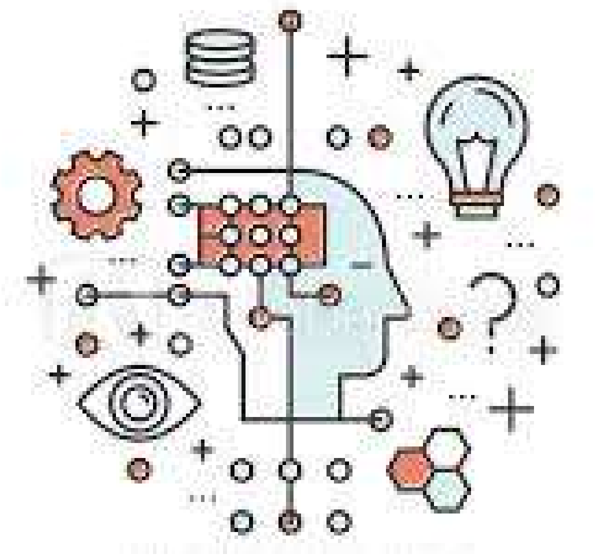
CS 321 SOFT COMPUTING



Dr. Shamama Anwar
Assistant Professor
Department of Computer Science and Engineering,
BIT, Mesra

By:
Shamama Anwar

Module - III

Fundamentals of Genetic Algorithms:

- Basic Concepts
- Creation of Offsprings
- Encoding
- Fitness Functions
- Reproduction

Genetic Modelling:

- Inheritance Operators
- Cross over
- Inversion and detection
- Mutation operator
- Bitwise operators

Basic Concepts

- A **genetic algorithm** (or **GA**) is a search technique used in computing to find true or approximate solutions to optimization and search problems.

- Genetic algorithms are categorized as global search heuristics.

- Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

- Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions.

- Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

Basic Concepts

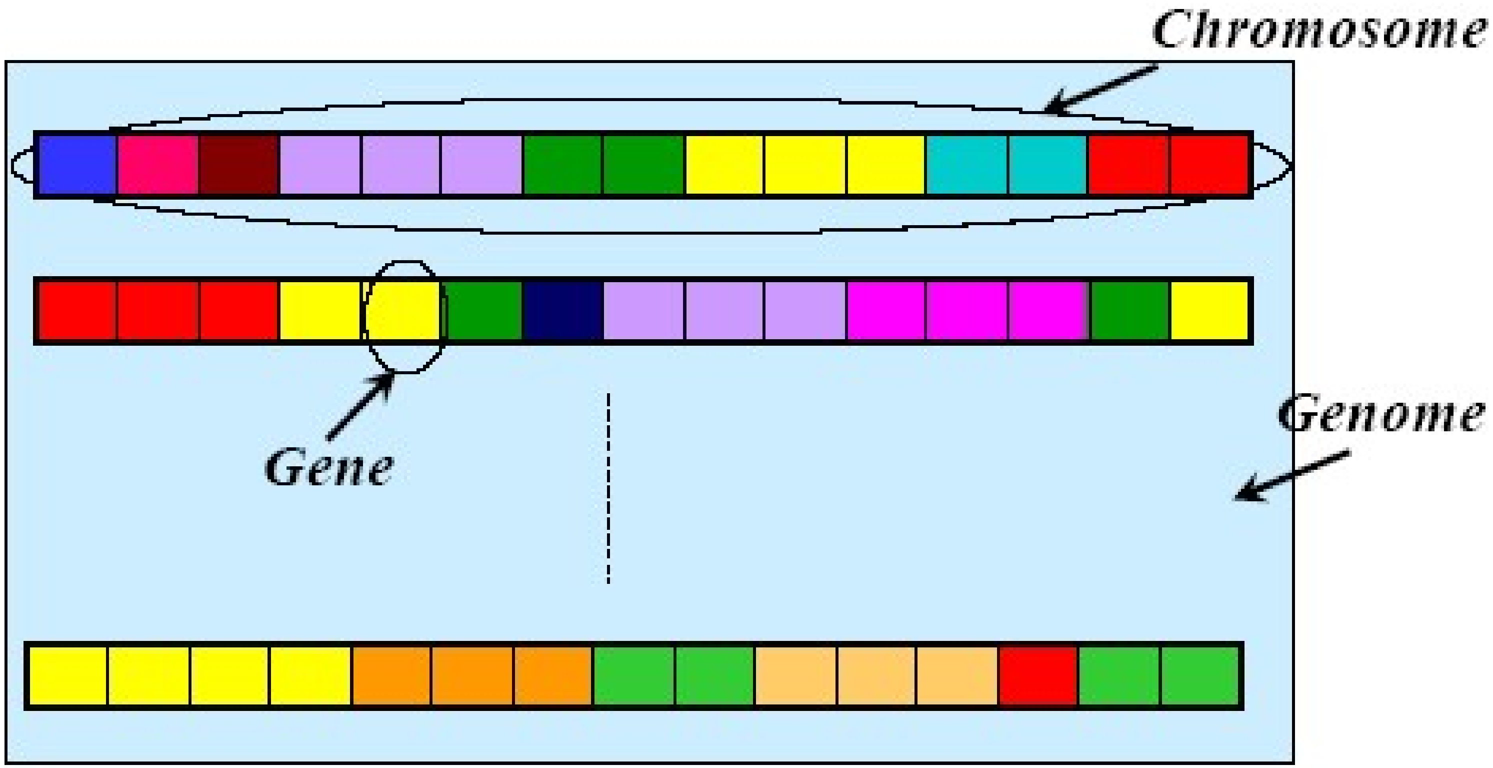
- The evolution usually starts from a population of randomly generated individuals and happens in generations.
- In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population (based on their fitness), and modified (recombined and possibly mutated) to form a new population.
- The new population is then used in the next iteration of the algorithm.
- Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.
- If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.
- Genotype: Particular set of genes in a genome
- Phenotype: Physical characteristic of the genotype (smart, beautiful, healthy, etc.)

