

INTRODUCTION TO GENETIC ALGORITHMS PART 1

JAVIER FERNANDEZ

UPNA

MOTIVATION

- Genetic Algorithms (GA) are a part of **EVOLUTIONARY COMPUTING**.
- They are adaptive and heuristic search algorithms based on the ideas of natural selection and genetics.

HEURISTICS: Set of rules not necessarily formalized or rigorous that lead to the resolution of a problem.

- AGs make smart use of searches carried out on optimization problems.
- The GAs, although they are random, make use of the information accumulated during the past search to guide the search in the future.

WHAT DOES IT MEAN TO SOLVE A PROBLEM?

We can think that solving a problem is equivalent to finding a solution that is better than all the others.

CAN WE CAN THINK THAT SOLVING A PROBLEM IS EQUIVALENT TO FINDING A SOLUTION THAT IS BETTER THAN ALL THE OTHERS?

- In Computing and AI it can be understood as a search through the solutions space.

DEFINITION: The solution space (or possible solutions) is the set of solutions that satisfy all the required conditions.

Such solutions do not have to be optimal or reasonable. It is enough that they satisfy the conditions of the problem.

- In Mathematics and Engineering they are interpreted as an optimization process.
 - A mathematical formalization of the problem is defined in terms of appropriate functions and it is about obtaining an optimum or extreme of some kind.

THE SEARCH OR SOLUTIONS SPACE

- The set of all possible solutions to a given problem constitutes the search space or solutions.
- Each possible solution is given by a point in that space.
 - This solution is "marked" by its adjustment or adaptation as a solution to the problem under consideration.
- AGs look for the best solution from a set of points within the search space.
- In general, not all the points in the search space are known, only some of them are known.
- When using an AG, new points are generated within the search space until the solution is reached.

OPTIMIZATION

- Optimization is a process to obtain the best solution - in some sense - to a given problem.
- Any optimization problem consists of at least the following elements:

1- OBJECTIVE FUNCTION: It is the function to be optimized, that is, to maximize or minimize.

- Suppose we want to build the bumper of a car. So we are interested in optimizing (maximizing) resistance to a crash. The function that measures this resistance will be **our objective function**.
- If we are responsible for a company, we are interested in optimizing (maximizing) the benefit we obtain. The utility function that measures profit is our objective function.

2- VARIABLES OR INCOGNITES. They are a set of variables that affect the objective function.

- In the case of the bumper, some variables can be its density, its dimensions or its weight.
- In the case of the company, the availability of raw materials or the time it takes to produce a product are variables of the problem.

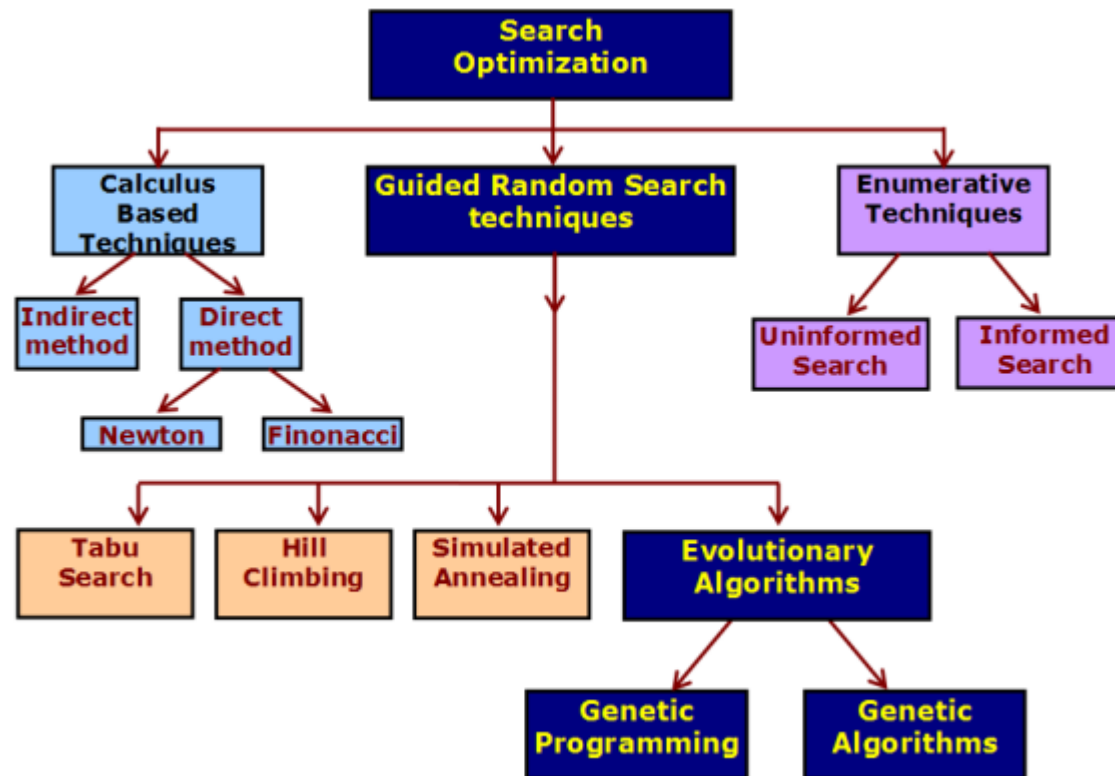
3 - SET OF RESTRICTIONS. They are a series of conditions that allow the variables to take some values but not others.

- In the case of the bumper, the size of the front of the vehicle in which it is to be installed or the maximum allowable weight are restrictions.
- In the case of the company, the costs of raw materials or the time cost must be positive.

