

Module VI

Introduction to genetic algorithm, operators in genetic algorithm - coding - selection - cross over – mutation, Stopping condition for genetic algorithm flow, Genetic neuro hybrid systems, Genetic-Fuzzy rule based system.

Genetic Algorithms are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics.

As such they represent an intelligent exploitation of a random search used to solve. Although randomized, GAs are by no means random; instead they exploit historical information to direct

the search into the region of better performance within the search space. The basic techniques of the GAs

are designed to simulate processes in natural systems necessary for evolution, especially those that follow the

principles first laid down by Charles Darwin, "survival of the fittest," because in nature, competition among

individuals for scarce resources results in the fittest individuals dominating over the weaker ones.

OPERATORS IN GENETIC ALGORITHM

1) Encoding

Encoding is a process of representing individual genes. The process can be performed using bits, numbers, trees, arrays, lists or any other objects. The encoding depends mainly on solving the problem. For example, one can encode directly real or integer numbers.

Binary Encoding

The most common way of encoding is a binary string, which would be represented as in Figure below.

Chromosome 1	1 1 0 1 0 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 1 1 1 1 1 1 0 0

Each chromosome encodes a binary (bit) string. Each bit in the string can represent some characteristics of the solution. Every bit string therefore is a solution but not necessarily the best solution.

Another possibility is that the whole string can represent a number. The way bit strings can code differs from problem to problem.

Binary encoding gives many possible chromosomes with a smaller number of alleles. On the other hand, this encoding is not natural for many problems and sometimes corrections must be made after generic operationis completed. Binary coded strings with 1s and 0s are mostly used. The length of the string depends on theaccuracy. In such coding

1. Integers are represented exactly.
2. Finite number of real numbers can be represented.
3. Number of real numbers represented increases with string length.

Octal Encoding

This encoding uses string made up of octal numbers (0-7)

Chromosome 1	03467216
Chromosome 2	15723314

Hexadecimal Encoding

This encoding uses string made up of hexadecimal numbers (0-9, A-F).

Chromosome 1	9CE7
Chromosome 2	3DBA

Permutation Encoding (Real Number Coding)

Every chromosome is a string of numbers, represented in a sequence. Sometimes corrections have to be doneafter generic operation is complete. In permutation encoding, every chromosome is a suing of integer/realvalues, which represents number in a sequence.

Permutation encoding is only useful for ordering problems. Even for this problem,sometypes of crossover and mutation corrections must be made to leave the chromosome consistent (i.e., have realsequence in it).

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Value Encoding

Every chromosome is a string of values and the values can be anything connected to the problem. This encoding produces best results for some special problems. On the other hand, it is often necessary to developnew genetic operator's

specific to the problem. Direct value encoding can be used in problems, where some complicated values, such as real numbers, are used. Use of binary encoding for this type of problems would be very difficult.

In value encoding, every chromosome is a string of some values. Values can be anything connected to problem, from numbers, real numbers or characters to some complicated objects. Value encoding is very good for some special problems. On the other hand, for this encoding it is often necessary to develop some new crossover and mutation specific for the problem.