Basic Concepts

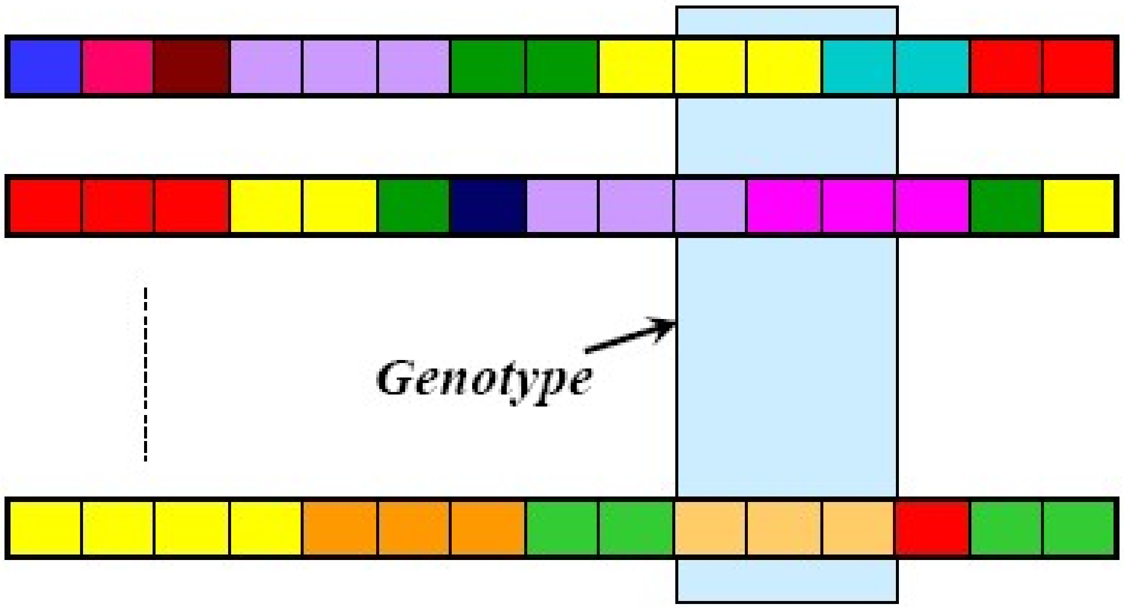
- Three important aspects of using GA:
  - Definition of objective function
  - Definition and implementation of genetic representation
  - Definition and implementation of genetic operators.

- Biological Background
  - All living organism consists of cells
  - Each cell is a set of chromosome which are strings of DNA and consists of genes on blocks of DNA.
  - All genetic information gets stored in the chromosomes.
  - Chromosomes are divided into genes.
  - The possibilities of combination of the genes for one property are called alleles
  - The set of all possible alleles is called a gene pool.
  - The set of all the genes of a specific species is called genome.

Basic Concepts



Basic Concepts



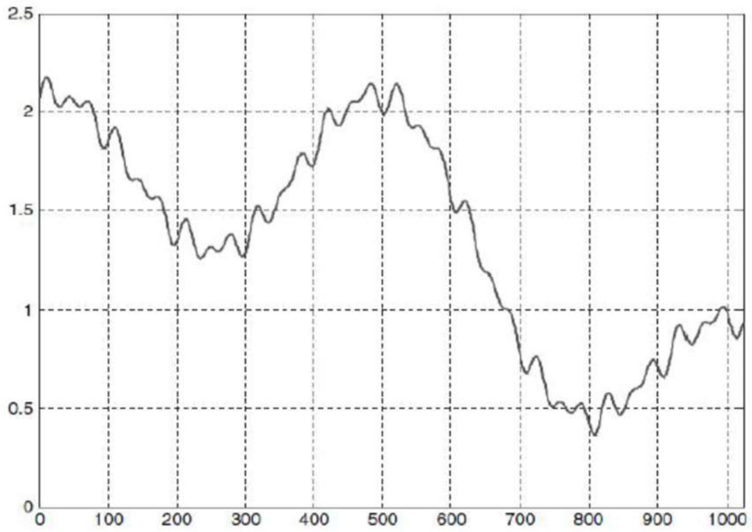
Basic Concepts

Basic Terminology

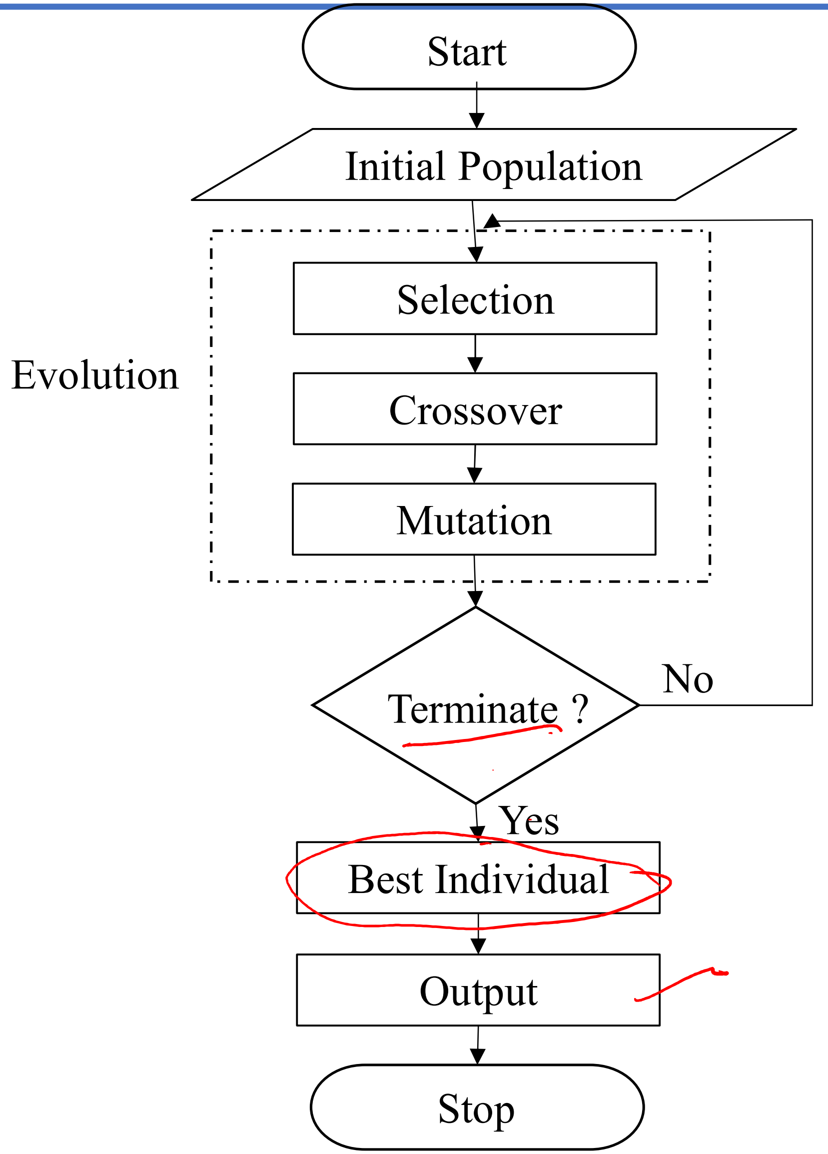
- Individual
  - Individual is a single solution.
  - A chromosome defines one unique solution but that does not mean that each solution is encoded by exactly one chromosome.
- Genes
  - These are the basic instructions for building the a GA.
  - A chromosome is a sequence of genes.
  - Genes may describe a possible solution to a problem.
- Fitness
  - The fitness of an individual in a GA is the value of an objective function for its phenotype.
  - The fitness indicates how good a solution is and also corresponds to how close the chromosome is to the optimal one.
- Populations
  - A population is a collection of individuals.
  - Two important aspect of a population is initial population generation and population size.

