GENETIC ALGORITHM AS A METHOD OF SOLVING SELECTED OPTIMIZATION PROBLEMS

Józef Gil

Institute of Building Engineering
University of Zielona Góra
Poland

ABSTRACT

Genetic algorithms, which were created on the basis of observation and imitation of processes happening in living organisms, are used to solve optimisation tasks. The idea of genetic algorithms was presented by Holland [3], and they were developed and implemented for solving optimisation tasks by Goldberg [2]. Choice of particular variables of the vector $\mathbf{w} = [w_1, w_2, ..., w_n]$ in order to maximize or minimize a fitness function takes place as a result of a sequence of genetic operations in the form of selection, crossbreeding and mutation. The article describes the basic genetic (classic) algorithm including its components.

1. INTRODUCTION

Genetic algorithms are included in evolutionary methods, which enable searching global extremes for a wide class of optimisation tasks. In subject literature [2] three types of methods of searching for optimum solutions are mentioned, namely: analytical methods, survey (enumerative) methods and random methods. The use of analytical methods (direct and indirect) is conditioned by the necessity of existence of derivatives of an objective function, which limits the range of their application. Survey methods, the simplest of which consists in surveying the values of an objective function in all the points of the space under research are, briefly speaking, ineffective. The last of the abovementioned methods is the random method, which is in general also ineffective because of a random search for an optimum point in the space of solutions.

A way out of this situation is dealing with the problem by means of genetic algorithms as a general model of adaptation processes proceeding on binary representations of solutions with the use of operations borrowed from genetics. Genetic algorithms, as a search for optimum solutions, make use of a random, multidirectional search through the space of potential solutions to various problems in technical science.

2. ELEMENTS OF GENETIC ALGORITHMS

The idea of genetic algorithms was formulated by Holland [3] in order to recreate mechanisms operating in biological systems and adapting them as a technique of solving mainly tasks of optimisation in a wide spectrum of problems. Differences between traditional optimisation methods and genetic algorithms are as follows[6]:

- **genetic algorithms do not directly process parameters of the task but their encoded form,**
- **they keep searching starting not from a single point but from a certain population of points,**
- **they make use of only an objective function, and not of its derivatives or other additional information,**
- **they make use of probabilistic and not deterministic rules of choice.**

These four features prove the advantage of genetic algorithms as an optimisation method over the other widely used techniques.