An optimization problem is written as:

Find the values of the variables that maximize or minimize the objective function and satisfy all the restrictions.

DIFFERENT OPTIMIZATION METHODS



- The calculation techniques correspond to the algorithms that we have seen in the introduction to the subject (Newton-Raphson, bisection, etc)
- Las técnicas enumerativas recorren todo el espacio de soluciones posibles para hallar el óptimo.
- Los AG llevan a cabo una búsqueda aleatoria guiada.

WHAT ARE THE ADVANTAGES OF AGS?

- GAs are more robust than other artificial intelligence methods. That is, they are less sensitive to small variations in the initial parameters or to disturbances due to noise, incomplete information, etc.
- The search in very large solution spaces is more effective than with other Artificial Intelligence methods. In particular, they are well suited for searching very large solution spaces or for achieving optimal combinations of elements that would otherwise be temporarily very expensive, if not impossible, to obtain.

<https://www.youtube.com/watch?v=uQj5UNhCPuo>

BASIC CONCEPTS ABOUT AG

- Genetic algorithms constitute the fundamental paradigm within evolutionary computing.
- They are inspired by the theory of Darwinian evolution and the survival of the best adapted thesis.
- In this sense, GAs imitate the functioning of nature.
- Starting from a series of individuals, it determines which are the best adapted (those that best approximate the solution), crosses them, mutates them and takes into account the new generation of individuals (solutions) obtained in order to advance towards the desired optimal solution .

BIOLOGICAL BASES OF AG

- Each body has a set of rules that determines how it is constituted.
- In particular, organisms are made up of cells. Each of these cells consists of the same set of chromosomes.
- Chromosomes are DNA chains that encode all the information necessary for the structure of the organism.
- Each chromosome is made up of a set of genes, that is, building blocks of DNA. Each of these genes encodes a protein responsible for a characteristic (eg, eye color).
- The possible values (characteristics) that the same gene can take are called alleles. For example, in the case of eye color, there is one allele for blue, another for green, etc.
- Genes are located in a specific place on the chromosome, always the same. This place is called a locus.
- The set of all chromosomes (all genetic material) is called a genome.
 - The particular set of genes in a given genome is called the genotype.
 - The set of manifestations of the genotype (that is, the set of characteristics exhibited by the organism) is called the phenotype.
- In sexual reproduction, two individuals mix their genotypes in a process called combination or cross-over. The resulting individual has exactly half of its chromosomes from one of its parents and the other half from the other.
- The genes of an individual can be modified. This process is called mutation.
- The adaptation of an organism is measured by its success in the fight for survival.
- We are going to imitate the reproductive and evolutionary process of living beings.

BASIC SCHEME OF GENETIC ALGORITHMS

The reproduction process that we have described could be transcribed in pseudo-code as follows:

INITIALIZE the population randomly

FOR EACH CANDIDATE evaluate your adaptation

REPEAT

- CHOOSE** parents

- CROSS** parents

- MUTE** some of the resulting descendants.

- SELECT** the individuals of the next generation (who become the new parents)

UNTIL condition

The condition, in general, will be one of the following three:

- 1- **Exact solution found**. In this case, that solution will be returned.
- 2- **Found an approximate solution** considered good enough (in terms of some measure of error or tolerance). This solution will be returned.
- 3- **Exceeded a predetermined number of iterations**. The best of the results found up to this moment will be returned as a solution.

BASIC CONCEPTS

Chromosome: Set of genes. A chromosome contains the solution to a given problem in the form of genes.

Each chromosome encodes exactly one possible solution to the problem.

Gene: Part of a chromosome. Each gene contains part of the solution.

Example: 16598 can be a chromosome. In this case, the genes are 1,6,5,9,8.

Individual: The same as chromosome.

Population: Number of individuals or chromosomes that exist at a given time.

Fitness (Adaptation, adjustment): Value assigned to each individual based on how far or close they are from the solution.

The higher the fitness value of a chromosome, the better solution it represents.

Adaptation or fitness or adjustment function: Function that assigns each individual their fitness. It depends on each specific problem.