Creation of Offspring

- Search space
  - While solving a problem, we work towards some solution which is the best among others.
  - The space for all possible feasible solutions is called search space.
  - Each solution is marked by its value of fitness.
  - The GA algorithm starts with a set of solution called the population.
  - Solutions for one population is taken to form new populations.
  - Solutions to form new population (offspring) is taken based on their fitness value.



GA Flowchart (Working Principle)



Example / Optimization Question

- Consider a problem of maximizing the function:

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

Where x is permitted to vary between 0 and 31.

- Objective function / Fitness function
  - GAs are usually suitable for solving maximization problems.
  - Minimization problem are transformed into maximization problem by some suitable transformation.
  - $F(X) = f(X)$ for maximization problem
  - $F(X) = 1/f(X)$ for minimization problem, if $f(X) \neq 0$
  - $F(X) = 1/(1 + f(X))$, if $f(X) = 0$

Initial Population

- A population is a collection of individuals.
- Population size depends on the complexity of the problem.
- Ideally, the first population should have a gene pool as large as possible in order to explore the shole search space.
- The initial population is generally chosen randomly (heuristic maybe applied).
- If the population chosen lacks diversity, the algorithm will just explore a small part of search space.
- A large population size makes it easier to explore the search space but requires much more computational cost, memory and time.

Encoding

There are many ways of representing individual genes

- Binary Encoding

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

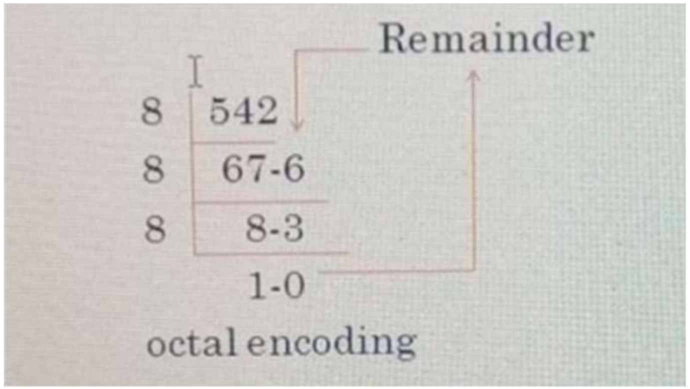
Genotype :

x	y

Phenotype :

0 1 1 0 1	1 1 0 0 1

- Octal Encoding



Encoding

- Hexadecimal Encoding

	Remainder Dec.	Remainder Hex.
16) 2861		
16) 178	13	D
16) 11	2	2
0	11	B

- Permutation Encoding
  - Every chromosome is a string of numbers which represents the number in the sequence:

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

Encoding

- Value Encoding
  - The value coded GA is most suitable for optimization in a continuous search space.
  - Uses the direct representations of the design parameters.
  - Avoids any intermediate encoding / decoding steps.

Chromosome A	1.235 5.323 0.454 2.321 2.454
Chromosome B	(left), (back), (left), (right), (forward)

Genotype :

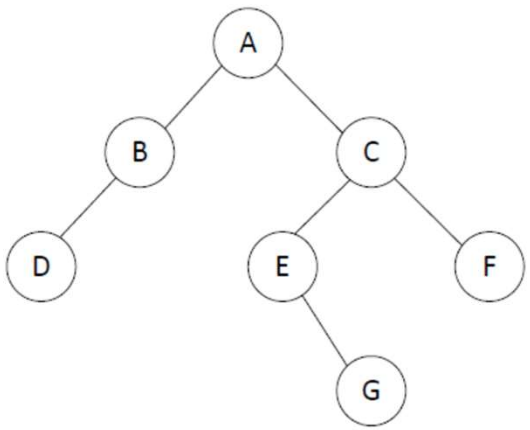
x	**y**

Phenotype :

5.28	**-475.36**

Encoding

- Tree Encoding
  - A solution is encoded in the form of a binary tree.



D A B E G C F (In-order)

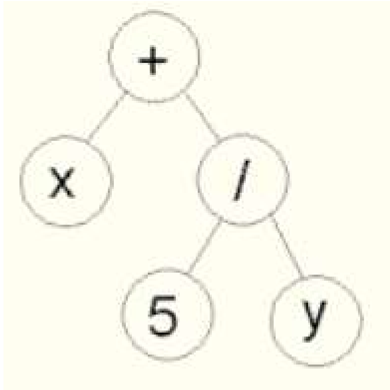
(T_L R T_R)

A B D C E G F (Pre-order)

(R T_L T_R)

D B G E E C A (Post-order)

(T_L T_R R)



Example / Optimization Question

- Consider a problem of maximizing the function:

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

Where x is permitted to vary between 0 and 31.

- Objective function / Fitness function: $f(x) = x^2$
- Initial Population:
  - Here, initial population of size 4 is chosen (but any number of population can be selected) using binary encoding.
  - 5 bits will be required.
  - Let the initial population be:

Chromosome No.	Initial Population	Value
1	0 1 1 0 0	12
2	1 1 0 0 1	25
3	0 0 1 0 1	5
4	1 0 0 1 1	19

Selection / Reproduction

- Selection is the process of choosing two parents from the population for crossing.
- The purpose of selection is to emphasize fitter individuals in the population in hope of finding offspring with higher fitness.
- There are two types of selection scheme
  - Proportionate based selection
    - Picks out individual based on their fitness values relative to the fitness of other individuals in the population.
  - Ordinal based selection
    - Selects individual not upon their raw fitness but upon their rank within the population.
  - Types
    - Roulette wheel selection
    - Boltzmann selection
    - Tournament selection
    - Rank selection
    - Steady state selection

