required.

Here an overview of partly well-known termination criteria and a newly defined criterion (ClusTerm) based on cluster analysis is given. Conventional criteria either use objective values or the distribution of individuals in the search space to decide the end of a run. To get other, probably more reliable and intelligent termination criteria, it is reasonable to combine and evaluate both, information about objective values and distribution of individuals. This approach is implemented by using cluster analysis on the fittest individuals. The intelligence behavior of ClusTerm is incorporated in the fluctuations of the aggregate sizes of elitist clusters.

All termination criteria were systematically tested by extensive experiments and evaluated with respect to their reliability and performance. Initiated by the necessity of reliable and intelligent termination criteria in evolutionary testing, where the objective functions are discrete, we primarily considered multidimensional step functions in our empirical analysis. This test set can also be used for benchmarking EA strategies on discrete problems. The results of the empirical analysis were verified by real-world applications in evolutionary testing. The criterion ClusTerm proved to be promising for problem domains with discrete objective functions.

Finally, for each termination criterion guidelines for the practical employment and automated application of EA as well as references when certain criteria should not be used are suggested. Obviously, terminiation criteria behave differently for varying EA strategies and objective functions. Therefore it is impossible to formulate a general rule for optimal use of a termination criterion. But in real-world applications like evolutionary testing an automated employment of EA is desired, since software tester often are not familiar with the behavior of EA. For this reason we give some practical guidelines of the application of termination criteria for unskilled practioners and the automated use of EA. Our recommendations are not only restricted to discrete objective functions. They can also be accepted under reserve if one is dealt with continuous optimization problems. Our suggestions for the continuous case are based on first unsystematical experiments with common continuous test functions.