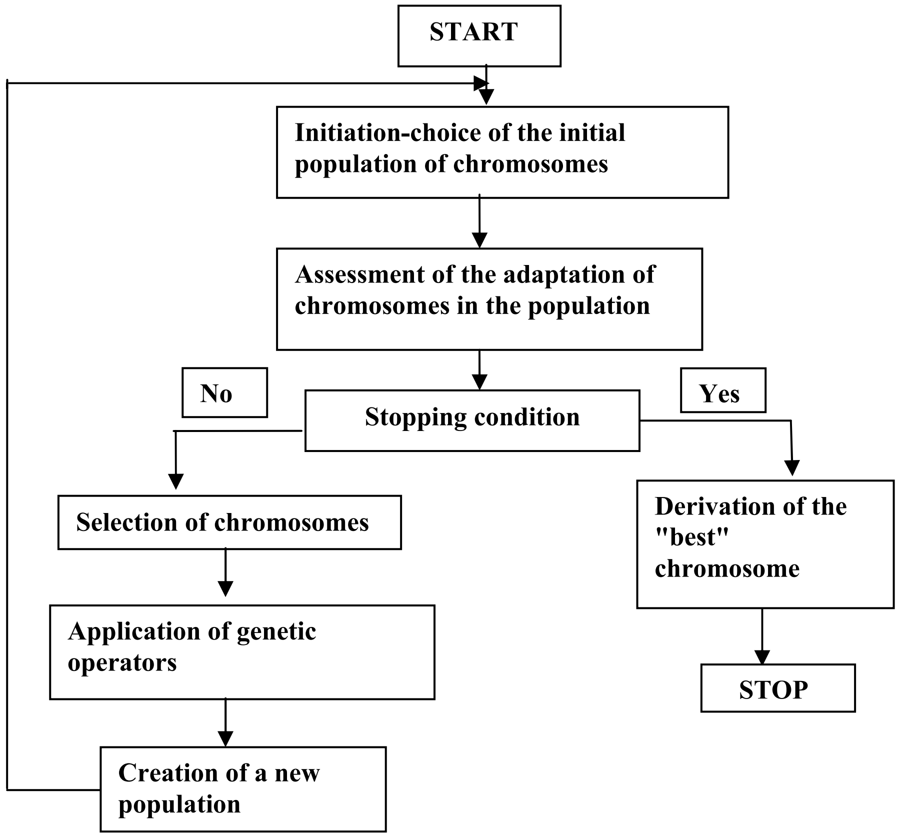

Fig. 1. Bloc diagram of the genetic algorithm.

Genetic algorithms require encoding a set of parameters of the optimisation task and use three basic operations: selection, crossbreeding and mutation. Encoding particular parameters can be achieved in a binary system in the form of a natural binary code or the Gray code. In order to shorten the length of code sequences (chromosomes) logarithmic encoding is used. The length of sequences depends on the required accuracy of the solution of the task and on the width of the range of independent variables of the function being optimised. In search of the minimum (maximum) of the function of several variables $f(x_1, x_2, ..., x_n)$ for $x_i \in [a_i, b_i] \subset R$; i = 1, 2,…, n, the length of the

binary sequence necessary to encode numbers from the range $[a_i, b_i]$ with the resolution $r = 10^{-q}$ is described by the inequality

$$(b_i - a_i) \cdot 10^{q} \leq 2^{m_i} - 1 \qquad \textbf{(1)}$$

The smallest natural number m_i satisfying the inequality (1) determines the length of the binary sequence. In genetic algorithms an important notion is the adaptation function, which in problems of optimisation is the function being optimised called the objective function. The adaptation function is used for the assessment of the adaptation of a particular individual in order to create a new population of individuals as a set of potential solutions to the task of optimisation. The basic genetic algorithm sufficient for solving a number of practical problems consists of a sequence of the following operations: selection, crossbreeding, mutation. After a particular population of chromosomes which are statistically independent is initiated they are selected according to the rule of exclusivity in terms of the adaptation function (objective function). At this stage of the algorithm the best adopted individuals remain, and the other are rejected. From among a number of selection methods the most popular method is the so called roulette wheel selection, which consists in identifying a chromosome belonging to the slot of the roulette wheel which came up, corresponding to the value proportional to the value of the function of adaptation of that chromosome. For this reason the probability of selecting a particular chromosome increases with an increase in the value of the function of its adaptation. The selection process ends when a mating pool is created with a size initiated at the beginning of the algorithm.

Apart from the roulette wheel selection there is the tournament selection, which is used as the basic selection method in the software FlexTool [1] for solving tasks of minimization, maximization and multi-criteria optimisation, i.e. simultaneous optimisation of a number of functions. There are two methods of choosing individuals corresponding to the best adaptation function: deterministic tournament selection and stochastic tournament selection. The first selection method has probability equal 1, and the other has probability less than one. The scheme of tournament selection with a division into subgroups consisting of two individuals (random size) is presented in fig. 2 [6].

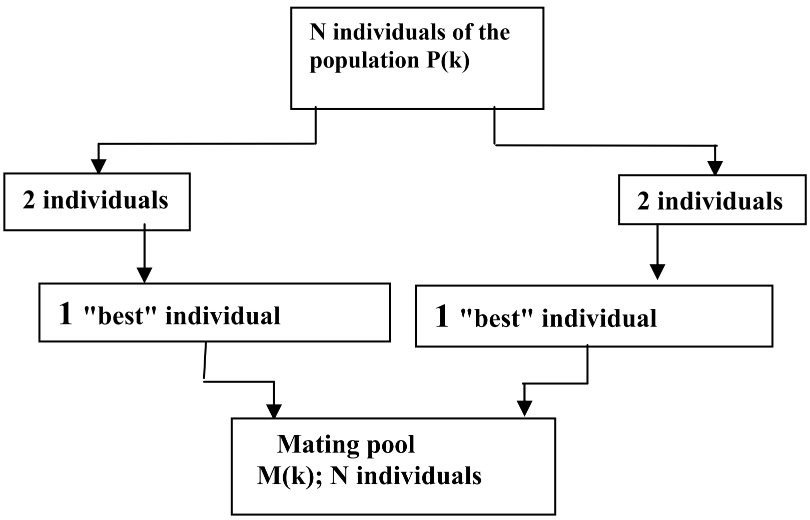

Fig. 2. Tournament selection for the size of subgroups equal 2.