Selection / Reproduction

Roulette wheel selection

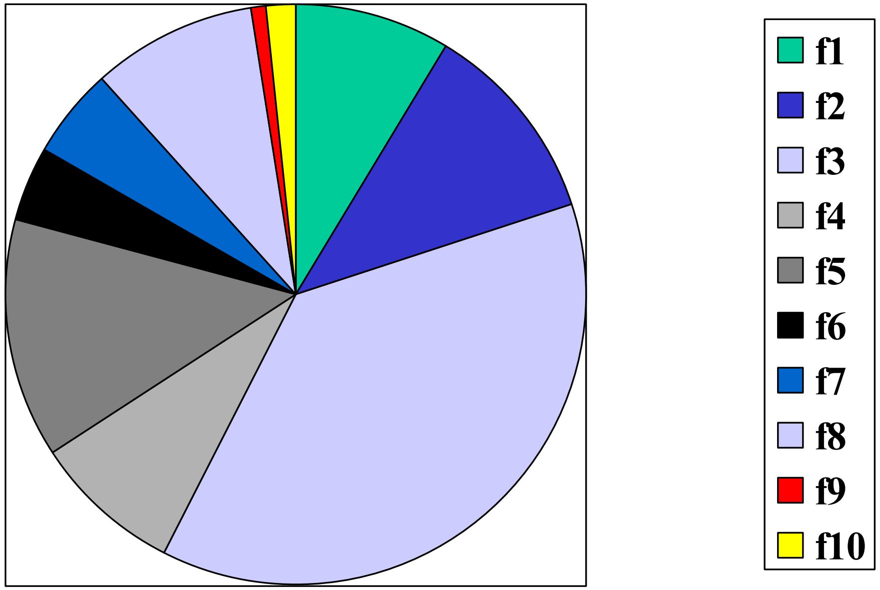
- A string is selected from the population with a probability proportional to the fitness.
- Suppose there are *P* individuals with fitnesses $f_1, f_2, \ldots, f_P$; *and* higher values mean better fitness.
- The probability of selecting individual *i* is simply:

$$\frac{f_i}{\sum_{k=1}^{P} f_k}$$

- This type of selection is similar to using a roulette wheel where the fitness of an individual is represented as proportionate slice of wheel. The wheel is then spun and the slice underneath the wheel when it stops determine which individual becomes a parent.

Selection / Reproduction

Roulette wheel selection



$$\frac{f_i}{\sum_{k=1}^{P} f_k}$$

- Average fitness

$$\bar{F} = \sum_{k=1}^{n} F_k / n$$

Example / Optimization Question

- Consider a problem of maximizing the function:

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

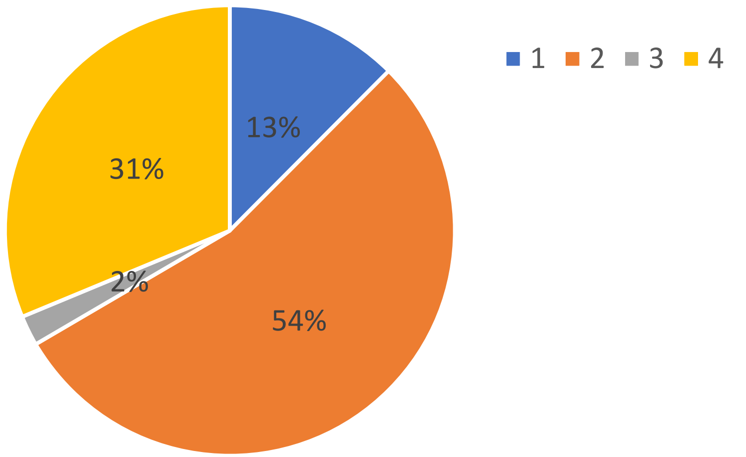
Where x is permitted to vary between 0 and 31.

- Objective function / Fitness function: $f(x) = x^2$
- Initial Population
- Selection: Calculate fitness

Chromosome No.	Initial Population	Value	Fitness	Selection Prob (P_i)	% P_i
1	0 1 1 0 0	12	144	0.1247	12.47
2	1 1 0 0 1	25	625	0.5411	54.11
3	0 0 1 0 1	5	25	0.0216	2.16
4	1 0 0 1 1	19	361	0.3126	31.26
Sum			1155		

Example / Optimization Question

Chromosome No.	Initial Population	Value	Fitness	Selection Prob (P_i)	% P_i	Expected Count (Fitness/Average)	Actual Count
1	0 1 1 0 0	12	144	0.1247	12.47	0.4987	0
2	1 1 0 0 1	25	625	0.5411	54.11	2.1645	2
3	0 0 1 0 1	5	25	0.0216	2.16	0.0866	0
4	1 0 0 1 1	19	361	0.3126	31.26	1.2502	1
Sum			1155				
Average			288.75				



Selection / Reproduction

Boltzmann selection

- Simulated Annealing (SA) is a method that simulates the process of slow cooling of molten metal to achieve the minimum function value in a minimization problem.
- Controlling a temperature like parameter introduced with the concept of Boltzmann probability distribution simulates the cooling phenomenon.
- 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. Else, it is selected with Bolt/Mann probability as:

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

- Where $T = T_0(1-\alpha)^k$ and $k = (1 + 100 * \frac{g}{G})$; g is the current generation number; G the maximum value of g.
- The α can be chosen from the range [0, 1] and T_0 from the range [5, 100].
- The final state is reached when the computation approaches zero value of *T*.