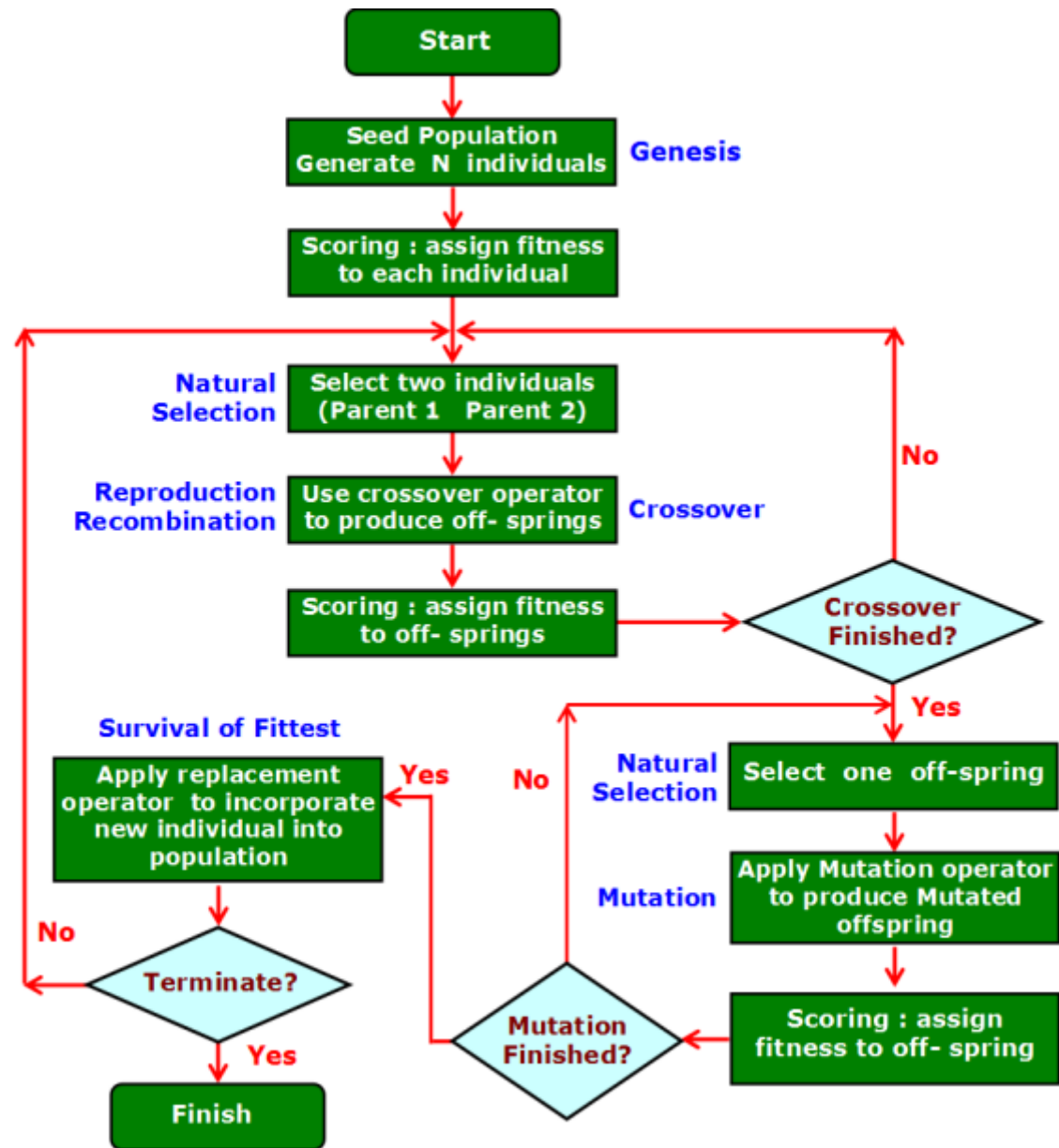
Crossing: Take two individuals and mix their chromosomes to obtain two new individuals.

Mutation: Modify a gene in an individual randomly.

Selection: Choose individuals within the population to generate the next generation.

SCHEME OF A BASIC GENETIC ALGORITHM

- 1-Randomly generate a population of n chromosomes (n possible solutions)
- 2-Evaluate the adaptation $f(x)$ of each chromosome x of the population.
- 3-Create a new population repeating the following steps as necessary:
 - a) select two parent chromosomes from the population according to their adaptation. The greater the adaptation, the greater the probability of being chosen.
 - b) Taking into account a probability of crossing, cross the parents to obtain the daughter chromosomes. If no crossover takes place, the daughter chromosomes are exactly the same as the parent chromosomes.
 - c) With a mutation probability, mutate each gene of each individual of the new generation.
 - d) Place the resulting descendant chromosomes in the new population.
- 4- Apply the algorithm to the new population.
- 5- If the stop condition is satisfied, stop and give the best solution for the current population as an answer.
- 6- Go back to step 2.



COMPONENTS OF A GENETIC ALGORITHM

1-REPRESENTATION

- We need to relate the problem in the real world to our “genetic universe”.
- Basically, we need to build an application that goes from phenotypes (the way the problem is expressed) to genotypes (the way we see solutions as chromosomes).
- A very common way to do it, especially for numerical data, is the binary representation.

Each chromosome is a sequence of zeros and ones:

Chromosome 1: 001001100110

Chromosome 2: 101111000010

Example: Consider a function of two variables x and y , with each variable represented by a 4-bit string. A chromosome would be given, for example, by:

$$(x \ y) = (1001 \ 0011)$$

NOTE: A string of n bits can represent values between 0 and $2^n - 1$.

- It is also possible to use a **non-binary** representation. For example, we can represent a chromosome by a series of alphanumeric values like this:

Chromosome 1: (ABFGDEDFRDGF)

Chromosome 2: (145789654)

- Representation through **permutations** is very useful in problems where there is an order that plays a relevant role. In this case, each gene represents position in a sequence.

Example: Let us consider the traveling salesman problem.

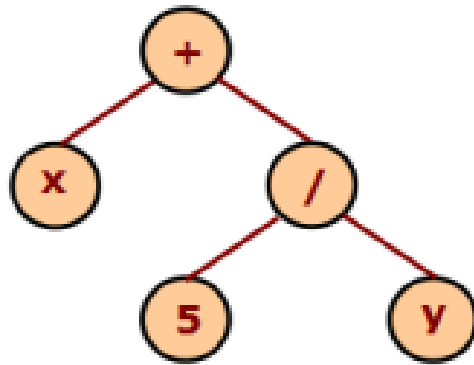
A traveller must visit a series of cities.

The distances between cities are known and we want to minimize the distance travelled.

In this case, the chromosomes represent the order in which the traveller travels through the cities.

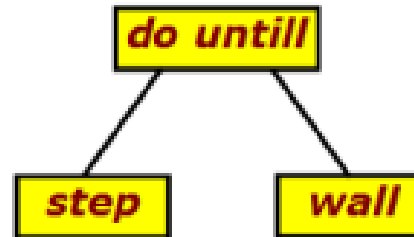
- In **tree coding**, each chromosome is a tree of objects such as functions or commands.

