Abstract – This article aims to give a general view about Genetic Algorithms, one the most fashionable technology of the last years. Genetic Algorithms can be seen as a method of automatic answer to problems that are solved by the combination of solutions.

Genetic Algorithms are based on Darwinian principles of Natural Origin of Species. All the species on Earth are the result of an evolution throughout millions years, each species evolved in order to adapt better to the environment, taking advantage of its innate features to survive.

On this paper it is described the general working of a Genetic Algorithm, terminology, concepts of evaluation and fitness, individuals selection of population of solutions, and of crossover and mutation operators. At last, two simple examples will be given, allowing to form some concepts of Genetic Algorithms.

Key-Words: - Genetic Algorithms, chromosomes, selection, crossover, mutation, inversion and fitness.

1 Introduction
Genetic Algorithms (GA) are based on population’s process of natural evolution [4]. Firstly, it is chosen a set of individuals, a set of possible solutions. On each generation (iteration), the best members of the population generate new members (sons) from a set of operators. At the end of each iteration, the worst sons are removed, forming the population for the next generation with the remaining ones. If a GA is developed correctly, the population will converge on a good solution, perhaps on the best solution.

GA are not the only technique based on an analogy of nature. For instance, the Neural Networks are based on the behaviour of brain’s neurones.

GA advantages over the other tools of conventional search:
- Being used as a searching method, with a wide range of possible solutions;
- A wide spectrum of use;
- Simplicity of use;
- Suitable for using on dynamic spaces;
- A strong possibility of finding a near-perfect solution.

Nowadays, GAs are applied to several areas, such as: attribution of tasks, optimising and learning of machines, planning paths on robots manipulators and mobile robots, scheduling, biomedical engineering, artistic creation, and so on.

The pseudocode of a basic genetic algorithm is shown on figure 1. We can see that genetic algorithms start with a population $P$ of $n$ individuals, where each individual codifies a solution to the problem. The evaluation of each individual’s performance is based on a function of fitness evaluation. The best ones will tend to be the progenitors of the following generation, allowing to transmit their features to the next generations [10].
Algorithm GA

{ t := 0; //counter
Starts_population(P,t); //starts a population on n individuals
Evaluation (P,t); //evaluates individuals fitness
Repeat until (t = d) //tests criteria (duration, fitness, and so on.)
{ t := t + 1; //increases the counter of generations
Selection_of_parents (P,t); //selects the couples for crossover
Recombination (P, t); //accomplishes selected couples crossover
Mutation (P, t); //disturbs the group generated by crossover
Evaluation (P, t); //evaluates new fitness
Survive (P, t) //selects survivors
}
}

\( t \) – actual generation;
\( d \) – criterion to finish the algorithm;
\( P \) – population

Fig. 1 – Basic pseudocode of a Genetic Algorithm.

Stopping criteria:
- When a solution with a certain value of fitness is reached (for the type of problems demanding a cheaper solution, without finding the best solution);
- When a certain time limit (for problems demanding solutions in time);
- When a certain number of generations is reached.

When a GA should be used? There isn’t a rigorous consensus on the Genetic Algorithms application, but many researchers agree on some points. GA should be used when:
- The researched space is wide;
- The researched space is not plain or unimodal;
- The researched space is not well understood;
- The problem doesn’t require the best solution.

2 Genetic Operators

Genetic operators allow to manipulate chromosomes and are responsible for the evolution of quality of solutions during the performance of GA. From the point of view of mathematics, operators function is to create new searching points in the space of solution, based on the actual population’s members.

There are countless genetic operators, but the most common are [1], [2], [5] and [7]:
- Selection;
- Crossover;
- Mutation;
- Inversion;
- Elitism.

2.1 Selection

Selection means to choose, among the individuals that belong to a population, those who will be used to create descendants for the next generation and how many descendants each one can generate. If brief, selection is the choice of parents chromosomes. It is considered that the chromosomes that belong to a population are filtered in order to generate better individuals according to a certain criterion.

