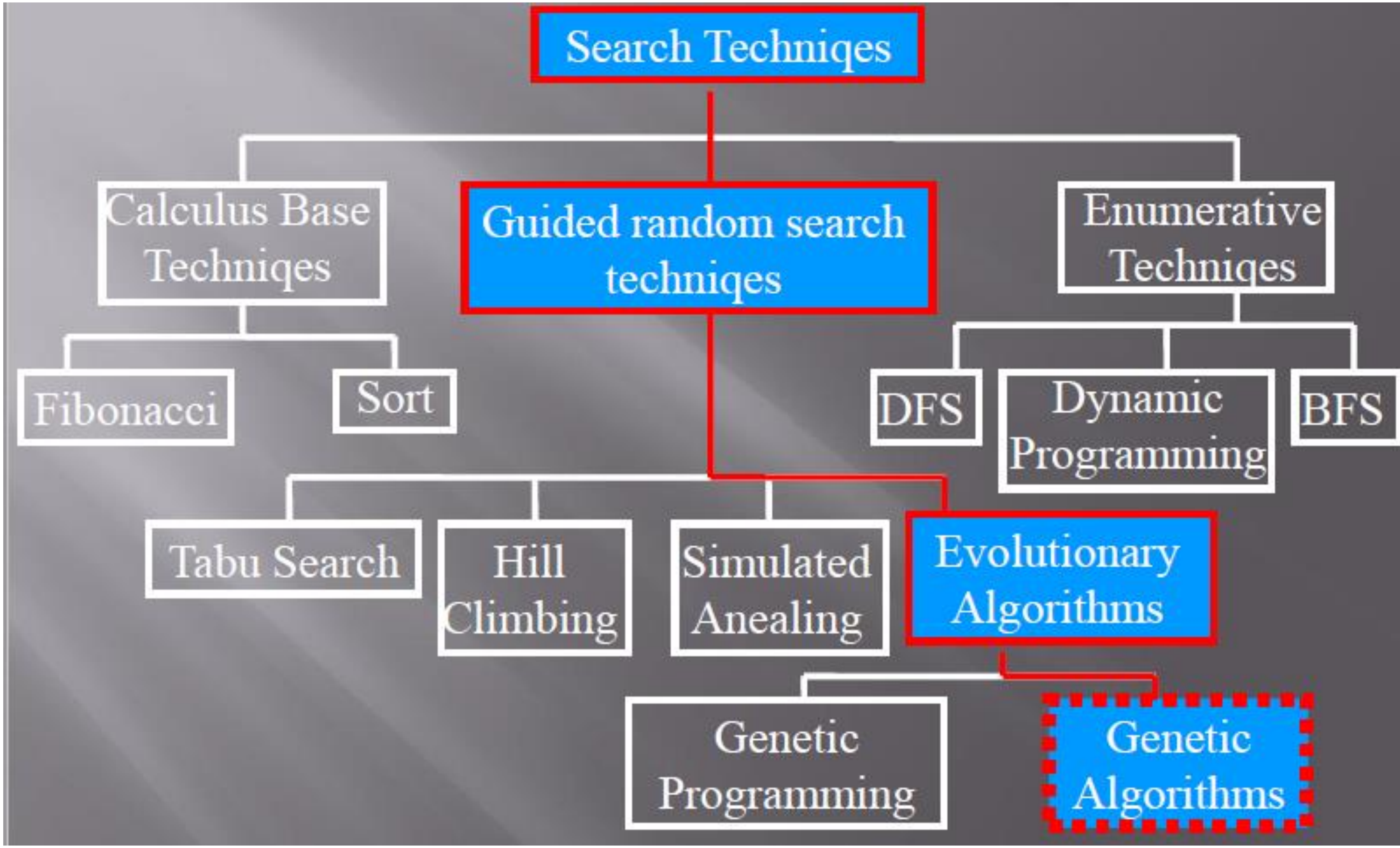
Genetic Algorithm

Classes of Search Techniques



Introduction

- Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection
- (GA)s are categorized as global search heuristics
- (GA)s are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination)
- (GA)s are inspired by Darwin's theory about evolution-"survival of the fittest"