Chromosome A



(+ x (/ 5 y))

Chromosome B



(do until step wall)

It is very useful for evolutionary programs and is generally used with the LISP language.

2-EVALUATION FUNCTION:

- The evaluation function represents the demands that must be adjusted.

Example:

Suppose we want to maximize the function x^2 over the integers.

If we use a binary representation, the evaluation function of chromosome 10010 would be simply given by the evaluation of the corresponding genotype

$$18^2 = 324.$$

3-POPULATION:

- The population represents all the solutions that exist at any given time.
- This population can be specified only by the set of individuals that constitute it or by some relationships (distance, etc.) between them that must be specified.
- The diversity of a population refers to the number of solutions that exist in that population.
- The initial population constitutes the first approximation to the solution of the problem.
- It is usually initialized randomly, it consists of between 20 and 1,000 individuals, and this number remains fixed throughout the entire execution of the algorithm.

4-PARENT SELECTION MECHANISM:

- This mechanism, which must also be set in advance, chooses from the population the best adapted to generate the next population.
- An individual is a parent if he has been selected to generate a new population.
- In general, better quality individuals are more likely to be chosen as parents.
- However, low-quality individuals have a positive, albeit small, probability of being chosen, to avoid falling into optimal locations.

6- CROSSING:

- It is an operator that takes two individuals, combines them and returns one or two descendants.
- In AG it is the main search operator for solutions and is also probabilistic in nature.

5- MUTATION:

- A mutation operator acts on a single genotype (individual) and returns a slightly modified individual, called a mutant.
- The role of mutations in AGs is to provide “fresh blood”.
- However, there are evolutionary algorithms in which they are not used at all and others in which they are the only source of change used to reach the solution.
- The mutation is carried out by means of stochastic operators (random probabilistic) and in certain cases they allow to guarantee that the GAs reach a solution.

7- SELECTION OF SURVIVORS OR SUBSTITUTION:

- It consists of choosing the individuals that will be part of the following population.
- In general, it is deterministic:

the oldest or worst adapted individuals of the previous generation are eliminated.

EXAMPLE

Suppose we want to maximize the function

$$f(x) = x^2$$

between the integers of the interval $[0, 4095]$.

1.- We are going to represent the solutions as **binary strings of length 12** (that is, our chromosomes are sequences of 12 zeros and ones, which correspond to the binary representation of the integers between 0 and 4095).

Clearly, we already know a priori that the exact solution in this case exists and is equal to the chromosome:

$$C_{\text{sol}} = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$$

2.- In this case, the **evaluation function** is the function **f** itself.

That is, given a chromosome C, its fitness is obtained by passing C to decimal representation and taking the square of this value.

3.- We create an initial population of 8 individuals. Suppose it is the following:

population member	string	x	fitness
1	110101100100	3428	11751184
2	010100010111	1303	1697809
3	101111101110	3054	9326916
4	010100001100	1292	1669264
5	011101011101	1885	3553225
6	101101001001	2889	8346321
7	101011011010	2778	7717284
8	010011010101	1237	1530169

- These individuals (chromosomes) have been chosen **at random**.
- For each chromosome, x is the solution they represent (that is, their decimal value) and fitness is the value given by the **adjustment function**, $f(x) = x^2$

- In this case, none of the individuals is in fact a solution, so we are going to create a **new population** from this that allows us to move towards the solution.
- To do this, in this case we are going to select the **4 chromosomes** with the best fitness value: 1,3,6 and 7 and discard the rest.

population member	string	x	fitness	
1	110101100100	3428	11751184	←
2	010100010111	1303	1697809	
3	101111101110	3054	9326916	←
4	010100001100	1292	1669264	
5	011101011101	1885	3553225	
6	101101001001	2889	8346321	←
7	101011011010	2778	7717284	←
8	010011010101	1237	1530169	

- This is our selection mechanism.
- We randomly choose two of these chromosomes (1 and 3, for example) and cross them to obtain two descendants.
- That is, we are going to break the two chromosomes at a random point (the same on both chromosomes) and “cross” the pieces. For example, if we break at the indicated point, we obtain

11/0101100100 11 1111101110
 →
 10/1111101110 10 0101100100

- We do the same process with the other two chromosomes (you can choose a different cut-off point).
- At the end we have 8 chromosomes, the 4 from the initial population and the 4 that we have obtained in the cross.

population member	string	x	fitness
1	11/0101100100	3428	11751184
2	10/1111101110	3054	9326916
3	101101/001001	2889	8346321
4	101011/011010	2778	7717284
5	111111101110	4078	16630084
6	100101100100	2404	5779216
7	101101011010	2906	8444836
8	101011001001	2761	7623121

- Chromosomes 1-4 are those that were in the initial population (indicating where the cut was made to cross them) and chromosomes 5-8 are the descendants.
- The initial population had a mean fitness of 5065797 and the best individual had a fitness of 11751184.
- This new population has a mean fitness of 8402107 and the best individual has a fitness of 16630084.