2.1.1 Types of selection

The selection of parents chromosomes can be done in many ways. The following mechanisms of selection can be used:
- Fitness-Proportionate Selection;
- Boltzmann Selection;
- Sigma Scaling;
- Rank Selection.

Roulette-Wheel Selection Method [2] is the most used method in selection through Fitness-Proportionate Selection. With this method the probability of a chromosome being chosen is directly proportional to the value of its merit.

Fitness is an evaluation function that evaluates the chromosome’s quality and the solution that it represents.

Roulette-Wheel Selection Method takes to a premature convergence, because individuals with higher fitness get a strong probability of being chosen.

Boltzmann Selection [2] and [7], Sigma Scaling [2], Rank Selection [7] are operators that avoid premature convergence.

2.1.2 Substitution of individuals in a population

The substitution of individuals in a population can be divided into two main types: traditional substitution and substitution in a stationary state.

Traditional substitution is the substitution of all the chromosomes of a population for their descendants in order to create a new population.

The substitution in a stationary state avoids the exclusion of many good individuals that don’t reproduce themselves. Some parents chromosomes move to the next generation.

2.2 Crossover

Crossover is an operator of “coupling”, that allows to produce new chromosomes (sons) through the exchange of partial information among pairs of chromosomes (parents). Types of crossover that can be used [7] and [11]:
- Single-point crossover;
- Two-point crossover;
- Uniform crossover;
- Multi-parent crossover.

In single-point crossover the choice of a crossover point is random. Each pair of chromosomes chosen as parents generates two descendants through the exchange of its final parts, after the crossover point. To exemplify the single-point crossover, we will consider the chromosomes S1 and S2. The determination of crossover point was randomly determined and is represented by the symbol [.

\[
S1 = 1 - 0 - 1 - [1 - 0 - 0 - 1 - 0 - 1 \\
S2 = 0 - 1 - 1 - [0 - 1 - 1 - 1 - 0 - 1
\]

The descendants, S1' and S2', are:

\[
S1' = 1 - 0 - 1 - [0 - 1 - 1 - 1 - 1 - 0 \\
S2' = 0 - 1 - 1 - [1 - 0 - 0 - 1 - 0 - 1
\]

Two points crossover is the choice of two cutting points. Using the chromosomes S1 and S2, and cutting points 3 and 7 (random choice) we have sons chromosomes, S1' and S2':

\[
S1 = 1 - 0 - [1 - 1 - 0 - 0] - 1 - 0 - 1 \\
S2 = 0 - 1 - [1 - 0 - 1 - 1] - 1 - 0 - 1 \\
S1' = 1 - 0 - [1 - 0 - 1 - 1 - 1 - 0 \\
S2' = 0 - 1 - 1 - [1 - 0 - 0 - 1 - 0 - 1
\]

In uniform crossover, it is created, in a random way, a mask that indicates the genes of the two chromosomes that will be exchanged. Using chromosomes S1 and S2 and mask M we have sons chromosomes, S1' e S2':

\[
S1 = 1 - 0 - [1 - 1 - 0 - 0] - 1 - 0 - 1 \\
S2 = 0 - 1 - [1 - 0 - 1 - 1] - 1 - 0 - 1 \\
M = 0 - 1 - 0 - 1 - 0 - 1 - 0 - 1 - 0 - 1 - 0 \\
S1' = 1 - 1 - 1 - 0 - 0 - 1 - 1 - 1 - 0 - 1 \\
S2' = 0 - 1 - 1 - [1 - 0 - 0 - 0] - 1 - 1 - 0
\]

In Multi-parent crossover [8] we use more than two parents chromosomes to create sons chromosomes. In the Travelling Salesman Problem the codification of chromosome is made through integers that represent the towns. The previously presented types of crossover are not suitable due to the repetitions and or lacks of towns resulting from the operation.