Introduction

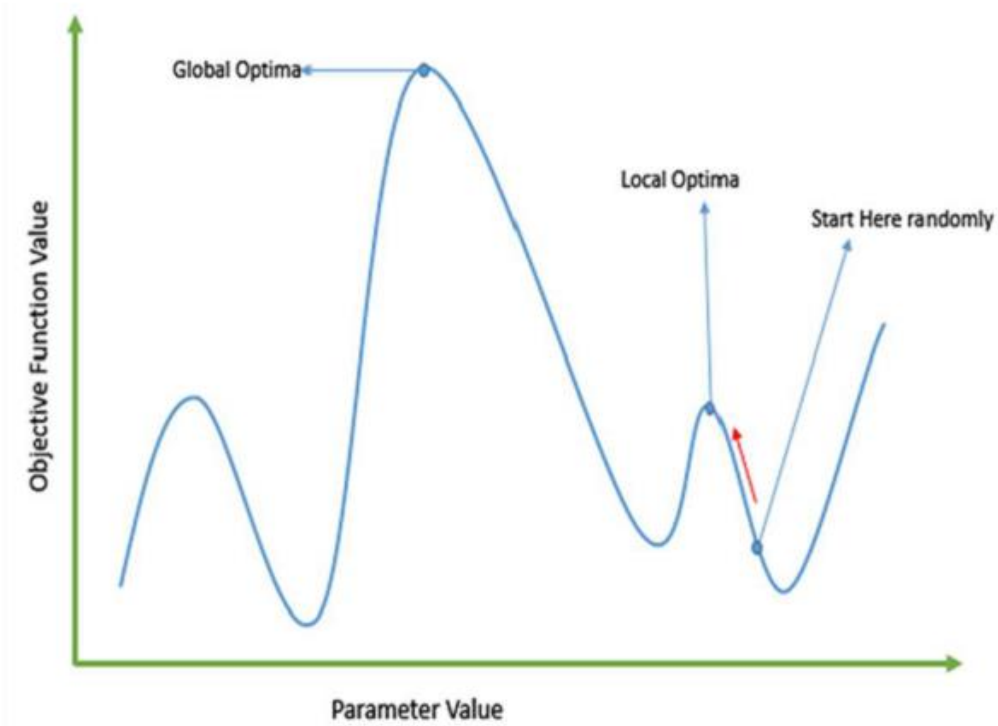
- GAs were developed by John Holland and his students and colleagues at the University of Michigan, most notably David E. Goldberg and has since been tried on various optimization problems with a high degree of success
- In GAs, we have a pool or a population of possible solutions to the given problem. These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations
- Each individual (or candidate solution) is assigned a fitness value (based on its objective function value) and the fitter individuals are given a higher chance to mate and yield more "fitter" individuals

Introduction

- In this way we keep "evolving" better individuals or solutions over generations, till we reach a stopping criterion
- Genetic Algorithms are sufficiently randomized in nature, but they perform much better than random local search (in which we just try various random solutions, keeping track of the best so far), as they exploit historical information as well

GA-Motivation

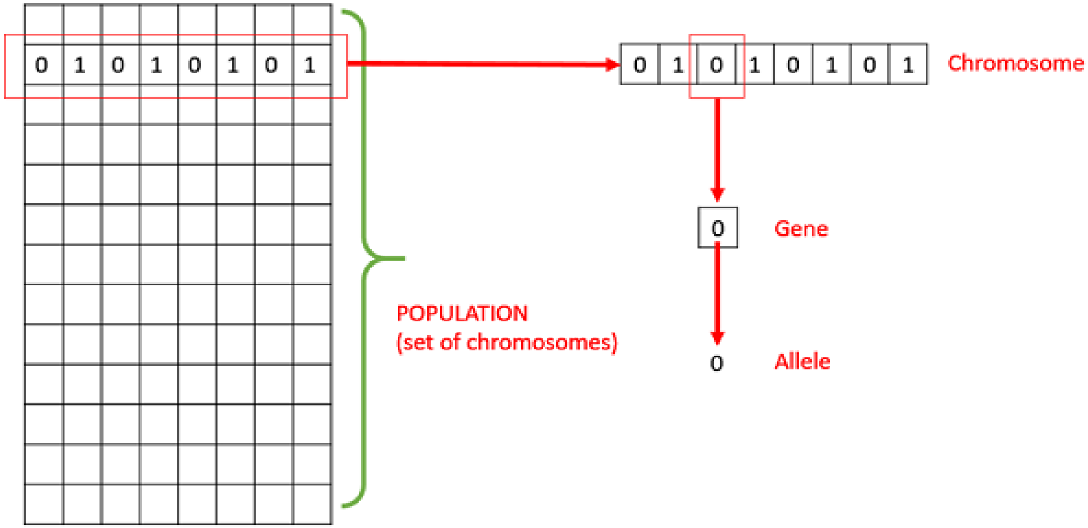
- Solving Difficult problems
- Failure of Gradient Based Methods