Ranking selection consists in sorting out individuals in terms of the value of the adaptation function. Ranking selection can be used for maximizing as well as minimizing functions, at the same time a premature convergence of the algorithm is avoided. After the mating pool is filled up, i.e. after the selection process is finished, an operation of crossbreeding follows, which consists of two stages. The first stage of crossbreeding consists in pairing encoding sequences at random, according to the adopted probability of crossbreeding. In the second stage, for each pair a position for a gene in the chromosome, which is a crossbreeding point, is drawn by lot. The crossbreeding point l_k is a natural number less than the length of the chromosome *L*, dependent on the conditions of the task.

As an example we will consider two chromosomes $\mathrm{ch}_1 = [10110 | 01100]$ and $\mathrm{ch}_2 = [10010 | 10011]$ consisting of 10 genes ($L = 10$), for which an integer from the range [1, 9] equal 5 ($l_k = 5$) has been drawn by lot. As a result of the operation of crossbreeding (figure below), we obtain two new chromosomes (pair of descendants).

Pair of parents		**Pair of descendants**
[10110 | **01100**]		[10110 | **10011**]
	crossbreeding	
[10010 | **10011**]		[10010 | **01100**]

At this point it is necessary to mention that according to the Hardy-Weinberg rule based on the theory of probability, in large populations the frequency of existence of genes remains steady, i.e. in the state of genetic balance. On the other hand, according to the Kolmogorov-Smirnov rule, the smaller is the random sample the greater is the probability of deviation from real distribution [4].

The procedure of mutation is similar, but according to the probability of mutation, the value of only one gene in a particular chromosome changes to the opposite one. If, for example, in the chromosome [1000110111] the gene in position 4 undergoes mutation, then as a result of the mutation we obtain the chromosome [1001110111]. The operators of the mutation change values of the bit with a very small probability.

Most optimisation tasks solved by means of genetic algorithms consisted in one criterion, represented by an objective function. In tasks of searching for an optimising solution with the participation of a number of criteria it is necessary to search for a compromise, which is an optimum solution in the Pareto sense [5] as a multi-criteria optimisation. The condition of optimality in the Pareto sense is formulated as follows[2]: the vector x is less (partly) than the vector y then and only then if

$$(\forall i)(x_i \le y_i) \wedge (\exists i)(x_i < y_i) \,. \qquad (2)$$

3. EXAMPLE OF OPTIMIZING A FUNCTION

Example 1. Finding the minimum and maximum of the function (fig. 3)

$$f(x,y) = 2(1-x)^2 \exp[-x^2 - (y+1)^2] -$$
$$7(\frac{x}{3} - x^3 - y^2)\exp(-x^2 - y^2) - (\frac{1}{5}\exp(-(x+1)^2 - y^2)$$

for $x, y \in [-3, 3]$ with an accuracy of 0,01.

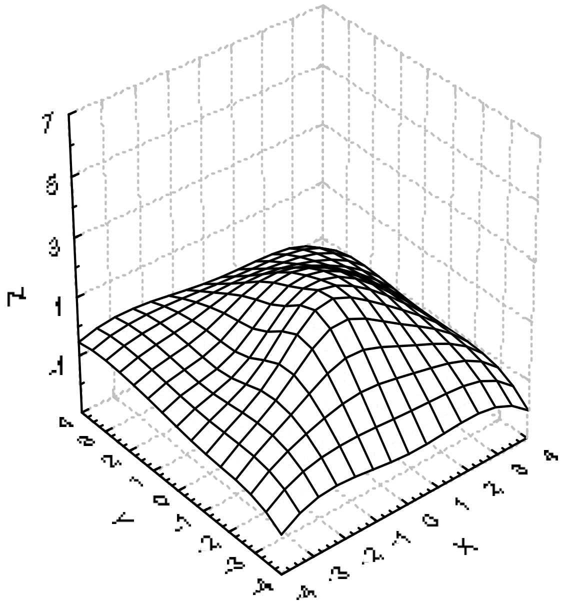

Fig. 3. Diagram of the function being optimised.