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

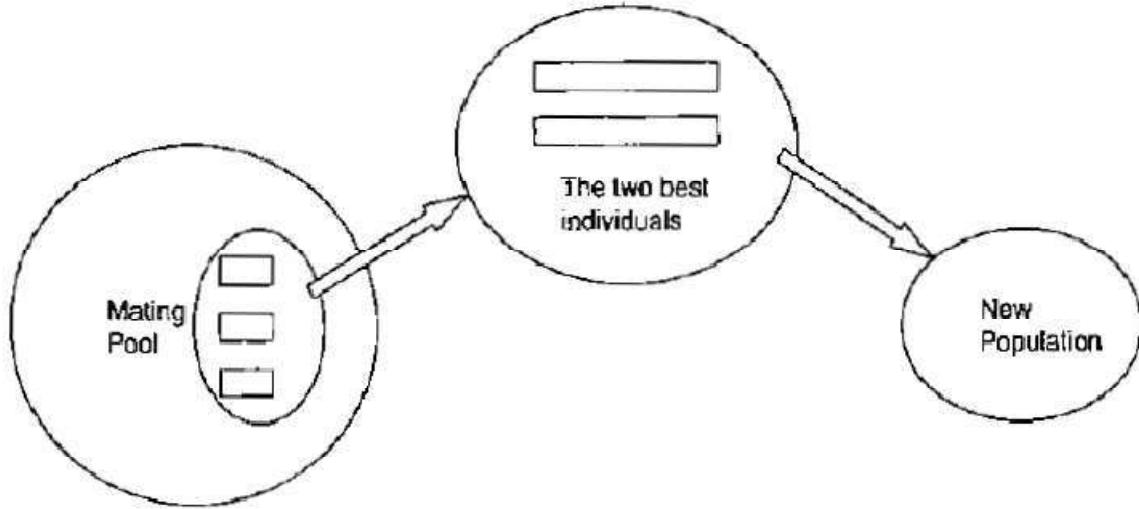
Tree Encoding

This encoding is mainly used for evolving program expressions for genetic programming. Every chromosome is a tree of some objects such as functions and commands of a programming language.

Selection

Selection is the process of choosing two parents from the population for crossing. After deciding on an encoding, the next step is to decide how to perform selection, i.e., how to choose individuals in the population that will create offspring for the next generation and how many offspring each will create. The purpose of selection is to emphasize fitter individuals in the population in hopes that their offspring have higher fitness.

Chromosomes are selected from the initial population to be parents for reproduction. The problem is how to select these chromosomes. According to Darwin's theory of evolution the best ones survive to create new offspring. Figure below shows the basic selection process.



Selection is a method that randomly picks chromosomes out of the population according to their evaluation function. The higher the fitness function the better chance that an individual will be selected. The selection pressure is defined as the degree to which the better individuals are favored. The higher the selection pressure, the more the better individuals are favored. This selection pressure drives the GA to improve the population fitness over successive generations.

The convergence rate of GA is largely determined by the magnitude of the selection pressure, with higher selection pressures resulting in higher convergence rates. GAs should be able to identify optimal or nearly optimal solutions under a wide range of selection scheme pressure. However, if the selection pressure is too low the convergence rate will be slow, and the GA will take unnecessarily longer to find the optimal solution.

If the selection pressure is too high, there is an increased chance of the GA prematurely converging to an incorrect (sub-optimal) solution. In addition to providing selection pressure, selection schemes should also preserve population diversity, as this helps to avoid premature convergence.

Typically we can distinguish two types of selection scheme:

- proportionate-based selection and
- ordinal-based selection.

Proportionate-based selection picks out individuals based upon their fitness values relative to the fitness of the other individuals in the population. Ordinal-based selection schemes select individuals not upon their raw fitness, but upon their rank within the population. This requires that the selection pressure is independent

of the fitness distribution of the population, and is solely based upon the relative ordering(ranking) of the population.

Roulette Wheel Selection

Roulette selection is one of the traditional GAselection techniques. The commonly used reproduction operatoris the proportionate reproductive operator where a string is selected from the mating Pool with a probabilityproportional to the fitness. The principle of Roulette selection is a linear search through a Roulette wheel withthe slots in the wheel weighted in proportion to the individual's fitness values.

A target value is set, which is a random proportion of the sum of the fitnesses inthe population. The population is stepped through untilthe target value is reached. This is only a moderately strong selection technique, since for individuals are not guaranteed to be selected for, bur somewhat have a greater chance. A fit individual will contribute more to the target value, but if it does not exceed it, the next chromosome in line has a chance, and it may be weak.