- Getting a Good solution Fast

Basic Terminology

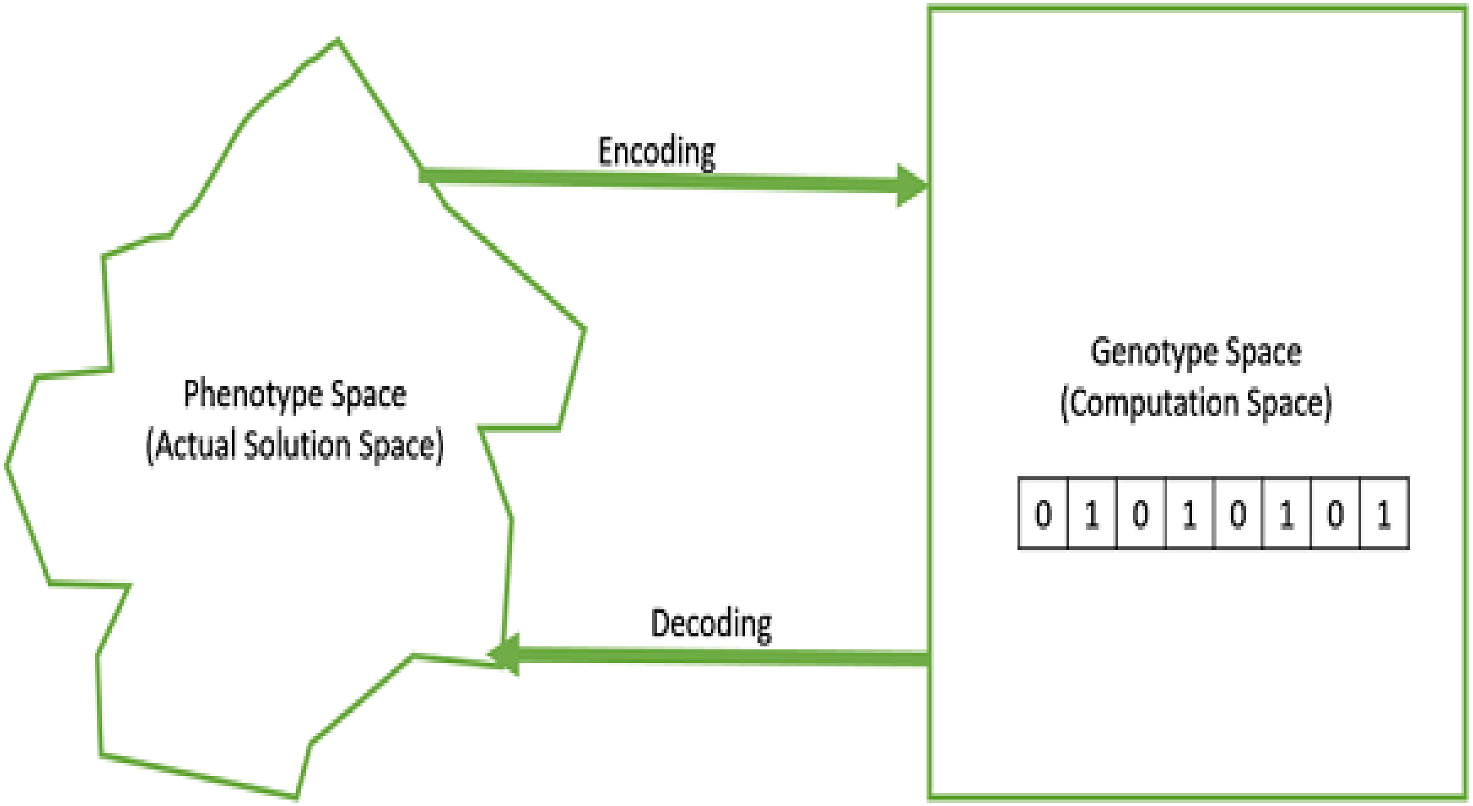
- Population – It is a subset of all the possible (encoded) solutions to the given problem
- Chromosomes – A chromosome is one such solution to the given problem
- Gene – A gene is one element position of a chromosome
- Allele – It is the value a gene takes for a particular chromosome