The minimum of the function has been obtained in the form of chromosomes with values of phenotypes equal -1,40 and 0,15, for which the value of the adaptation function is -1,908. In search of the maximum of the function the best solution has been obtained for the chromosomes with the values of phenotypes equal -0,39 and -0,99, and the value of the adaptation function equal 5,638.

Example 2. Finding a solution simultaneously optimising two functions:

$$f_1(x) = (x - 2)^2$$
$$f_2(x) = (x - 3)^2$$

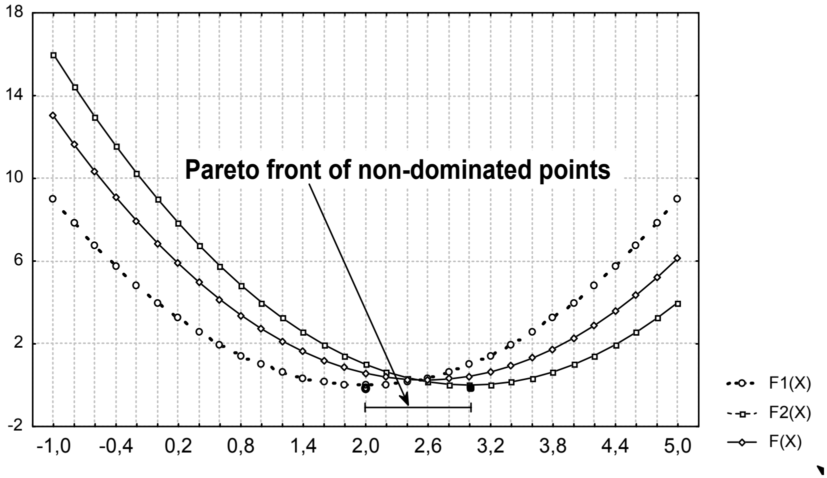

Fig. 4. Optimum solution in the Pareto sense for:
$\alpha = 0{,}4275$, $x = 2{,}57$, $f_{\min} = 0{,}245$

using the method of objective weighting

$$f(x) = \sum_{\mathrm{i=1}}^{2} w_i f_i(x)$$

where $w_i \in [0,1]$ **and** $\sum_{i=1}^{2} w_i = 1$, **for** $w_1 = \alpha$ **and** $w_2 = 1 - \alpha$ **we obtain an optimum solution in the Pareto sense (fig. 4).**

Example 3. Determining values of the parameters $\boldsymbol{\alpha}$ **and** $\boldsymbol{\beta}$ **of the function**

$$H(t) = \boldsymbol{\alpha}_1 [1 - 8/\boldsymbol{\pi}^2 \exp(-\boldsymbol{\beta} t)],$$

describing the curve of foundation settlement under a load placed suddenly.

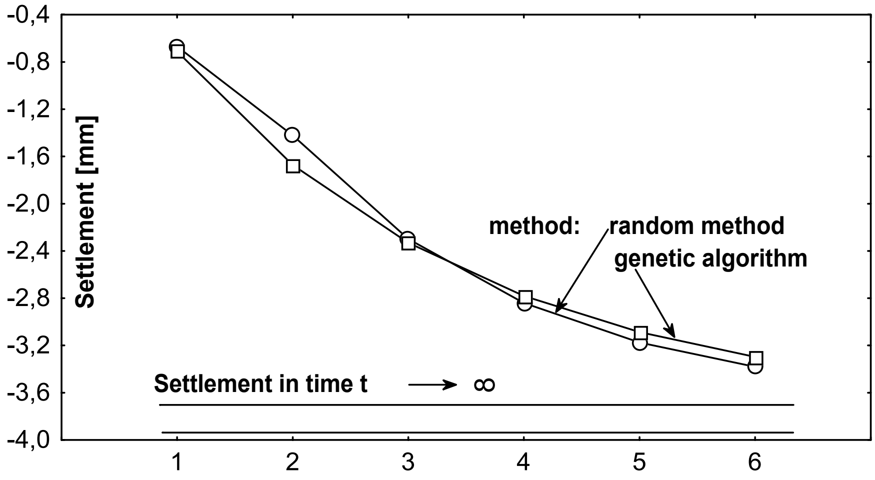

Fig. 5. Diagram of foundation settlement.