It is essential that the population not be sorted by fitness, since this would dramatically bias the selection.The Roulette process can also be explained as follows: The expected value of an individual is individual'sfitness divided by the actual fitness of the population. Each individual is assigned a slice of the Roulette wheel,the size of the slice being proportional to the individual's fitness. The wheel is spun N times, where N is the number of individuals in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation. This method is implemented as follows:

1. Sum the total expected value of the individuals in the population. Let it be T .
2. Repeat N times:
 - i. Choose a random integer " r " between 0 and T .
 - ii. Loop through the individuals in the population, summing the expected values, until the sum is greater than or equal to " r ." The individual whose expected value puts the sum over this limit is the oneselected.

Roulette wheel selection is easier to implement bur is noisy. The rate of evolution depends on the varianceof fitness's in the population.

Random Selection

This technique randomly selects a parent from the population. In terms of disruption of generic codes, random selection is a little more disruptive, on average, than Roulette wheel selection.

Rank Selection

The Roulette wheel will have a problem when the fitness values differ very much. If the best chromosome fitness is 90%, its circumference occupies 90% of Roulette wheel, and then other chromosomes have too few chances to be selected. Rank Selection ranks the population and every chromosome receives fitness from the ranking. The worst has fitness 1 and the best has fitness N . It results in slow convergence but prevents too quick convergence. It also keeps up selection pressure when the fitness variance is low. It preserves diversity and hence leads to a successful search. In effect, potential parents are selected and a tournament is held to decide which of the individuals will be the parent.

There are many ways this can be achieved and two suggestions are:

1. Select a pair of individuals at random. Generate a random number R between 0 and 1. If $R < r$ use the first individual as a parent. If $R \geq r$ then use the second individual as the parent. This is repeated to select the second parent. The value of r is a parameter to this method.
2. Select two individuals at random. The individual with the highest evaluation becomes the parent. Repeat to find a second parent.

Tournament Selection

An ideal selection strategy should be such that it is able to adjust its selective pressure and population diversity so as to fine-tune GA search performance. Unlike, the Roulette wheel selection, the tournament selection strategy provides selective pressure by holding a tournament competition among N individuals.

The best individual from the tournament is the one with the highest fitness, who is the winner of N . Tournament competitions and the winner are then inserted into the mating pool. The tournament competition is repeated until the mating pool for generating new offspring is filled. The mating pool comprising the tournament winner has higher average population fitness. The fitness difference provides the selection pressure, which drives GA to improve the fitness of the succeeding genes. This method is more efficient and leads to an optimal solution.

Boltzmann Selection

In Boltzmann selection, a continuously varying temperature controls the rate of selection according to a preset schedule. The temperature starts out high, which means that the selection pressure is low. The temperature is gradually lowered, which gradually increases the selection pressure, thereby allowing the GA to narrow in more closely to the best part of the search space while maintaining the appropriate degree of diversity.

A logarithmically decreasing temperature is found useful for convergence without getting stuck to a local minima state. However, it takes time to cool down the system to the equilibrium state.

Let f_{\max} be the fitness of the currently available best string. If the next string has fitness $f(X_i)$ such that $f(X_i) > f_{\max}$, then the new string is selected. Otherwise it is selected with Boltzmann probability

$$P = \exp[-(f_{\max} - f(X_i))/T]$$

Where, $T = T_0(1-\alpha)^k$ and $k = (1 + 100 * g/G)$; g is the current generation number; G the maximum value of g . The value of α can be chosen from the range [0, 1] and that of T_0 from the range [5, 100]. The final state is reached when computation approaches zero value of T_1 , i.e., the global solution is achieved at this point.