Basic Terminology

- **Genotype – Genotype is the population in the computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system**
- **Phenotype – Phenotype is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations**
- **Decoding and Encoding – Decoding is a process of transforming a solution from the genotype to the phenotype space, while encoding is a process of transforming from the phenotype to genotype space. Decoding should be fast as it is carried out repeatedly in a GA during the fitness value calculation.**

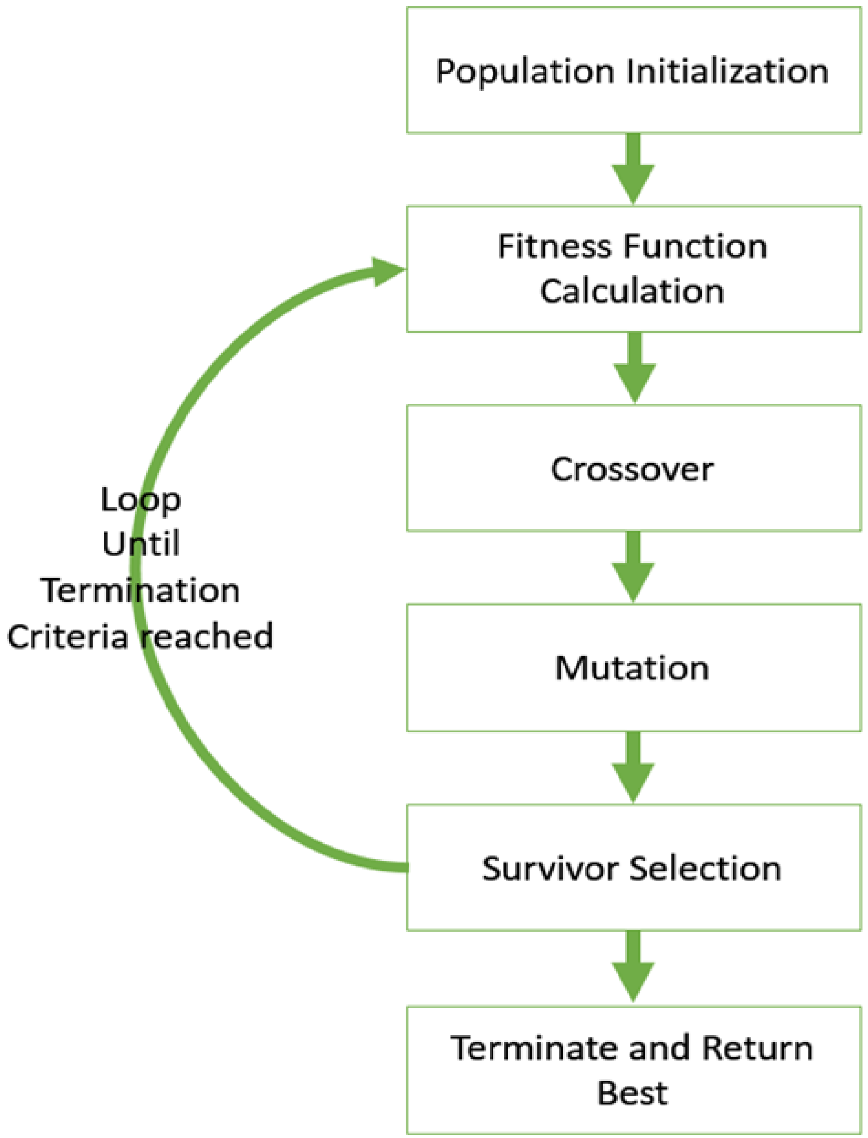
Basic Terminology



Basic Terminology

- **Fitness Function − A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output. In some cases, the fitness function and the objective function may be the same, while in others it might be different based on the problem**
- **Genetic Operators − These alter the genetic composition of the offspring. These include selection, crossover, mutation**

Basic Structure of GA



Population Initialization

- There are two primary methods to initialize a population in a GA
  - Random Initialization – Populate the initial population with completely random solutions.
  - Heuristic initialization – Populate the initial population using a known heuristic for the problem.