Selection / Reproduction

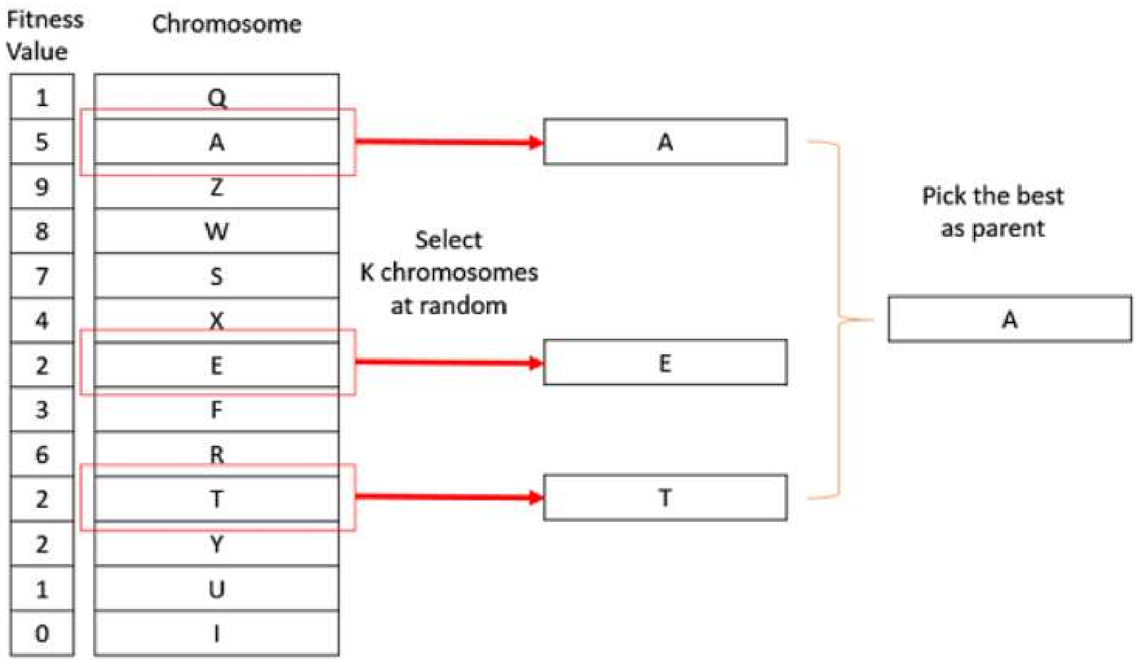
Tournament selection

- Individuals in the mating pool are the ones whose genes will be inherited by the next generation, it is desirable that the mating pool consists of good individuals.
- *Population diversity* means that the genes from the already discovered good individuals are exploited while promising the new areas of the search space continue to be explored.
- *Selective pressure* is the degree to which the better individuals are favoured.
- The higher the selective pressure the more, the better individuals are favoured.
- The selective pressure drives GA to improve population fitness over succeeding generations.
- If the selective pressure is too high there is an increased chance of GA prematurely converging to local optimal solution.
- If the selective pressure is too low, the convergence rate will be slow and the GA will take unnecessarily long time to find the optimal solution.
- Unlike the Roulette wheel selection, the tournament selection strategy provides selective pressure by holding a tournament competition among *N* individuals.

Selection / Reproduction

Tournament selection

- The best individual (winner) from the tournament is the one with the highest fitness.
- Tournament competitors and winners will be inserted in the mating pool.
- The tournament competition is repeated until the mating pool for generating new offspring is filled.



Selection / Reproduction

Rank selection

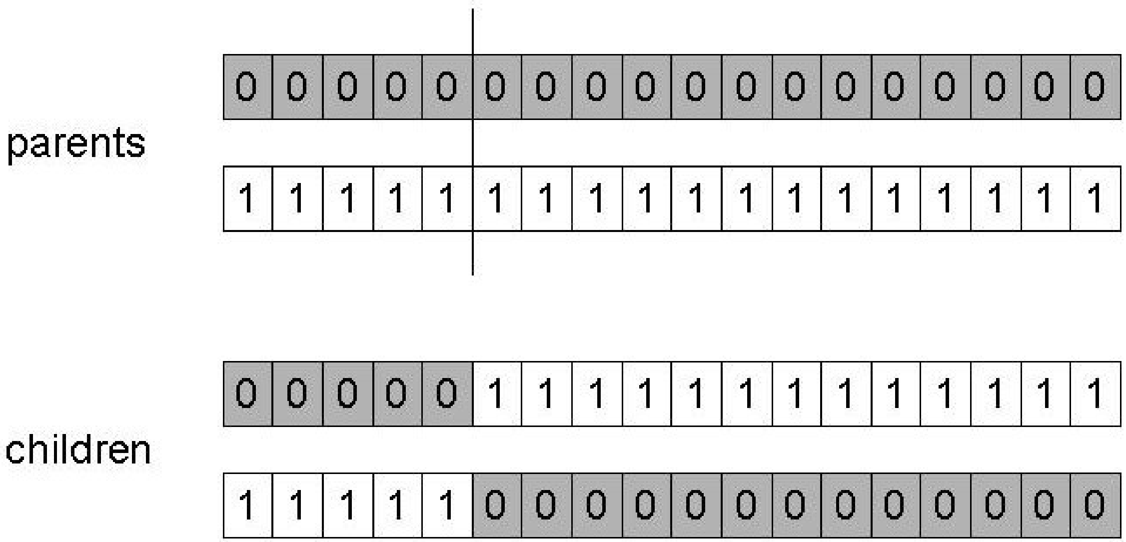
- Rank selection first ranks the population and every chromosome receives fitness from the ranking.
- The worst will have fitness 1 and so on.
- The best will have fitness N.
- Then Roulette wheel selection is applied to the modified wheel.

Steady state selection

- Good individuals with higher fitness are selected for creating new offsprings.
- Then, some bad ones are removed and the new offsprings replace them.
- The rest of the population survives the new generation.

Crossover

- Crossover operator is applied to the mating pool to create better offcrpings.
- The aim of the crossover operator is to search the parameter space.
- Steps:
  - Select a pair of two individuals
  - Select a cross site
  - Swap the position values at the cross site.
- **Single site cross over**