Some crossover operators were created to avoid the repetition of genes in chromosome, for instance:

- Order Crossover [1]
- Partially Mapped Crossover [2]
- Cycle Crossover [2]
- Linear Order Crossover [10]

The aim of mutation is to assure the genetic diversity of a population of chromosomes. The mutation consists of random changes of one or more genes in chromosomes. The mutation rate is usually very low, like in natural populations. Types of mutation:

- Simple Mutation [2]
- Position-Based Mutation [1]
- Order-Based Mutation [1]
- Scramble Mutation [1]

Simple mutation is the random choice of a position in a binary chromosome, as we exemplify:

\[
S1 = 1 - 1 - [0] - 1 - 0 - 0 - 1 - 0 - 1 \\
S1' = 1 - 1 - [1] - 1 - 0 - 0 - 1 - 0 - 1
\]

In Position-Based Mutation two positions are chosen in a random way and the content of the second position is placed before the first position, as we exemplify:

\[
S1 = 8-1-[3]-8-5-4-2-[6]-9 \\
M = 0 - 1 - 0 - 1 - 0 - 1 - 0 - 1 - 0 - 1 - 0 \\
S1' = 8-1-[6]-3-8-5-4-2-9
\]

In Order-Based Mutation two positions are chosen in a random way and their contents are exchanged, as we exemplify:

\[
S1 = 8-1-[3]-8-5-4-2-[6]-9 \\
S1' = 8-1-[6]-3-8-5-4-2-9
\]

The Scramble Mutation is used when the neighbourhood of positions is important. In this type of mutation we choose a sub list in a random way, and the order of contents is exchanged, as we exemplify:

\[
S1 = 8-1-[3-8-5-4]-2-6-9 \\
S1' = 8-1-[8-4-3-5]-2-6-9
\]

2.4 Inversion

The previously presented crossover techniques don’t take into account an order among genes of a chromosome. However, in problems such as Project Scheduling Problem, due to precedence relation among activities, these genetic operators generate very often impracticable descendants. In these conditions the reordering operators, particularly Inversion Operator, are very important. Inversion operator acts only on a chromosome inverting the order of bits between two points chosen in a random way:

\[
S1 = 1-0-1-0-[1-1-0-1]-1-0-0-1
\]

The new configuration,
2.5 Elitism
To assure that the best member of a population is not eliminated, there is the elitism operator [1] that puts the best chromosome or the best ones of a generation on the next generation.

3 Decisions to take into account to implement a GA
The success of a GA depends mainly on four decisions described on the following questions:

Which codification should be used?
According to Mitchell [5], there isn’t yet a rigorous conclusion about the codification that should be used on chromosomes. On most applications it is used alphabets with few characters, real numbers, fractional numbers or even binary codification.

How to do a selection?
A selection must be balanced against crossover and mutation with relation to diversity.

Which operators should be used?
This decision depends on the problem, as well as the strategy of codification of chromosome and the type of relationship that each gene keeps with the others.

How to fit parameters?
There are not conclusive results about the establishment of parameters such as the size of population, the rate of crossover and the rate of mutation. An interesting fact is that parameters interact in a non linear way, so that they can’t be optimised separately. Some authors [3], [7] and [8] point out the values presented on the table 1 as being the best sizes of population, rates of crossover and rates of mutation.

<table>
<thead>
<tr>
<th></th>
<th>Size of population</th>
<th>Rate of crossover</th>
<th>Rate of mutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reves (1993) [7]</td>
<td>50 - 100</td>
<td>0.6</td>
<td>0.001</td>
</tr>
<tr>
<td>Grefenstette (1986) [3]</td>
<td>30</td>
<td>0.95</td>
<td>0.01</td>
</tr>
<tr>
<td>Schaffer, Caruana and Eshelman (1989) [8]</td>
<td>20 - 30</td>
<td>0.75 - 0.95</td>
<td>0.005 - 0.01</td>
</tr>
</tbody>
</table>

Table 1 – Size of population, rate of crossover and mutation.

4 Examples of applications of GAs
4.1 Example 1
To summarize all the aspects referred up to now, we will present a simple example for optimise a mathematical function and to see as AG develops along a group of generations.

We want to find the maximum value of the mathematical function \( f(x_1, x_2) \) defined for:

\[
f(x_1, x_2) = 21.5 + x_1 \cdot \text{sen}(4\pi x_1 + x_2 \cdot \text{sen}(20\pi x_2))
\]

with \(-3.0 \leq x_1 \leq 12.1\) e \(4.1 \leq x_2 \leq 5.8\)

The graphical representation of the function can be observed in the figure 2.