Fitness

- A fitness score is given to each individual which shows the " ability of an individual to compete"
- Individual having better fitness score are given more chance to reproduce than others
- Individuals with better fitness scores are selected who mate and produce better offspring by combining chromosomes of parents

GA Operators

- ❑ **Methods of representation**
- ❑ **Methods of selection**
- ❑ **Methods of Reproduction**

Methods of representation

- **Encode solutions as binary strings: sequences of 1's and 0's, where the digit at each position represents the value of some aspect of the solution.**
- **Second approach is encode solutions as arrays of integers or decimal numbers.**
- **A third approach is to represent individuals in a GA as strings of letters, where each letter again stands for a specific aspect of the solution.**

Methods of Selection

- There are many different techniques which a genetic algorithm can use to select the individuals to be copied over into the next generation
  - Roulette-wheel selection.
  - Elitist selection.
  - Fitness-proportionate selection.
  - Scaling selection.
  - Rank selection.
  - Generational selection.
  - Hierarchical selection

Roulette-wheel selection

- ❑ **Roulette Wheel selection is used for generating chromosomes for the next generation. This is a way of choosing members from the population of chromosomes in a way that is proportional to their fitness.**
- ❑ **It does not guarantee that the fittest member goes through to the next generation merely that it has a very good chance of doing so**

➢ How it works?

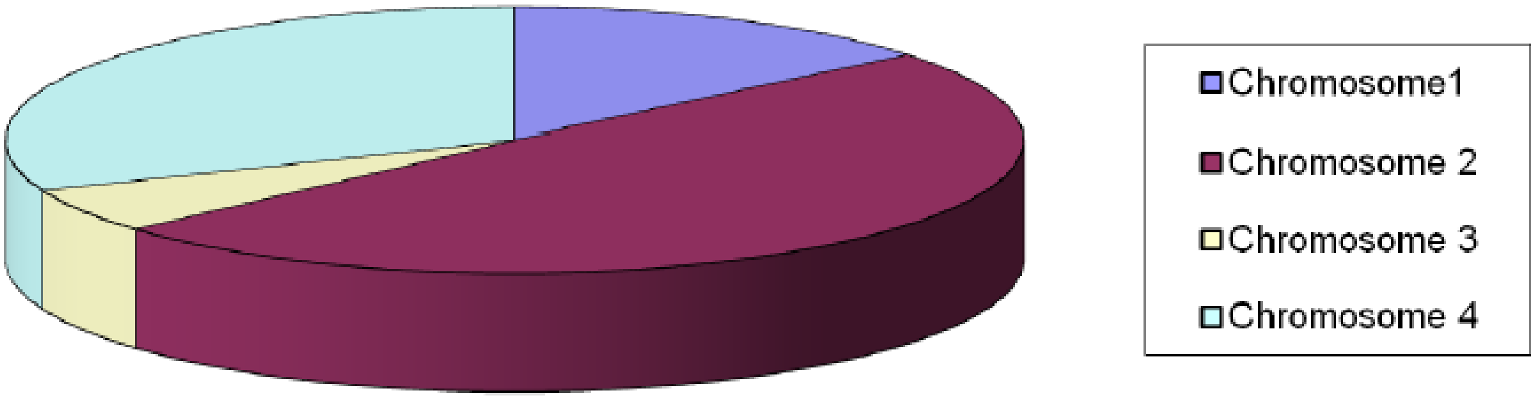
- ❑ **Imagine that the population’s total fitness score is represented by a roulette wheel.**
- ❑ **Now you assign a slice of the wheel to each member of the population. The size of the slice is proportional to that chromosomes fitness score (i.e. the fitter a member is the bigger the slice of pie it gets).**

Roulette-wheel selection

- Now, to choose a chromosome all you have to do is spin the wheel and grab the chromosome at the point it stops

No.	String	Fitness	% Of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

Roulette-wheel selection



- Let the points at which it has stopped are: 2, 3, 4 and 1
- Then the four chromosomes are selected based on the fitness probability and these four chromosomes are called the mating pool

Another methods of selection

- **Elitist selection:**

- The most fit members of each generation are guaranteed to be selected.