Crossover

- **Single site cross over**

Before Crossover

Parent 1 :	0	1	1	0	0	0	1	0
Parent 2 :	1	0	1	0	1	1	0	0

Crossover Point - k

Select crossover points randomly

Offspring 1:	0	1	1	0	1	1	0	0
Offspring 2:	1	0	1	0	0	0	1	0

After Crossver

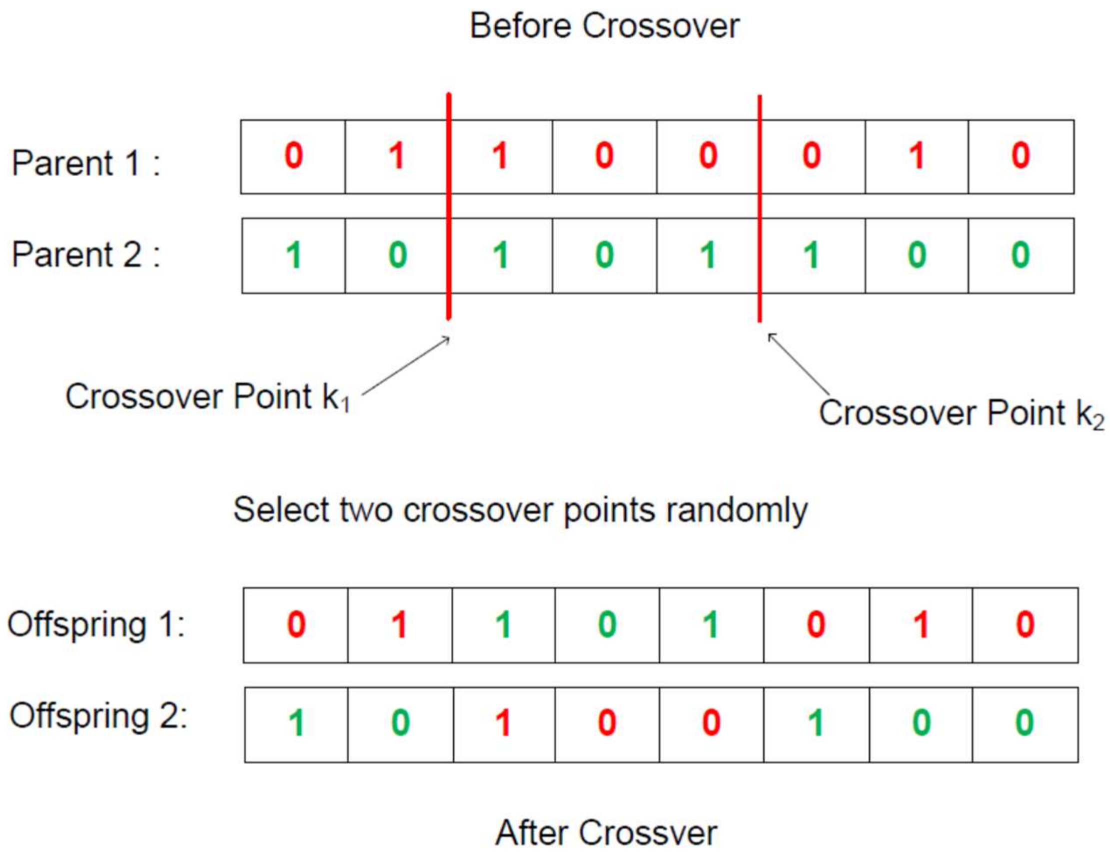
Crossover

Chromosome No.	Initial Population	Value	Fitness	Selection Prob (P_i)	% P_i	Expected Count (Fitness/Average)	Actual Count
1	0 1 1 0 0	12	144	0.1247	12.47	0.4987	0
2	1 1 0 0 1	25	625	0.5411	54.11	2.1645	2
3	0 0 1 0 1	5	25	0.0216	2.16	0.0866	0
4	1 0 0 1 1	19	361	0.3126	31.26	1.2502	1
Sum			1155				
Average			288.75				

Chromosome No.	Mating pool	Cross over point	Offspring after crossover	*X value*	*Fitness*
1	0 1 1 0 0	4	0 1 1 0 1	13	169
2	1 1 0 0 1	4	1 1 0 0 0	24	576
3	1 1 0 0 1	2	1 1 0 1 1	27	729
4	1 0 0 1 1	2	1 0 0 0 1	17	289

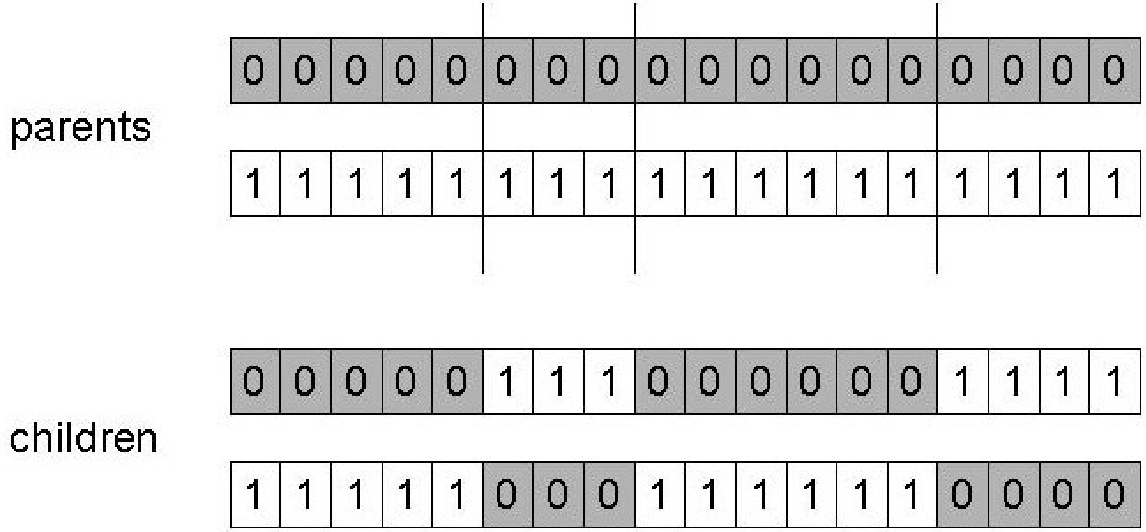
Crossover

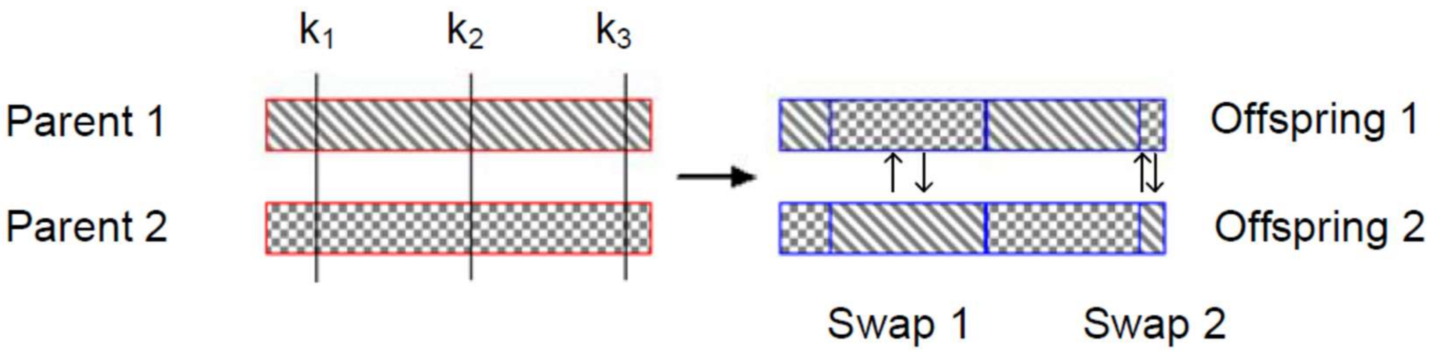
- **Two point site cross over**



Crossover

- **Multi point crossover**





Crossover

- **Uniform cross over**

parents	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
children	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1
	1	0	1	1	0	0	0	0	1	1	1	0	1	0	0	1	1	0

Crossover

- **Uniform cross over (with mask)**

Before Crossover

Parent 1 :	1	1	0	0	0	1	0	1
Parent 2 :	0	1	1	0	0	1	1	1
Mask	1	0	0	1	1	1	0	1
Offspring 1:	1	1	1	0	0	1	1	1
Offspring 2:	0	1	0	0	0	1	0	1