- Now we repeat the process with this population. On this occasion, the 4 best chromosomes are 1,2,5 and 7.

population member	string	x	fitness	
1	11/0101100100	3428	11751184	←
2	10/1111101110	3054	9326916	←
3	101101/001001	2889	8346321	
4	101011/011010	2778	7717284	
5	111111101110	4078	16630084	←
6	100101100100	2404	5779216	
7	101101011010	2906	8444836	←
8	101011001001	2761	7623121	

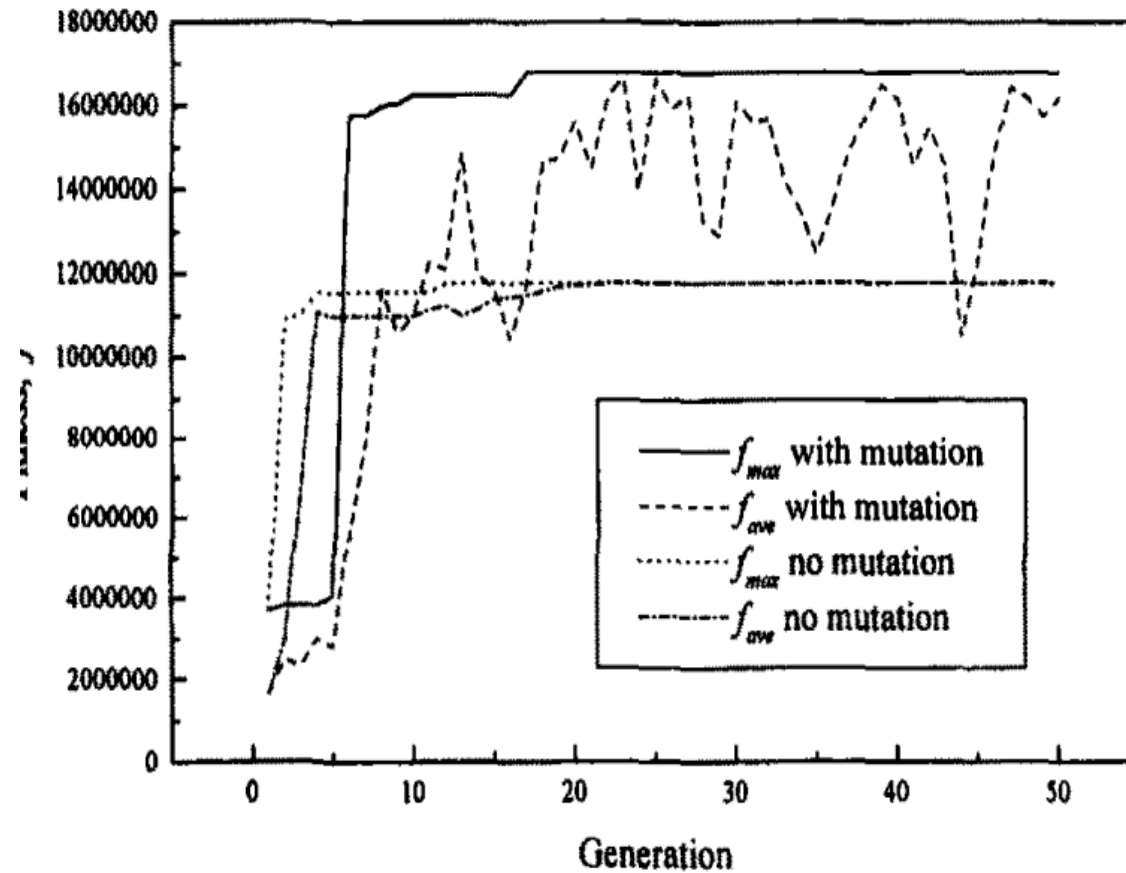
- However, a problem arises here. None of these 4 chromosomes contain a 1 in the last position. Therefore, if we start from them, we will never get a 1 in that position and we will never reach the optimum.

- This justifies the introduction of one more operator apart from the ones we have already seen (selection and crossing):

THE MUTATION

- The idea of the mutation operator is to modify some of the genes at random, in the hope that this will lead in some cases to better solutions.
- In this case, what we do is **mutate each gene of each of the 4 chromosomes chosen with probability 1/12**.
 - o That is, for each gene, we choose a random number between 0 and 1.
 - o If this number is less than 1/12, we change the value of the gene (if it is 0 we put a 1 and vice versa).
 - o If it is greater than or equal to 1/12 we do not modify the gene.
 - o After applying this process to all genes on all chromosomes, we cross over and move on to the next generation.

In this specific case, starting with a somewhat larger population, the algorithm evolves as shown in the figure.



Where f_{max} represents the fitness of the best individual and f_{ave} the mean fitness of the population considered.

PARENT SELECTION METHODS

- o All selection methods try to ensure that the chromosomes with the best fit have a greater probability of being chosen, but chance is involved in all of them.
- o In any case, it is important to point out that the number of parents to select is a parameter set by the programmer.
- o In general, p individuals are chosen from the total population of k individuals, where:
 - If $p = k$ and the descendants replace the parents, we have the **generational model**
 - If $p < k$ and therefore in the new population there are p descendants of the previous one and $k-p$ individuals who were already in the previous generation, we speak of the **stationary model**.
- o In any case, at least a priori, the size of the population remains **constant**.

ROULETTE METHOD (OR SELECTION PROPORTIONAL TO THE FIT):

If we have k individuals C_1, \dots, C_k , the probability of choosing chromosome i as parent for the next generation is given by:

$$P(C_i) = f(C_i) / (f(C_1) + \dots + f(C_k))$$

where f is the fit function.

That is to say:

the probability of choosing the i -th chromosome is equal to the value of the fit function on the i -th chromosome divided by the sum of the fit values of all chromosomes in the population.

IMPORTANT:

1.-The above procedure works well if the best solutions correspond to the highest values of the adjustment function.

ç

Furthermore, since we are looking to obtain a probability (that is, a number in $[0,1]$), the adjustment function must be positive.