- **Rank selection:**

- Each individual in the population is assigned a numerical rank based on fitness, and selection is based on this ranking.

Methods of Reproduction

- Once selection has chosen fit individuals, they must be randomly altered in hopes of improving their fitness for the next generation
- There are two basic strategies to accomplish this:
  - Crossover
  - Mutation

Crossover

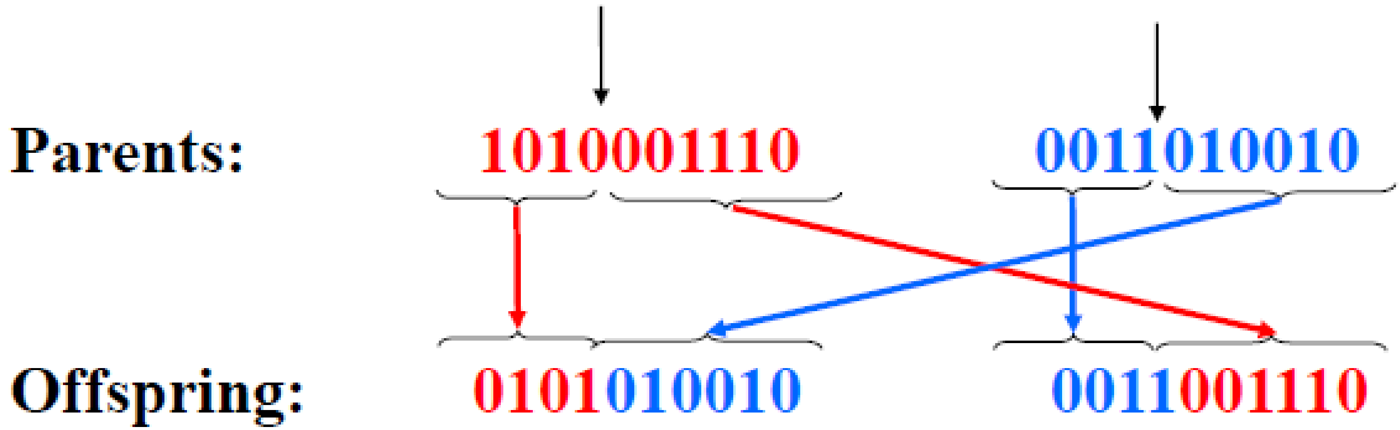
- Two parents produce two offspring
- There is a chance that the chromosomes of the two parents are copied unmodified as offspring
- There is a chance that the chromosomes of the two parents are randomly recombined (crossover) to form offspring

Types

- Single point crossover
- Two point crossover (Multi point crossover)
- Uniform crossover

Single point crossover

- Randomly one position in the chromosomes is chosen
- Child 1 is head of chromosome of parent 1 with tail of chromosome of parent 2
- Child 2 is head of 2 with tail of 1



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 produce 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

- In Uniform Crossover, a probability, p, is used to determine whether a given bit from parent 1 will be used, or from parent 2
- In other words, a child can receive any random bits from each of its parents
- Uniform Crossover operator decides with some probability known as the mixing ratio
- Consider the two parents

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

Uniform Crossover

- **If the mixing ratio is 0.5 approximately then half of the genes in the offspring will come from parent1 and other half come from parent2**

Offspring 1	1_1	1_2	0_2	1_1	1_1	1_2	1_2	0_2	0_1	0_1	0_2	1_1	1_2	1_1	1_1	0_2
Offspring 2	1_2	1_1	0_1	1_2	1_2	0_1	0_1	1_1	0_2	0_2	1_1	1_2	0_1	1_2	1_2	0_1

Mutation

- Mutation is a genetic operator used to maintain genetic diversity from one generation of population of chromosomes to the next

- Mutation occurs during evolution according to a user-defined mutation probability, usually set to fairly low value, say 0.01. For example, with a mutation rate of 0.01, it might be expected that one gene in a chromosome of 100 genes might be reversed

- Mutation simply involves reversing the value of a bit in a chromosome. 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

- Mutation helps to prevent the population from stagnating at any local optima
- 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

Termination Criteria

- **Two ways in which a run of a genetic algorithm is terminated**
  - **Limit is put on the number of generations, after which the run is considered to have finished.**
  - **Run can stop when a particular solution has been reached, or when the highest fitness level in the population has reached a particular value**