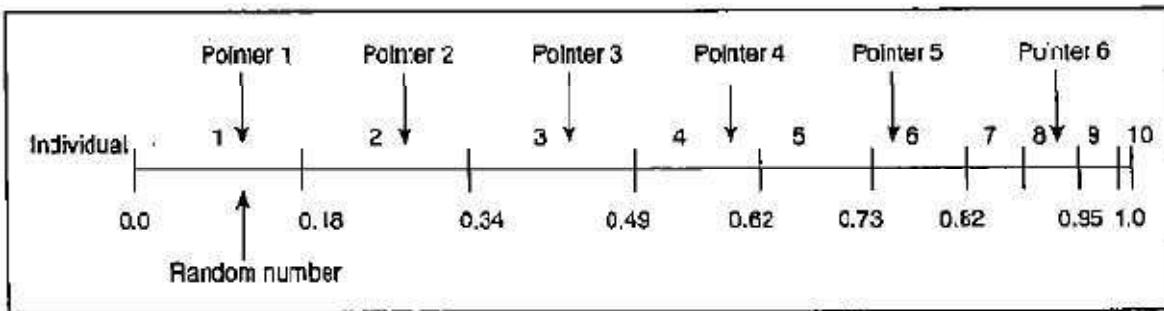
Elitism

The first best chromosome or the few best chromosomes are copied to the new population. The rest is done in a classical way. Such individuals can be lost if they are not selected to reproduce or if crossover or mutation destroys them. This significantly improves the GA's performance.

Stochastic Universal Sampling

Stochastic universal sampling provides zero bias and minimum spread. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness exactly as in Roulette wheel selection. Here equally spaced pointers are placed over the line, as many as there are individuals to be selected. Consider $N_{Pointer}$ the number of individuals to be selected, then the distance between the pointers are $1/N_{Pointer}$ and the position of the first pointer is given by a randomly generated number in the range

$[0,1/N\text{Pointer}]$. For 6 individuals to be selected, the distance between the pointers is $1/6 = 0.167$. Figure below shows the selection for the above example.



Sample of 1 random number in the range $[0, 0.167]$: 0.1.

After selection the mating population consists of the individuals,

1,2,3,4,6,8

Crossover (Recombination)

Crossover is the process of taking two parent solutions and producing from them a child. After the selection(reproduction) process, the population is enriched with better individuals. Reproduction makes clones of good strings but does not create new ones.

Crossover operator is applied to the mating pool with the hope that it creates a better offspring.

Crossover is a recombination operator that proceeds in three steps:

1. The reproduction operator selects at random a pair of two individual strings for the mating.

2. A cross site is selected at random along the string length.

3. Finally, the position values are swapped between the two strings following the cross site.

That is the simplest way how to do that is to choose randomly some crossover point and copy everything before this point from the first parent and then copy everything after the crossover point from the other parent.

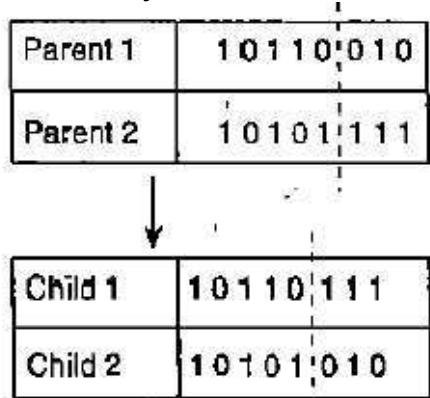
The various crossover techniques are in the following:

Single-Point Crossover

The traditional genetic algorithm uses single-point crossover, where the two mating chromosomes are occurrence at corresponding points and the sections after the cuts exchanged. Here, a cross site or crossover point is selected randomly along the length of the mated strings and bits next to the cross sites are exchanged.

If appropriate site is chosen, better children can be obtained by combining good parents, else it severely hampers string quality.

The Figure below illustrates single point crossover and it can be observed that the bits next to the crossover point are exchanged to produce children. The crossover point can be chosen randomly.