Fig. 2 – Graphical representation of the function.

4.1.1 Representation
The function \( f \) is dependent of two variables. Opting for the binary representation of the chromosomes, the chain of bits should code the information regarding the two variables \((x_1, x_2)\).

In order to obtain a precision of four decimal digits in each variable, they are necessary:
- 18 bits to code the variable \( x_1 \), once this possesses a size domain the same to \( 15,1 \) \((2^{17} \leq 151000 \leq 2^{18})\);
- 15 bits to code the variable \( x_2 \), once this possesses a size domain the same to \( 1,7 \) \((2^{14} \leq 17000 \leq 2^{15})\).

They will be necessary 33 bits to code a solution for the function \( f(x_1, x_2) \).

4.1.2 Function of evaluation
Consider the chromosomes represented for:

\[\begin{align*}
11101001011100000100101110000000
\end{align*}\]

The first 18 bits are used to code the variable \( x_1 \). Making the conversion of binary for decimal, we have:

\[x_1' = \text{decimal}(11101001011100000100101110000000) = 250552\]
The real number corresponding to $x$ is:

$$x_1 = \frac{\text{Limit} \_ \text{Left} \_ \text{Dom} + x' \times \frac{\text{Size} \_ \text{Dom}}{2^N \_ \text{Bits}} - 1}{2^8} = 11.4323$$

Using the remaining 15 bits of the chromosomes, we have:

$$x_2 = \frac{1.7}{2^{15}} = 4.3490$$

The used chromosomes correspond to the point of coordinates (11.4323, 4.3490) and the value for the evaluation function is $f(11.4323, 4.3490) = 13.1733$.

### 4.1.3 Simulation of the algorithm

To exemplify the operation of the operators selection, crossover and mutation, we will simulate the operation of GA for a population of 20 individuals. We will use the selection method for roulette with elitism in that the two better individuals of each generation will pass to the next generation. The single-point crossover and the mutation will be applied with probabilities of 65% and 1%, respectively.

The initial population of 20 chromosomes, obtained in a random way, is:

- $i_1 = 010011000101111101100100101001111$
- $i_2 = 01101101111110101000101111111$
- $i_3 = 0011000011111001001101100101010$
- $i_4 = 010111000111110010111010001010$
- $i_5 = 1011110100111101010011011011001$
- $i_6 = 111010100111110111100110000110$
- $i_7 = 1100010111111100111001111111111$
- $i_8 = 1000100000101010011110100110101$
- $i_9 = 11111011111101110100001110110001$
- $i_{10} = 00100010110110101001100110110110$
- $i_{11} = 01010111011010001010110010110001$
- $i_{12} = 01111000111111111101101011001000$
- $i_{13} = 01110010111111111111011111001110$
- $i_{14} = 01001000010010110111110110001011$
- $i_{15} = 11111110110100001110101010011001$
- $i_{16} = 01101111110001100101011000110000$
- $i_{17} = 00000010000000011010011011100101$
- $i_{18} = 11101011001010001010110010000100$
- $i_{19} = 01111001110111111111111111111000$
- $i_{20} = 10100000101011010001000000000010$

The function used to evaluate the individuals is the own mathematical function. Each individual will have a value that indicates its quality ($f(x)$), a probability of being chosen ($p(x)$) and it will be with a segment of the size roulette ($r(x)$), as shown in table 2.

The value of each individual's individual probability is calculated dividing the value of its merit for the sum of all the individuals' merit - $F = \sum_{j=1}^{20} f(i_j) = 437.53$.