2.-The assignment of probabilities to the chromosomes in this way is not exclusive to the roulette method

EXAMPLE:

Suppose we have 4 chromosomes C1, C2, C3 and C4. Suppose their corresponding adjustment values are:

$$f(C1)=4 \quad f(C2)=8 \quad f(C3)=6 \quad f(C4)=12$$

So, the probability that each of these chromosomes is chosen to give rise to the next generation is:

$$P(C1)=4/(4+8+6+12)=4/30=2/15$$

$$P(C2)=8/30=4/15$$

$$P(C3)=6/30=1/5$$

$$P(C4)=12/30=2/5$$

IMPLEMENTATION OF THE METHOD OF ROULETTE

- o The basic idea is to divide the unit interval $[0,1]$ into disjoint “pieces” proportional to the selection probabilities that we just calculated.
- o If we have k individuals, and the i th individual has a probability of being selected equal to $P(C_i)$, we can partition the interval as follows:

Chromosome C_1 is assigned the subinterval $[0, P(C_1)]$

To chromosome C_2 we assign the subinterval $[P(C_1), P(C_1)+P(C_2)]$

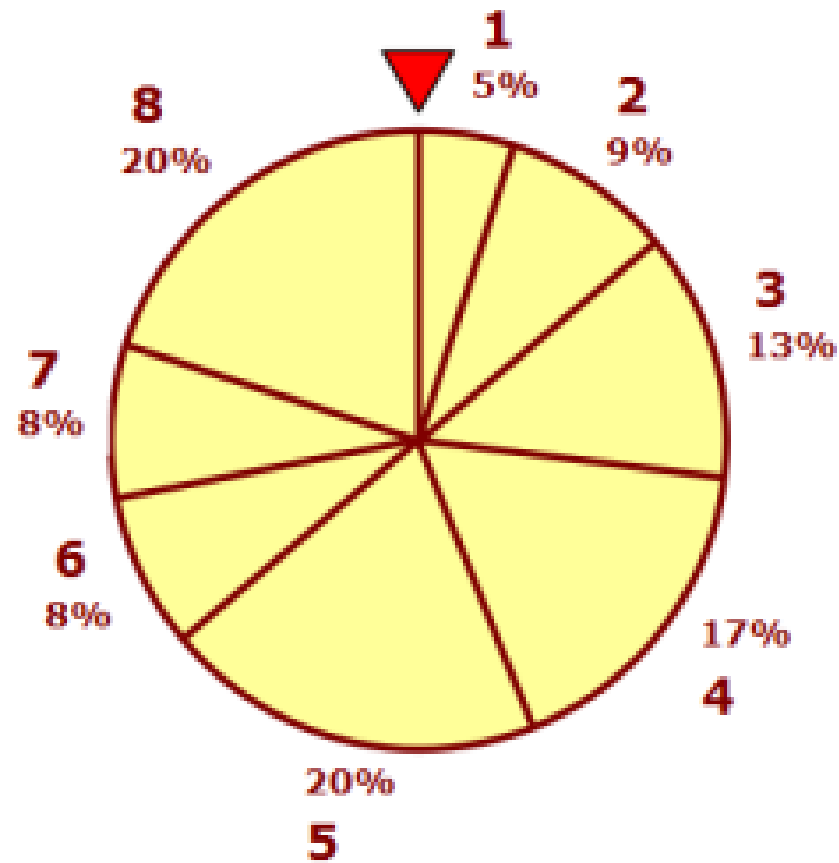
Chromosome C_3 is assigned the subinterval $[P(C_1)+P(C_2), P(C_1)+P(C_2)+P(C_3)]$

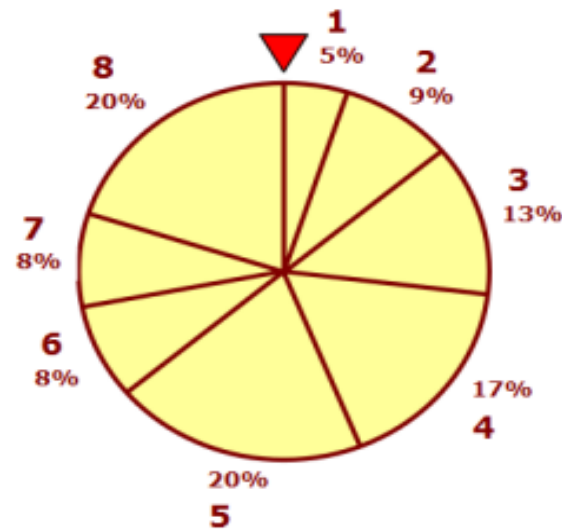
and so on until chromosome k , which will be assigned the subinterval

$$[P(C_1)+\dots+P(C_{k-1}), 1]$$

- o The order in which the subintervals are considered is irrelevant.
- o Also, some of the subintervals could have length 0.

o Now, we can think of the unit interval as a circle (if we fold it and glue the ends together). In this way, we obtain a roulette in which each sector i corresponds to the subinterval associated with chromosome C_i .





- o In the drawing, we have represented a population of 8 individuals.
- o The percentage represented is proportional to the fitness value of each one of them.
- o In particular, as chromosome 5 has a higher value, it will be chosen more times.
- o To select a parent with this method, a random number p in $[0,1]$ is chosen.
- o The number p will be in one of the previous subintervals.
- o The chromosome chosen as the parent is the one that corresponds to said subinterval.
- o The process is repeated, each time choosing a number at random, until the desired number of parents is obtained.

NOTE: The method allows the same parent being chosen more than once.