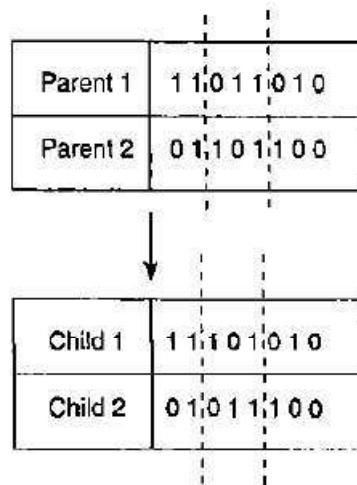


Two Point Crossover

Apart from single point crossover, many different crossover algorithms have been devised, often involving more than one cut point. It should be noted that adding further crossover points reduces the performance of the GA. The problem with adding additional crossover points is that building blocks are more likely to be disrupted. However, an advantage of having more crossover points is that the problem space may be searched more thoroughly.

In two-point crossover, two crossover points are chosen and the contents between these points are exchanged between two mated parents.

In Figure below the dotted lines indicate the crossover points. Thus the contents between these points are exchanged between the parents to produce new children for mating in the next generation.



Multipoint Crossover (N-Point Crossover)

There are two ways in this crossover. One is even number of cross sites and the other odd number of cross sites.

In the case of even number of cross sires, the cross sites are selected randomly around a circle and informationis exchanged.

In the case of odd number of cross sites, a different cross point is always assumed at the stringbeginning.

Uniform Crossover

Uniform crossover is quite different from the N-point crossover. Each gene in the offspring is created bycopying the corresponding gene from one or the other parent chosen according to a random generated binarycrossover mask of the same length as the chromosomes. Where there is a 1 in the crossover mask, the gene is copied from the first parent, and where there is a 0 in the mask the gene is copied from the second parent.

Anew crossover mask is randomly generated for each pair of parents. Offspring, therefore, contain a mixtureof genes from each parent. The number of effective crossing point is not fixed, but will average $L/2$ (where L is the chromosome length).

In Figure below, new children are produced using uniform crossover approach. It can be noticed thatwhile producing child 1, when there is a 1 in the mask, the gene is copied from parent 1 else it is copied fromparent 2. On producing child 2, when there is a 1 in the mask, the gene is copied from parent 2, and when there is a 0 in the mask, the gene is copied from the parent 1.

Parent 1	1 0 1 1 0 0 1 1
Parent 2	0 0 0 1 1 0 1 0
Mask	1 1 0 1 0 1 1 0
Child 1	1 0 0 1 1 0 1 0
Child 2	0 0 1 1 0 0 1 1

Three-Parent Crossover

In this crossover technique, three parents are randomly chosen. Each bit of the first parent is compared withthe bit of the second parent. If both are the same,

the bit is taken for the offspring, otherwise the bit from the third parent is taken for the offspring. This concept is illustrated in Figure below.

Parent 1	1 1 0 1 0 0 1
Parent 2	0 1 1 0 1 0 1
Parent 3	0 1 1 0 1 1 0 0
Child	0 1 1 0 1 0 0 1

Crossover with Reduced Surrogate

The reduced surrogate operator constraints crossover to always produce new individuals wherever possible. This is implemented by restricting the location of crossover points such that crossover points only occur where gene values differ.

Shuffle Crossover

Shuffle crossover is related to uniform crossover. A single crossover position (as in single-point crossover) is selected. But before the variables are exchanged, they are randomly shuffled in both parents. After recombination, the variables in the offspring are unshuffled. This removes positional bias as the variables are randomly reassigned each time crossover is performed.

Precedence Preservative Crossover

Precedence preservative crossover (PPX) was independently developed for vehicle routing problems. The operator passes on precedence relations of operations given in two parental permutations to one offspring at the same race, while no new precedence relations are introduced.

PPX is illustrated below for a problem consisting of six operations A-F. The operator works as follows:

1. A vector of length Σ_m , sub $i = 1$ to m , representing the number of operations involved in the problem, is randomly filled with elements of the set {1, 2}.
2. This vector defines the order in which the operations are successively drawn from parent 1 and parent 2.
3. We can also consider the parent and offspring permutations as lists, for which the operations "append" and "delete" are defined.
4. First we start by initializing an empty offspring.
5. The leftmost operation in one of the two parents is selected in accordance with the order of parents given in the vector.
6. After an operation is selected, it is deleted in both parents.
7. Finally the selected operation is appended to the offspring.