<table>
<thead>
<tr>
<th>Individual</th>
<th>$f(x)$</th>
<th>$p(x)$</th>
<th>$r(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_1$</td>
<td>26.849</td>
<td>0.061</td>
<td>0.061</td>
</tr>
<tr>
<td>$i_2$</td>
<td>24.623</td>
<td>0.056</td>
<td>0.118</td>
</tr>
<tr>
<td>$i_3$</td>
<td>24.750</td>
<td>0.057</td>
<td>0.174</td>
</tr>
<tr>
<td>$i_4$</td>
<td>24.347</td>
<td>0.056</td>
<td>0.230</td>
</tr>
<tr>
<td>$i_5$</td>
<td>25.677</td>
<td>0.059</td>
<td>0.289</td>
</tr>
<tr>
<td>$i_6$</td>
<td>21.028</td>
<td>0.048</td>
<td>0.337</td>
</tr>
<tr>
<td>$i_7$</td>
<td>14.702</td>
<td>0.034</td>
<td>0.370</td>
</tr>
<tr>
<td>$i_8$</td>
<td>26.613</td>
<td>0.061</td>
<td>0.431</td>
</tr>
<tr>
<td>$i_9$</td>
<td>16.521</td>
<td>0.045</td>
<td>0.476</td>
</tr>
<tr>
<td>$i_{10}$</td>
<td>10.491</td>
<td>0.024</td>
<td>0.500</td>
</tr>
<tr>
<td>$i_{11}$</td>
<td>21.428</td>
<td>0.049</td>
<td>0.549</td>
</tr>
<tr>
<td>$i_{12}$</td>
<td>17.122</td>
<td>0.039</td>
<td>0.588</td>
</tr>
<tr>
<td>$i_{13}$</td>
<td>30.268</td>
<td>0.069</td>
<td>0.657</td>
</tr>
<tr>
<td>$i_{14}$</td>
<td>28.323</td>
<td>0.065</td>
<td>0.722</td>
</tr>
<tr>
<td>$i_{15}$</td>
<td>13.478</td>
<td>0.031</td>
<td>0.752</td>
</tr>
<tr>
<td>$i_{16}$</td>
<td>25.725</td>
<td>0.059</td>
<td>0.811</td>
</tr>
<tr>
<td>$i_{17}$</td>
<td>23.548</td>
<td>0.054</td>
<td>0.865</td>
</tr>
<tr>
<td>$i_{18}$</td>
<td>16.200</td>
<td>0.037</td>
<td>0.902</td>
</tr>
<tr>
<td>$i_{19}$</td>
<td>28.801</td>
<td>0.066</td>
<td>0.968</td>
</tr>
<tr>
<td>$i_{20}$</td>
<td>14.036</td>
<td>0.032</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Table 2 - Each individual's quality ($f(x)$), probability of being chosen ($p(x)$) and segment of the size roulette ($r(x)$).

As for the following generation the two individuals of better quality will transit, we will to choose 9 pairs for they generate 18 new individuals. The 9 pairs went chosen through to roulette, rotating a generator of random numbers 18 times. The random values and the individuals chosen for generate descendants are shown in the table 3.

<table>
<thead>
<tr>
<th>Random values [0, 1]</th>
<th>Chosen individuals (pairs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.79; 0.84</td>
<td>$i_{16}$, $i_{17}$</td>
</tr>
<tr>
<td>0.84; 0.1</td>
<td>$i_{17}$, $i_2$</td>
</tr>
<tr>
<td>0.91; 0.24</td>
<td>$i_{10}$, $i_5$</td>
</tr>
<tr>
<td>0.5; 0.86</td>
<td>$i_{11}$, $i_{17}$</td>
</tr>
<tr>
<td>0.03; 0.94</td>
<td>$i_1$, $i_{19}$</td>
</tr>
<tr>
<td>0.67; 0.52</td>
<td>$i_{14}$, $i_{11}$</td>
</tr>
<tr>
<td>0.47; 0.92</td>
<td>$i_6$, $i_{19}$</td>
</tr>
<tr>
<td>0.01; 0.39</td>
<td>$i_1$, $i_8$</td>
</tr>
<tr>
<td>0.64; 0.55</td>
<td>$i_{13}$, $i_2$</td>
</tr>
</tbody>
</table>

Table 3 – Chosen individuals.
The crossing operator was applied with a probability of 65%, meaning that just some of the pairs chosen for reproduction will be used.