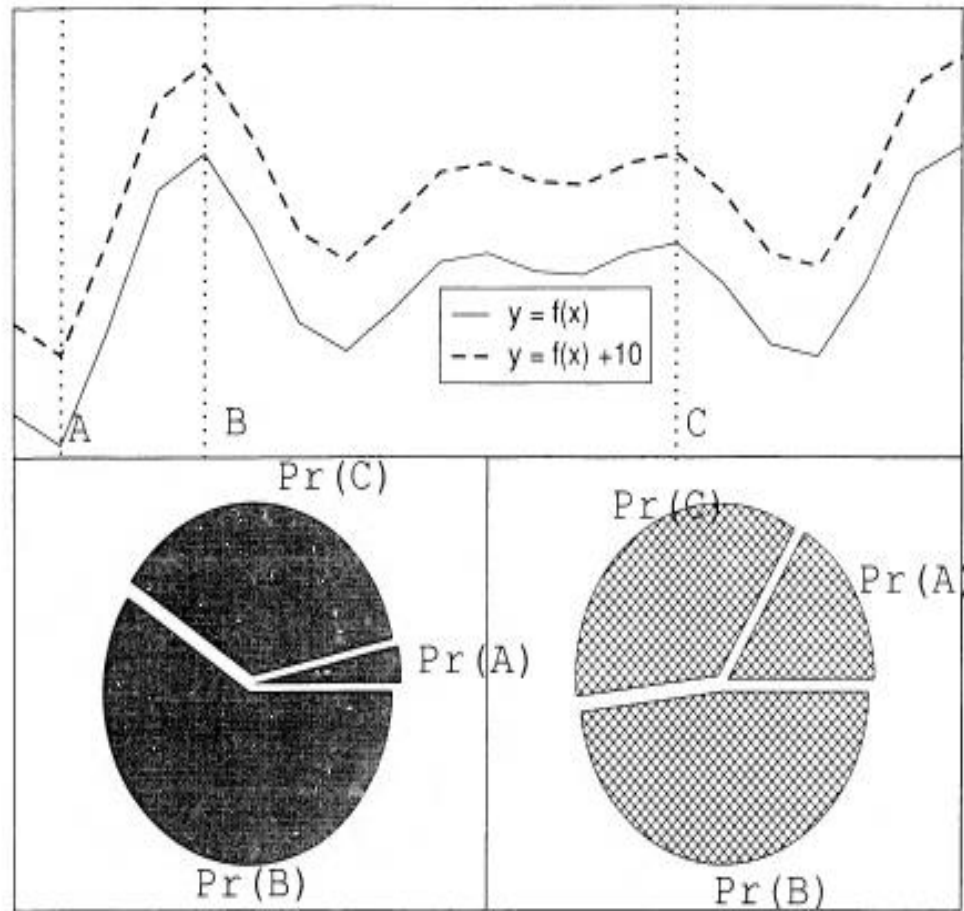
METHOD PROBLEMS

1.- Individuals with a much better fitness than the rest tend to dominate the population very quickly.

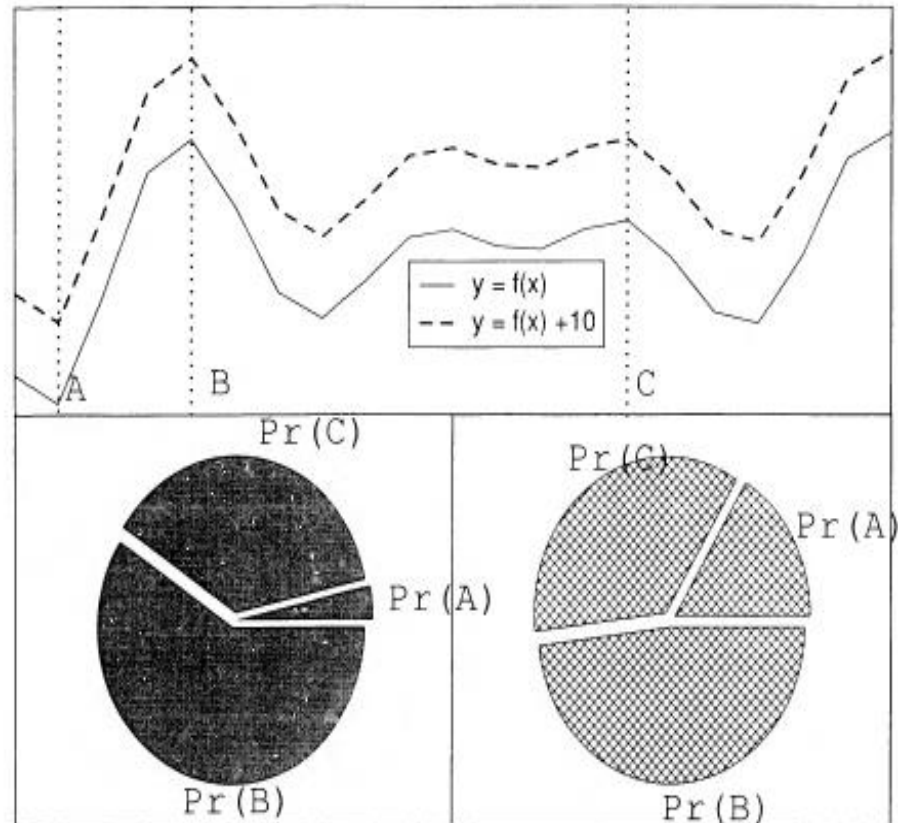
This is what is called **premature convergence**.

2.- If all individuals have a very similar adjustment value, there is no evolutionary pressure (it is not more likely to choose some than others), so the convergence of the algorithm will be **very slow**.

3.- Modifications of the adjustment function can have a great influence on this method.



o In this case, for the same population of three chromosomes (A, B and C) we show what happens for the adjustment function $f(x)$ y for the fitting function $g(x) = f(x) + 10$.