8. Step 7 is repeated until both parents are empty and the offspring contains all operations involved.

PPX does not work in a uniform-crossover manner due to the "deletion-append" scheme used. Example is shown in Figure below.

Parent permutation 1	A	B	C	D	E	F
Parent permutation 2	C	A	B	F	D	E
Select parent no. (1/2)	1	2	1	1	2	2
Offspring permutation	A	C	B	D	F	E

Ordered Crossover

Ordered two-point crossover is used when the problem is order based, for example in U-shaped assemblyline balancing, etc. Given two parent chromosomes, two random crossover points are selected partitioning them into a left, middle and right portions. The ordered two-point crossover behaves in the following way:

child 1 inherits its left and right section from parent 1, and its middle section is determined by the genes in the middle section of parent 1 in the order in which the values appear in parent 2. A similar process is applied to determine child 2. This is shown in Figure below.

Parent 1: 4 2 | 1 3 | 6 5 Child 1: 4 2 | 3 1 | 6 5

Parent 2: 2 3 | 1 4 | 5 6 Child 2: 2 3 | 4 1 | 5 6

Partially Matched Crossover

Partially matched crossover (PMX) can be applied usefully in the TSP. Indeed, TSP chromosomes are simply sequences of integers, where each integer represents a different city and the order represents the time at which a city is visited. Under this representation, known as permutation encoding, we are only interested in labels and not alleles.

It may be viewed as a crossover of permutations that guarantees that all positions are found exactly once in each offspring, i.e., both offspring receive a full complement of genes, followed by the corresponding filling in of alleles from their parents.

PMX proceeds as follows:

- I. The two chromosomes are aligned.
2. Two crossing sites are selected uniformly at random along the strings, defining a matching section.

3. The matching section is used to effect a cross through position-by-position exchange operation.

4. Alleles are moved to their new positions in the offspring.

The following illustrates how PMX works.

Name 9 8 4 . 5 6 7 . 1 3 2 1 0 Allele 1 0 1 . 0 0 1 . 1 1 0 0

Name 8 7 1 . 2 3 1 0 . 9 5 4 6 Allele 1 1 1 . 0 1 1 . 1 1 0 1

Consider the two strings shown in Figure above, where the dots mark the selected cross points. The marching section defines the position-wise exchanges that must take place in both parents to produce the offspring.

The exchanges are read from the marching section of one chromosome to that of the other. In the example illustrate in Figure above, the numbers that exchange places are 5 and 2, 6 and 3, and 7 and 10. The resulting offspring are as shown in Figure below.

Name 9 8 4 . 2 3 1 0 . 1 6 5 7

Allele 1 0 1 . 0 1 0 . 1 0 0 1

Name 8 1 0 1 . 5 6 7 . 9 2 4 3

Allele 1 1 1 . 1 1 1 . 1 0 0 1

Crossover Probability

The basic parameter in crossover technique is the crossover probability (Pt). Crossover probability is a parameter to describe how often crossover will be performed. If there is no crossover, offspring are exact copies of parents. If there is crossover, offspring are made from parts of both parents' chromosome. If crossover probability is 100%, then all offspring are made by crossover. If it is 0%, whole new generation is made from exact copies of chromosomes from old population. Crossover is made in hope that new chromosomes will contain good part of old chromosomes and therefore the new chromosomes will be better. However, it is good to leave some part of old population survive to next generation.

Mutation

After crossover, the strings are subjected to mutation. Mutation prevents the algorithm to be trapped in a local minimum. Mutation plays the role of recovering the lost genetic materials as well as for randomly distributing generic information. It is an insurance policy against the irreversible loss of genetic material. Mutation has been traditionally considered as a simple search operator. If crossover is

supposed to exploit the current solution to find better ones, mutation is supposed to help for the exploration of the whole search space.