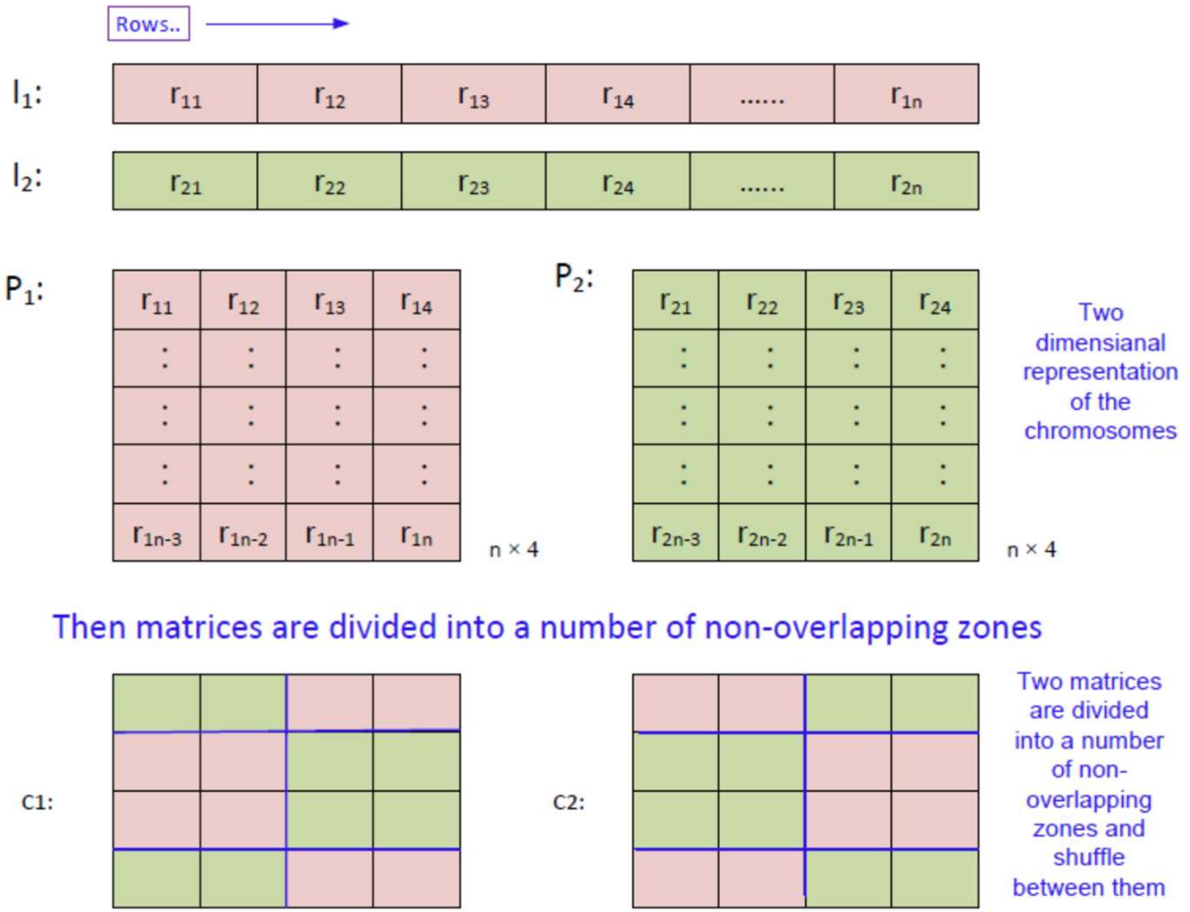
Offspring 1: When there is a 1 in the mask, the gene is copied from Parent 1 else from Parent 2.

Offspring 2: When there is a 1 in the mask, the gene is copied from Parent 2 else from Parent 1.

After Crossver

Crossover

- **Matrix cross over**



Crossover

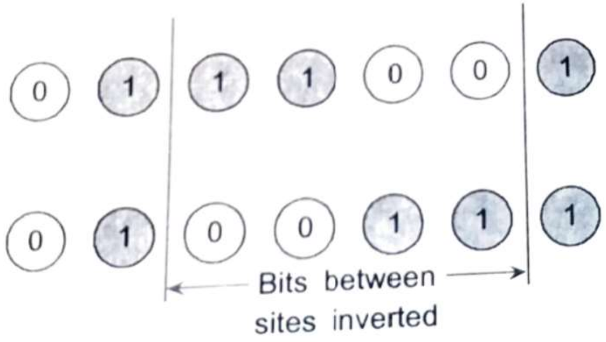
Cross over rate

- The term cross over rate is denoted as P_C , the probability of cross over.
- It varies from 0 to 1.
- Ratio of the number of pairs crossed to some fixed population.
- Typically for a population size of 30 to 200, cross over rates are ranged from 0.5 to 1.
- When a crossover probability of P_C is used only $100P_C$ percent strings in the population are used in the crossover operation and $100(1-P_C)$ percentage of the population remains as it is in the current population.

Inversion and Deletion

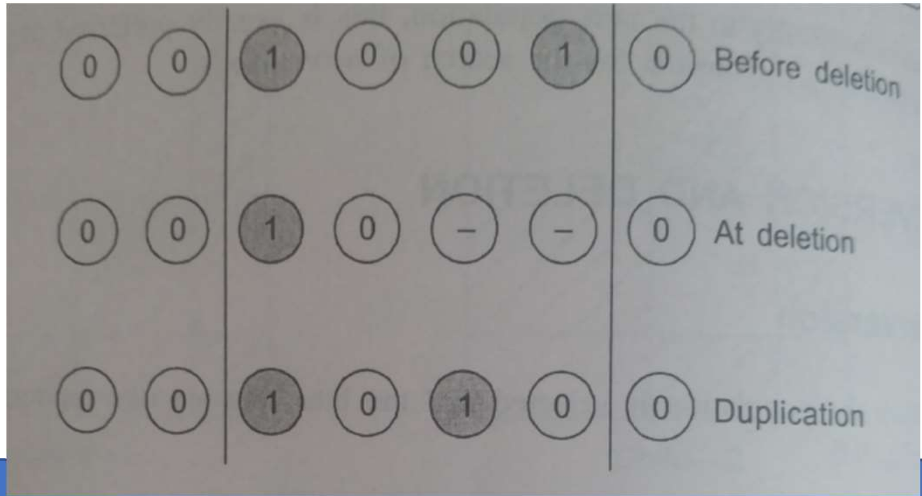
Inversion

- A string from the population is selected and the bits between two random sites are inverted.



Deletion and Duplication

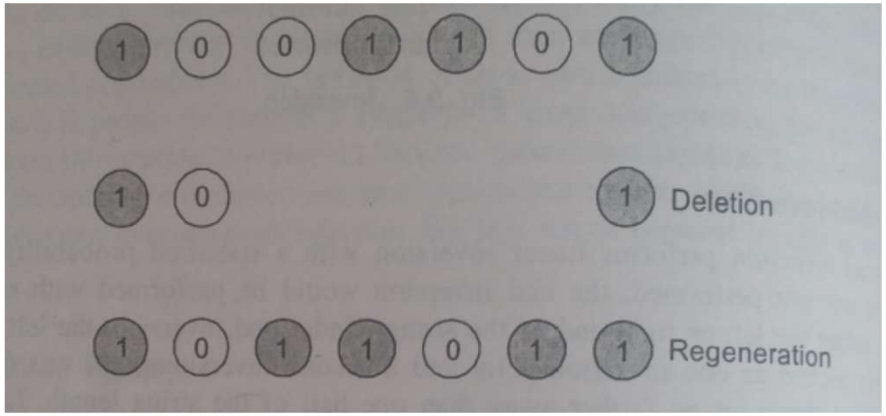
- Any two or three bits at random are selected and the previous bits are duplicated.