The generated random values and the respective cutting points are shown in the table 4. We can observe that random values greater than 0.65 indicate that crossover doesn’t happen.

<table>
<thead>
<tr>
<th>Random values [0, 1]</th>
<th>Cutting point</th>
<th>Progenitors (sons)</th>
<th>Descendants</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.18</td>
<td>23</td>
<td>i_{16}, i_{17}</td>
<td>i'<em>{3}, i'</em>{4}</td>
</tr>
<tr>
<td>0.76</td>
<td>---</td>
<td>i_{17}, i_{2}</td>
<td>i'<em>{5}, i'</em>{6}</td>
</tr>
<tr>
<td>0.34</td>
<td>24</td>
<td>i_{19}, i_{35}</td>
<td>i'<em>{7}, i'</em>{8}</td>
</tr>
<tr>
<td>0.07</td>
<td>9</td>
<td>i_{11}, i_{17}</td>
<td>i'<em>{9}, i'</em>{10}</td>
</tr>
<tr>
<td>0.9</td>
<td>---</td>
<td>i_{1}, i_{19}</td>
<td>i'<em>{11}, i'</em>{12}</td>
</tr>
<tr>
<td>0.45</td>
<td>17</td>
<td>i_{16}, i_{11}</td>
<td>i'<em>{13}, i'</em>{14}</td>
</tr>
<tr>
<td>0.07</td>
<td>16</td>
<td>i_{3}, i_{19}</td>
<td>i'<em>{15}, i'</em>{16}</td>
</tr>
<tr>
<td>0.91</td>
<td>---</td>
<td>i_{11}, i_{8}</td>
<td>i'<em>{17}, i'</em>{18}</td>
</tr>
<tr>
<td>0.42</td>
<td>31</td>
<td>i_{13}, i_{12}</td>
<td>i'<em>{19}, i'</em>{20}</td>
</tr>
</tbody>
</table>

Table 4 – Cutting point, progenitors and descendants.

The first two individuals of the descendants were chosen for elitism, that is, they are the two better individuals of the previous population.

i'_{1} = i_{13} = 01111000111100011111101101010011100
i'_{2} = i_{19} = 0111100111100111111011110110001101

The descendants i'_{3} and i'_{4} are generated by the individuals' i_{16} and i_{17} crossover, using the cutting point 23.

i'_{3} = 001011110010000010110001|0011000001
i'_{4} = 00000010000000000011000010111100101

The crossover operator was not used in the progenitors i_{17} and i_{2}, being these two individuals' descending copies (i'_{5}=i_{17} e i'_{6}=i_{2}).

Continuing to apply the crossover operator, we would obtain the following population:

i'_{1} = 0111100011110001111110101010011100
i'_{2} = 0111100111001110111011101010001000
i'_{3} = 0010111100100010110001111110101001
i'_{4} = 00000010000000000110000110111100001
i'_{5} = 000000100000000001110001111110001
i'_{6} = 01111001011011011101010111001110001
i'_{7} = 000000100000000011100011111110001
i'_{8} = 00000010000011110110110100011100110
i'_{9} = 01001110010111111101110101000111110
i'_{10} = 0111100011110001111110101010011100
i'_{11} = 10100010110011011101010110110011101
i'_{12} = 00010100111101110011101111110001111
i'_{13} = 10100010110011011101010110110011101
i'_{14} = 0111100011110001111110101010011100
i'_{15} = 0100111000101111110111010101100111110
i'_{16} = 110001011111111011111001111110111110
i'_{17} = 0111100011110001111110101010011100
i'_{18} = 1010011011010100010001100110001100
i'_{19} = 0111100011110001111110101010011100
i'_{20} = 00101111010000010110001111110000001

Suppose that the mutations are located in the following points:

Son i'_{3} – gene 14
Son i'_{6} – gene 16
Son i'_{12} – gene 7

Here is the result of a new population that will be evaluated again. The genes that suffered mutation are in bold.