Mutation is viewed as a background operator to maintain genetic diversity in the population. It introduces new generic structures in the population by randomly modifying some of its building blocks. Mutation helps escape from local minima's trap and maintains diversity in the population.

There are many different forms of mutation for the different kinds of representation. For binary representation, a simple mutation can consist in inverting the value of each gene with a small probability.

The probability is usually taken about $1/L$, where L is the length of the chromosome. It is also possible to implement kind of hill climbing mutation operators that do mutation only if it improves the quality of the solution. Such an operator can accelerate the search; however, care should be taken, because it might also reduce the diversity in the population and make the algorithm converge toward some local optima. Mutation of a bit involves flipping a bit, changing 0 to 1 and vice-versa.

Flipping

Flipping of a bit involves changing 0 to 1 and 1 to 0 based on a mutation chromosome generated. Figure below explains mutation-flipping concept.

Parent	1 0 1 1 0 1 0 1
Mutation chromosome	1 0 0 0 1 0 0 1
Child	0 0 1 1 1 1 0 0

A parent is considered and a mutation chromosome is randomly generated. For a 1 in mutation chromosome, the corresponding bit in parent chromosome is flipped (0 to 1 and 1 to 0) and child chromosome is produced. In the case illustrated in Figure 15-30, 1 occurs at 3 places of mutation chromosome, the corresponding bits in parent chromosome are flipped and the child is generated.

Interchanging

Two random positions of the string are chosen and the bits corresponding to those positions are interchanged shown in Figure below.

Parent	1 0 1 1 0 1 0 1
Child	1 1 1 1 0 0 0 1

Reversing

A random position is chosen and the bits next to that position are reversed and child chromosome is produced as shown Figure below.

Parent	1 0 1 1 0 1 0 1
Child	1 0 1 1 0 1 1 0

Mutation Probability

An important parameter in the mutation technique is the mutation probability(P_m). It decides how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover(or directly copied) without any change. If mutation is performed, one or more parts of a chromosome are changed.

If mutation probability is 100%, whole chromosome is changed; if it is 0%, nothing is changed. Mutation generally prevents the GA from falling into local extremes. Mutation should not occur very often,because then GA will in fact change to random search.

STOPPING CONDITION FOR GENETIC ALGORITHM FLOW

The various stopping condition are listed as follows:

- 1. Maximum generations:** The GA stops when the specified number of generations has evolved.
- 2. Elapsed time:** The generic process will end when a specified time has elapsed.
- 3. No change in fitness:** The genetic process will end if there is no change to the population's best fitness for a specified number of generations.

4. **Stall generations:** The algorithm stops if there is no improvement in the objective function for a sequence of consecutive generations of length "Stall generations."

5. **Stall time limit:** The algorithm stops if there is no improvement in the objective function during an interval of time in seconds equal to "Stall time limit."

The termination or convergence criterion finally brings the search to a halt. The following are the few methods of termination techniques.

Best Individual

A best individual convergence criterion stops the search once the minimum fitness in the population drops below the convergence value. This brings the search to a faster conclusion, guaranteeing at least one good solution.

Worst Individual

Worst individual terminates the search when the least fit individuals in the population have fitness less than the convergence criteria.

This guarantees the entire population to be of minimum standard, although the best individual may not be significantly better than the worst. In this case, a stringent convergence value may never be met, in which case the search will terminate after the maximum has been exceeded.

Sum of Fitness

In this termination scheme, the search is considered to have satisfaction converged when the sum of the fitness in the entire population is less than or equal to the convergence value in the population record.

This guarantees that virtually all individuals in the population will be within a particular fitness range, although it is better to pair this convergence criteria with weakest gene replacement, otherwise a few unfit individuals in the population will blow out the fitness sum. The population size has to be considered while setting the convergence value.

Median Fitness

Here at least half of the individuals will be better than or equal to the convergence value, which should give a good range of solutions to choose from.