Inversion and Deletion

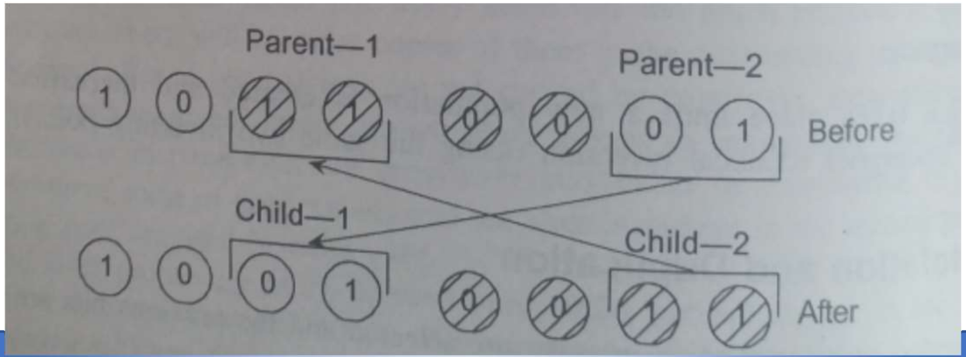
Deletion and Regeneration

- Genes between two cross sites are deleted and regenerated randomly.



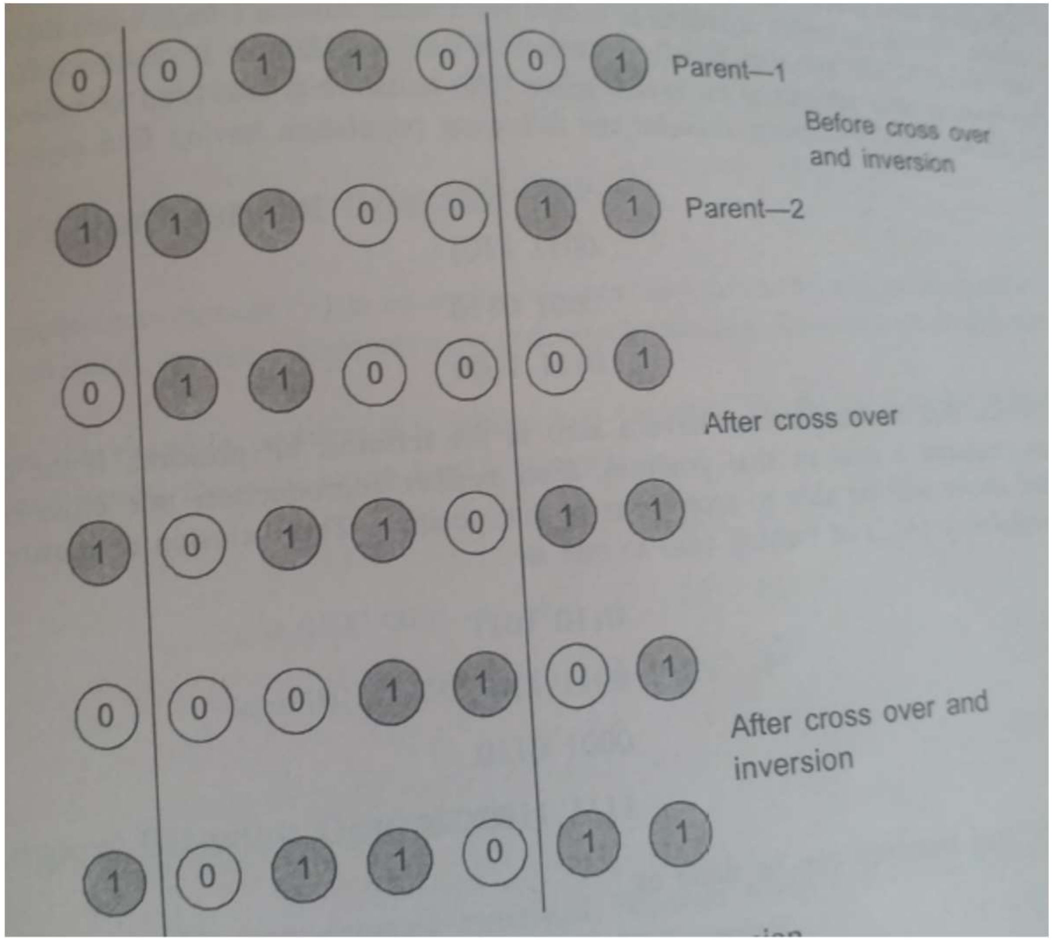
Segregation

- The bits of parents are segregated and then crossed over to produce offsprings.



Inversion and Deletion

Cross over and inversion



Mutation

- After crossover, the strings are subjected to mutation.
- Mutation involves flipping a bit, i.e. changing 0 to 1 and vice versa with a small mutation probability P_m.
- A number between 0 to 1 is chosen at random.
- If the random number is smaller than P_m then the bit is altered otherwise it remains unchanged.
- The mutation of the bit does not affect the probability of mutation of the other bits.
- The mutation operator introduces new genetic structures in the population.
- Typically, a genetic algorithm uses the population size of 30 to 200 with the mutation rates varying from 0.001 to 0.5.

Mutation

Chromosome No.	Mating pool	Cross over point	Offspring after crossover	*Mutation chromosomes*	*X value*	*Fitness*
1	0 1 1 0 0	4	**0** 1 1 0 1	1 1 1 0 1	29	841
2	1 1 0 0 1	4	1 1 0 0 0	1 1 0 0 0	24	576
3	1 1 0 0 1	2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 1 1	2	1 0 **0** 0 1	1 0 1 0 1	21	441

Bitwise Operator

- One's complement operator
  - Unary operator that causes the bits of its operand to be inverted.
- Logical Bitwise operators
  - Binary operator
  - Bit-wise AND (&) operator
  - Bit-wise OR (|) operator
  - Bit-wise exclusive – OR (^) operator
- Shift operators
  - Shift left (<<)
  - Shift right (>>)
  - Masking

Convergence of GA

- As GA progresses, there may not be much improvement in the new population generated.
- The population gets filled with more fit individuals and the average fitness comes very close to the individual fitness.
- The GA iteration can also stop after some fixed number of generations.