УДК 811.111:004.9

Makanov D., Vanik I.

Genetic Algorithms and Their Applications in Computer Science

Belarusian National Technical University
Minsk, Belarus

In Computer Science, there is a variety of well-known problems that help people optimize their certain routine in various fields. Very often, for the same problem, there are several approaches to its solution. In this case, a specialist in this field is faced with the question of choosing the best approach to solving the problem, thanks to which a solution can be obtained in the shortest possible time with the same data. For the purpose, the concept of big O-notation or algorithmic complexity was introduced in Computer science meaning the nature of the dependence of the number of operations in an algorithm on the size of the data being processed. For example, an algorithm that runs in O (1) is executed in the same time for any amount of data, and the execution time for an algorithm with an algorithmic complexity of O (N) depends linearly on the size of the input data.

According to the theory of algorithms, by the form of a function in O-notation, all algorithms can be divided into polynomial (class P) and non-polynomial problems (class NP). For example, problems of integer arithmetic or sorting a set of n numbers are referred to class P, and the latter includes various problems of finding the optimal solution out of millions of possible ones [1].

Genetic algorithms are one of the well-known ways to solve NP class problems. Genetic Algorithms (GA) are search algorithms based on the concepts of natural selection and genetics. They were developed by John Holland and his students and colleagues at the University of Michigan.

Let's consider the example with the "OneMax" task used a genetic algorithm. This problem has an array consisting of zeros and ones. It is necessary to find a solution that would give the maximum sum of the digits of this array. Obviously, the best solution is an array of all ones.

Firstly, it’s necessary to form many individuals of the same species called a population. For this problem, the population is the usual arrays of zeros and ones, which are generated randomly. The quality of the solution for each specific array is defined with the aid of the fitness function, which in the OneMax problem is the sum of all the numbers in the genes. The higher the value of this sum, the better the solution is presented by the individual.

The next steps of any genetic algorithm are determined by the following three evolution processes: selection of the fittest individuals; crossing parents to get new individuals; mutation is a random change in individual genes.

To choose the fittest individuals in the OneMax problem, it is possible to use a tournament selection, in which n individuals are randomly selected in the entire population, and then the most fit is selected among them. Then the winner will be used as a parent in the crossing operation. The selection is repeated with the original population until a new array of individuals of the same number as the original population is generated. Further, for getting new individuals it’s enough to use the single-point crossing scheme, when the little parts of an array cut from the parents is randomly selected, and then their exchange is carried out. To model mutation process some bits in arrays are inverted randomly. All these stages are performed until the best solution is found that satisfies the conditions of the algorithm.

In fact, the whole variety of genetic algorithms is based on various combinations of selection, crossing and mutation methods. The more successful this combination of methods is selected based on the criteria of the problem, the better the genetic algorithm will work.

GAs have various advantages such as efficiency and fast work compared to traditional methods, the ability to solve multi-criteria problems as well. They are also very useful when the search space is very large and many parameters are involved.

GAs also have some disadvantages. GA is not suitable for simple tasks for which derivative information is available. Since the solution is stochastic, there is no guarantee that the solution will be optimal. If not properly implemented, GA may not converge towards the best solution [2].

Basically, such algorithms are applied in engineering design, robotics and in solving logistics issues [3].

References:

1. Algorithms S. Dasgupta [Electronic resource] – Mode of access: http://algorithmics.lsi.upc.edu/docs/Dasgupta-Papadimitriou-Vazirani.pdf – Date of access: 17.11.2021.
2. Genetic Algorithms – Quick Guide [Electronic resource] – Mode of access: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_quick_guide.htm. – Date of access: 17.11.2021.
3. Introduction to Genetic Algorithm [Electronic resource] – Mode of access: https://www.analyticsvidhya.com/blog/2017/07/introduction-to-genetic-algorithm/. – Date of access: 17.11.2021.