The following values of parameters have been obtained by means of the random method and the genetic algorithm:

A. Random method	**B. Genetic algorithm**
$-\alpha = 3{,}71$	$-\alpha = 3{,}94$
$-\beta = 2.416$	$-\beta = 1{,}728$

In this case the random method of searching for values of parameters proved to be a little better than the method of searching by means of the genetic algorithm because mean errors of the approximation respectively assumed the values: $m_A = 0{,}06$, $m_B = 0{,}18$. **This results from the size of the population generated, which for the random method was a few times larger than for the method of the genetic algorithm.**

Example 4. Solving the task of optimality by means of the necessary and sufficient conditions

$$\mathbf{min}\ \boldsymbol{f}(\boldsymbol{x}) = (x_1)^2 + 4(x_2)^2]$$
$$\mathbf{p.o.}\ \ g(\boldsymbol{x}) = \text{-}x_1 \text{ - } x_2 + 2 \le 0$$

Solving the problem of minimization of the function $f(x)$ with restrictions leads to the task of non linear programming. The point of the minimum of the objective function without restrictions $(\mathbf{x} = \mathbf{0})$ is not the solution of the task but the point situated on the edge of the set of acceptable points. Because of the fact that the Karlin condition is satisfied (regularity of points), the solution, if it exists, has to satisfy the Kuhn-Tucker conditions. We introduce the Lagrange function

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}) = (x_1)^2 + 4(x_2)^2 + \lambda(\text{-}x_1 \text{ - } x_2 + 2)$$

and minimize the system of the Kuhn-Tucker conditions:

$$\frac{\partial L(\boldsymbol{x}, \lambda)}{\partial x_1} = 2x_1 - \boldsymbol{\lambda} = 0,\ \frac{\partial L(\boldsymbol{x}, \lambda)}{\partial x_2} = 8x_2 - \boldsymbol{\lambda} = 0,\ \boldsymbol{\lambda} g(\boldsymbol{x}) = \lambda(\text{-}x_1 \text{ - } x_2 + 2 = 0,\ \boldsymbol{\lambda} \ge \boldsymbol{0}$$

including the restriction $\text{-}\,x_1 \text{ - } x_2 + 2 \le 0$. The minimum of the function $\widehat{f} = 3{,}23$ has been obtained in the form of chromosomes with the values 1,61 and 0,40 $(\lambda = 3{,}18)$. The objective function reaches the minimum at the point $\widehat{\boldsymbol{x}} = (1{,}60,\ 0{,}40)$, situated on the straight line $x_1 + x_2 = 2$.

4. CONCLUSIONS

Genetic algorithms are procedures of searching and optimising which are based on imitating natural processes of natural selection and heredity. An increasing number of applications for these algorithms results from the lack of restrictions on the search space (continuity, existence of derivatives) while searching for optimum or almost optimum solutions at a minimum cost. Due to these characteristics genetic algorithms can be used in programming computers, artificial intelligence, optimisation, and cooperating with neural networks and fuzzy systems constituting hybrid systems.

REFERENCES

FlexTool (GA) M2.1, Flexible Intelligence Group, L.LC., Tuscaloosa, AL 35486-1477, USA.

Goldberg D.E., Algorytmy genetyczne i ich zastosowania. Wydawnictwa Naukowo-Techniczne, Warszawa 1998.

Holland J.H., Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press, 1975.

Łęski J., Systemy neuronowo-rozmyte, WNT, Warszawa 2008.

Michalewicz Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer -Verlag, 1992.

Rutkowska D., Piliński M., Rutkowsli L., Sieci neuronowe, algorytmy Genetyczne i systemy rozmyte, Wydawnictwo Naukowe PWN, Warszawa-Łódź, 1999.