Crossover

Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability. Crossover selects genes from parent chromosomes and creates a new offspring.

The Crossover operators are of many types.

- one simple way is, **One-Point crossover**.
- the others are **Two Point, Uniform, Arithmetic, and Heuristic crossovers**.

The operators are selected based on the way chromosomes are encoded.

One-Point Crossover

One-Point crossover operator randomly selects one crossover point and then copy everything before this point from the first parent and then everything after the crossover point copy from the second parent. The Crossover would then look as shown below.

Consider the two parents selected for crossover.

Parent 1 1 1 0 1 1 | 0 0 1 0 0 1 1 0 1 1 0

Parent 2 1 1 0 1 1 | 1 1 0 0 0 0 1 1 1 1 0

Interchanging the parents chromosomes after the crossover points -
The Offspring produced are :

Offspring 1 1 1 0 1 1 | 1 1 0 0 0 0 1 1 1 1 0

Offspring 2 1 1 0 1 1 | 0 0 1 0 0 1 1 0 1 1 0

Note : The symbol, a vertical line, | is the chosen crossover point.

Two-Point Crossover

Two-Point crossover operator randomly selects two crossover points within a chromosome then interchanges the two parent chromosomes between these points to produce two new offspring.

Consider the two parents selected for crossover :

Parent 1	**1 1 0 1 1 | 0 0 1 0 0 1 1 | 0 1 1 0**
Parent 2	**1 1 0 1 1 | 1 1 0 0 0 0 1 | 1 1 1 0**

Interchanging the parents chromosomes between the crossover points - The Offspring produced are :

Offspring 1	**1 1 0 1 1 | 0 0 1 0 0 1 1 | 0 1 1 0**
Offspring 2	**1 1 0 1 1 | 0 0 1 0 0 1 1 | 0 1 1 0**

Uniform Crossover

Uniform crossover operator decides (with some probability – know as the mixing ratio) which parent will contribute how the gene values in the offspring chromosomes. The crossover operator allows the parent chromosomes to be mixed at the gene level rather than the segment level (as with one and two point crossover).

Consider the two parents selected for crossover.

Parent 1	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**0**
Parent 2	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**

If the mixing ratio is **0.5** approximately, then half of the genes in the offspring will come from parent **1** and other half will come from parent **2**. The possible set of offspring after uniform crossover would be:

Offspring 1	$\mathbf{1}_1$	$\mathbf{1}_2$	$\mathbf{0}_2$	$\mathbf{1}_1$	$\mathbf{1}_1$	$\mathbf{1}_2$	$\mathbf{1}_2$	$\mathbf{0}_2$	$\mathbf{0}_1$	$\mathbf{0}_1$	$\mathbf{0}_2$	$\mathbf{1}_1$	$\mathbf{1}_2$	$\mathbf{1}_1$	$\mathbf{1}_1$	$\mathbf{0}_2$
Offspring 2	$\mathbf{1}_2$	$\mathbf{1}_1$	$\mathbf{0}_1$	$\mathbf{1}_2$	$\mathbf{1}_2$	$\mathbf{0}_1$	$\mathbf{0}_1$	$\mathbf{1}_1$	$\mathbf{0}_2$	$\mathbf{0}_2$	$\mathbf{1}_1$	$\mathbf{1}_2$	$\mathbf{0}_1$	$\mathbf{1}_2$	$\mathbf{1}_2$	$\mathbf{0}_1$

Note: The subscripts indicate which parent the gene came from.

Arithmetic

Arithmetic crossover operator linearly combines two parent chromosome vectors to produce two new offspring according to the equations:

Offspring1 = a * Parent1 + (1- a) * Parent2

Offspring2 = (1 – a) * Parent1 + a * Parent2

where **a** is a random weighting factor chosen before each crossover operation.

Consider two parents (each of 4 float genes) selected for crossover:

Parent 1	**(0.3)**	**(1.4)**	**(0.2)**	**(7.4)**
Parent 2	**(0.5)**	**(4.5)**	**(0.1)**	**(5.6)**

Applying the above two equations and assuming the weighting factor **a = 0.7**, applying above equations, we get two resulting offspring. The possible set of offspring after arithmetic crossover would be:

Offspring 1	**(0.36)**	**(2.33)**	**(0.17)**	**(6.87)**
Offspring 2	**(0.402)**	**(2.981)**	**(0.149)**	**(5.842)**

Heuristic

Heuristic crossover operator uses the fitness values of the two parent chromosomes to determine the direction of the search.

The offspring are created according to the equations:

Offspring1 = BestParent + r * (BestParent − WorstParent)

Offspring2 = BestParent

where **r** is a random number between **0** and **1**.

It is possible that **offspring1** will not be feasible. It can happen if **r** is chosen such that one or more of its genes fall outside of the allowable upper or lower bounds. For this reason, heuristic crossover has a user defined parameter **n** for the number of times to try and find an **r** that results in a feasible chromosome. If a feasible chromosome is not produced after **n** tries, the worst parent is returned as offspring1.

Mutation

After a crossover is performed, mutation takes place.
Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next.

Mutation occurs during evolution according to a user-definable mutation probability, usually set to fairly low value, say **0.01** a good first choice.

Mutation alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With the new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible.

Mutation is an important part of the genetic search, helps to prevent the population from stagnating at any local optima. Mutation is intended to prevent the search falling into a local optimum of the state space.

The Mutation operators are of many type.

- one simple way is, **Flip Bit**.
- the others are **Boundary, Non-Uniform, Uniform, and Gaussian**.

The operators are selected based on the way chromosomes are encoded .

Flip Bit

The mutation operator simply inverts the value of the chosen gene. i.e. **0** goes to **1** and **1** goes to **0**.

This mutation operator can only be used for binary genes.

Consider the two original off-springs selected for mutation.

Original offspring 1	1	1	0	1	1	1	1	0	0	0	0	1	1	1	1	0
Original offspring 2	1	1	0	1	1	0	0	1	0	0	1	1	0	1	1	0

Invert the value of the chosen gene as **0** to **1** and **1** to **0**

The Mutated Off-spring produced are :

Mutated offspring 1	1	1	0	**0**	1	1	1	0	0	0	0	1	1	1	1	0
Mutated offspring 2	1	1	0	1	1	0	**1**	1	0	0	1	1	0	1	**0**	0

- **Boundary**

  The mutation operator replaces the value of the chosen gene with either the upper or lower bound for that gene (chosen randomly).

  This mutation operator can only be used for integer and float genes.

- **Non-Uniform**

  The mutation operator increases the probability such that the amount of the mutation will be close to **0** as the generation number increases. This mutation operator prevents the population from stagnating in the early stages of the evolution then allows the genetic algorithm to fine tune the solution in the later stages of evolution.

  This mutation operator can only be used for integer and float genes.

- **Uniform**

  The mutation operator replaces the value of the chosen gene with a uniform random value selected between the user-specified upper and lower bounds for that gene.

  This mutation operator can only be used for integer and float genes.

- **Gaussian**

  The mutation operator adds a unit Gaussian distributed random value to the chosen gene. The new gene value is clipped if it falls outside of the user-specified lower or upper bounds for that gene.

  This mutation operator can only be used for integer and float genes.