- It is observed, for example, that the relative pre-eminence of B has decreased considerably.

- For the first problem (**premature convergence**), a possible solution is the so-called **Goldberg sigma scaling**.
 - The idea is to determine the selection probabilities using, not the original fit function $f(x)$, but the

modified fit function:

$$f'(x) = \max(f(x) - (\bar{f} - c \cdot \sigma_f), 0)$$

Where

\bar{f} denotes the mean of all the fit values,

σ_f denotes the mean deviation and

c is a constant that generally takes the value 2.

o Regarding the second problem (**very slow algorithm convergence**), the **windowing method** is usually used, which consists of subtracting a value from the evaluation function that depends on the current population and on how the algorithm has evolved so far.

o For example, the simplest windowing is to simply define a parameter **b** as the minimum of all fit values in the current population

$$\mathbf{b} = \min f(C_i)$$

o Then we consider the new fit function

$$\mathbf{g(x) = f(x) - b \text{ (which is non-negative)}}$$

or we use this adjustment function to carry out the selection of parents.

o As this minimum value **b** can change very quickly from one generation to the next, it can be replaced by the average value of the worst fitness in a certain number (set by the programmer) of previous generations.

RANKING METHOD:

- o In this case, the probability that a chromosome is chosen as a parent only depends on **its relative position with respect to the rest of the chromosomes in the population.**
- o The idea is to order all the chromosomes in the population according to their valuation, and then assign a probability to each of the positions in the resulting ranking.
- o One of the easiest ways to do this is the following:
 - o We consider a generational model with μ individuals.
 - o We order these individuals in an increasing way according to their adjustment function, so that position 0 of the ranking corresponds to the individual (or individuals) with the worst fitness and the **position μ** corresponds to the individual (or individuals) with the best fitness.
 - o Then, the probability of being chosen of the chromosomes that occupy position **i** of the ranking is given by:

$$P_{lin-rank}(i) = \frac{(2-s)}{\mu} + \frac{2i(s-1)}{\mu(\mu-1)},$$

where s is a parameter in] 1,2].

IMPORTANT: chromosomes with the same fitness value occupy the same position in the ranking

The following table shows the differences between the roulette method and the ranking method:

	Fitness	Rank	P_{selFP}	$P_{selLR} (s = 2)$	$P_{selLR} (s = 1.5)$
A	1	3	0.1	0	0.167
B	5	2	0.5	0.67	0.5
C	4	1	0.4	0.33	0.33
Sum	10		1.0	1.0	1.0

Table 3.7. Fitness proportionate (FP) versus linear ranking (LR) selection

Example: $s=2$, $\mu = 3$, $i(A)=0$, $P_{sel}(A) = (2-2)/2 + 2 \times 0 (2-1) / 3(3-1) = 0$

$s=2$, $\mu = 3$, $i(B)=2$, $P_{sel}(B) = (2-2)/2 + 2 \times 2 (2-1) / 3(3-1) = 2/3 = 0.67$

$s=2$, $\mu = 3$, $i(C)=1$, $P_{sel}(C) = (2-2)/2 + 2 \times 1 (2-1) / 3(3-1) = 1/3 = 0.33$

Now check the results for $s = 1.5$

- o An advantage of the ranking selection method is that it is **invariant due to changes in the adjustment function**.
- o On occasions, the linear expression that we have used above to assign the probability to each chromosome based on its position in the ranking may be **insufficient to induce selective pressure**.
- o Therefore, another expression that can be used instead of the previous one is:

$$P_{exp-rank}(i) = \frac{1 - e^{-i}}{c}.$$

where c is a constant chosen so that the sum of the probabilities assigned to all chromosomes is 1.

TOURNAMENT METHOD:

- o It is possibly the most used method currently in GA applications, due to its simplicity.
- o For both the roulette method and the ranking-based method, it is necessary to know the fitness of all the chromosomes before starting to apply them.
- o In the tournament method, however, such global knowledge is not necessary a priori.
- o The idea of the method is based on making the chromosomes “compete” in “tournaments” with each other, in small groups and choosing the winner or winners of each of the tournaments.
- o More specifically:
 - We randomly choose k individuals from the population (all individuals have the same probability of being chosen).
 - We select the best of these k individuals according to their fitness function.
 - We repeat the process until we complete the desired number of parents.

o Within the tournament method, in addition, we must distinguish two possibilities:

- **Tournament without replacement.**

The individual chosen for a tournament can not be chosen again for that tournament.

- **Tournament with replacement:**

The individual chosen for a tournament can be chosen again for that tournament.

o The probability that a given chromosome is chosen in the tournament method depends on 3 factors:

1.- The fitness of some chromosome in relation to the fitness of the rest of the chromosomes.

Note that in this case it is not necessary to previously order the entire population, as in the case of ranking.

2.- The size k of the tournaments.

The larger this size, the greater the probability that chromosomes with a higher than average fitness will be included.

3.- If the tournament is with or without replacement.

In the second case, for tournaments of size k , the worst individuals $k-1$ in the population have no chance of being chosen.

In the first case, however, every chromosome has a chance, even a small one, of being chosen.

- o One more variant is the **stochastic tournament method**.
- o In this case, a probability p is set at $[0,1]$, chosen the k individuals that take part in the tournament:
 - We assign probability p to the best of the k individuals
 - We assign probability $p(1-p)$ to the second best of the k individuals.
 - We assign probability $p(1-p)^2$ to the third best of the individuals
 - ...
 - We assign probability $p(1-p)^{(k-1)}$ to the worst of the individuals
- o Then we proceed to choose the winner using that probability p .

The case $p = 1$, which is the one we have used, corresponds to the deterministic case.