Runtime analysis of evolutionary algorithms

Filip Kuklis, xkukli03@stud.fit.vutbr.cz

Abstract

The use of evolutionary principles to solve some computational problems started in the 1950s. Evolutionary algorithms became widely used for optimization problems in the 1970s after the work of Ingo Rechenberg, who used these algorithms to solve complex engineering problems [1]. However, the functionality of the genetic algorithm was not fully understood. Holland's schema theorem was considered fundamental [2]. It first proposed to understand the behavior of the simple genetic algorithm, but on the other hand, it cannot explain the performance or limit behavior of evolutionary algorithms.

Most often the typical research methodology of EA is to implement the algorithm and apply that algorithm several times to some test problems. Do statistical analysis from test runs and then find out how the algorithm behaves. But in this work, we would like to describe a mathematical theory that allows us to predict the behavior of EA. We would like to know the runtime of EA until finding a solution for any problem size. Also, we would like to know how to parameterize the algorithm to be efficient.

The presentation starts with the quick introduction into genetic algorithms. Describe the idea and main parts of EA – population, selection, mutation fitness function and more. And the origin of these algorithms on the example from nature. Then show how the algorithm works on the pseudocode.

Then we present criteria for evaluating algorithms and its variances which are also applicable to evolutionary algorithms. The problem of EA is that the computation time of the same problem with the same parameters is every-time different because of the randomness of EA. Most relevant computational resource of EA is the number of function evaluations which will be defined.

The main part of the presentation is a runtime analysis of population-based evolutionary algorithms. Starting with the presentation of different approaches. Then we focus on Level-based Theorem[3] approach. Defining the levels of the search space and current level of population. Next, declare the statement of the theorem and its conditions.

Finally, apply Level-based theorem on the simple example – computing parameters to satisfy all theorem conditions and find out the expected running time of the example.

[1] Rechenberg, I. (1973). Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (PhD thesis) (in German). FrommanHolzboog.

*[2]*Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press. ISBN 0-262-58111-6.

[3] Corus, D., Dang, D., Eremeev, A. V., and Lehre, P. K. (2014). Level-based analysis of genetic algorithms and other search processes. In Parallel Problem Solving from Nature - PPSN XIII - 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings, pages 912–921.