<table>
<thead>
<tr>
<th>Generation</th>
<th>The best individual’s quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30.2680</td>
</tr>
<tr>
<td>2</td>
<td>30.2680</td>
</tr>
<tr>
<td>3</td>
<td>33.4468</td>
</tr>
<tr>
<td>4</td>
<td>33.4468</td>
</tr>
<tr>
<td>5</td>
<td>33.4468</td>
</tr>
<tr>
<td>6</td>
<td>33.4468</td>
</tr>
<tr>
<td>7</td>
<td>33.8733</td>
</tr>
<tr>
<td>8</td>
<td>34.1697</td>
</tr>
<tr>
<td>9</td>
<td>34.1697</td>
</tr>
</tbody>
</table>

At this time, a cycle of GA is ended. The process is now repeated along several generations. Shown in table 5 are the best individual's values in the first 11 generations and in the generation number 100. As one can see GA is going developing for higher values, as it is wanted.
Table 5 - The best individual’s values in the first 11 generations and in the generation number 100.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>34.1697</td>
</tr>
<tr>
<td>11</td>
<td>35.4115</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>37.1530</td>
</tr>
</tbody>
</table>

### 4.2 Example 2 – Scheduling of tasks in a milling CNC machine

Let us consider the scheduling of 9 tasks \{1, 2, 3, 4, 5, 6, 7, 8, 9\} in a milling CNC machine, where the odd tasks are executed by a tool \( t_i \) and the even tasks are executed with a tool \( t_p \).

The code will be made by a sequence of 9 digits and the evaluation function will try to minimize the number of tool changes. The elements of the population with smaller number of tool changes will be the chosen ones for the next generation.

We will have to use the order crossover, in order to not having repeated and absent tasks.

We will consider the table 6 that gives us the elements of the population and the value of the evaluation function \( f(x) \).

<table>
<thead>
<tr>
<th>Element number</th>
<th>Generation 1</th>
<th>( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>394165287</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>821397465</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>136974285</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>754381296</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 6 - Elements of the population and the value of the evaluation function.

Applying the order crossover and 2 court points (between the 2\(^{nd}\) and 3\(^{rd}\) genes and between the 6\(^{th}\) and 7\(^{th}\) genes) and crossing the element 1 with the 2 and the element 3 with the 4, we obtain:

<table>
<thead>
<tr>
<th>Element number</th>
<th>Generation 1</th>
<th>( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39-4165-287</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>82-1397-465</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>13-6974-285</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>75-4381-296</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element number</th>
<th>Generation 2</th>
<th>( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>821397465</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>136974285</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>651397284</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>816974253</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 7 - Generation number 1 after the application of the operator order crossover.

After the elitism, we arrived to the generation 2, shown in table 8.

<table>
<thead>
<tr>
<th>Element number</th>
<th>Generation 2</th>
<th>( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>821397465</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>136974285</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>651397284</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>816974253</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 8 – Generation number 2.

**Comments:**
- It is possible to obtain a better element than the one of the previous generation, besides the population to have better evaluation functions.
- The best solution, for this case, would be that in that all the equal tasks were accomplished and later all the odd tasks or vice-versa, with cost of 1.

### 5 Conclusion

A Genetic Algorithm works with base in the generation and in the continuous evolution of chromosomes populations. Each chromosome structure codes a solution of the problem and it represents a point in the search space, therefore, its aptitude is related to the value of the function objective.

GA constitute a new instrument of research, capable of being used in several areas of the human activities. Their main advantage resides in the genetic diversity: to each step of the search process, a group of candidates is considered and involved in the new candidates’ creation.

A good result to be obtained by the Genetic Algorithms will depend on the code method, of the types of used genetic operators, of the fittings of the parameters, and so on. A GA in its basic form can be insufficient for a certain type of problem. For example the code bit-string can be inadequate; the selection through the proportionality of the fitness can drive to the premature convergence; the crossing and mutation operators can turn the descending solutions unviable; and an inadequate adjustment of the parameters can decrease the global performance. For this reason, the decisions on each step of the elaboration of the algorithm should be analysed carefully, because there exist many possibilities to implement a GA and the relationship among the objects of decision are not generally sufficiently clear and simple.

**References:**


