Genetic Algorithms

What is GA?

- A class of probabilistic optimization algorithms
- Inspired by the biological evolution process
- Uses concepts of “Natural Selection” and “Genetic Inheritance” (Darwin 1859)
- Originally developed by John Holland (1975)
- Widely-used in business, science and engineering

- 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.

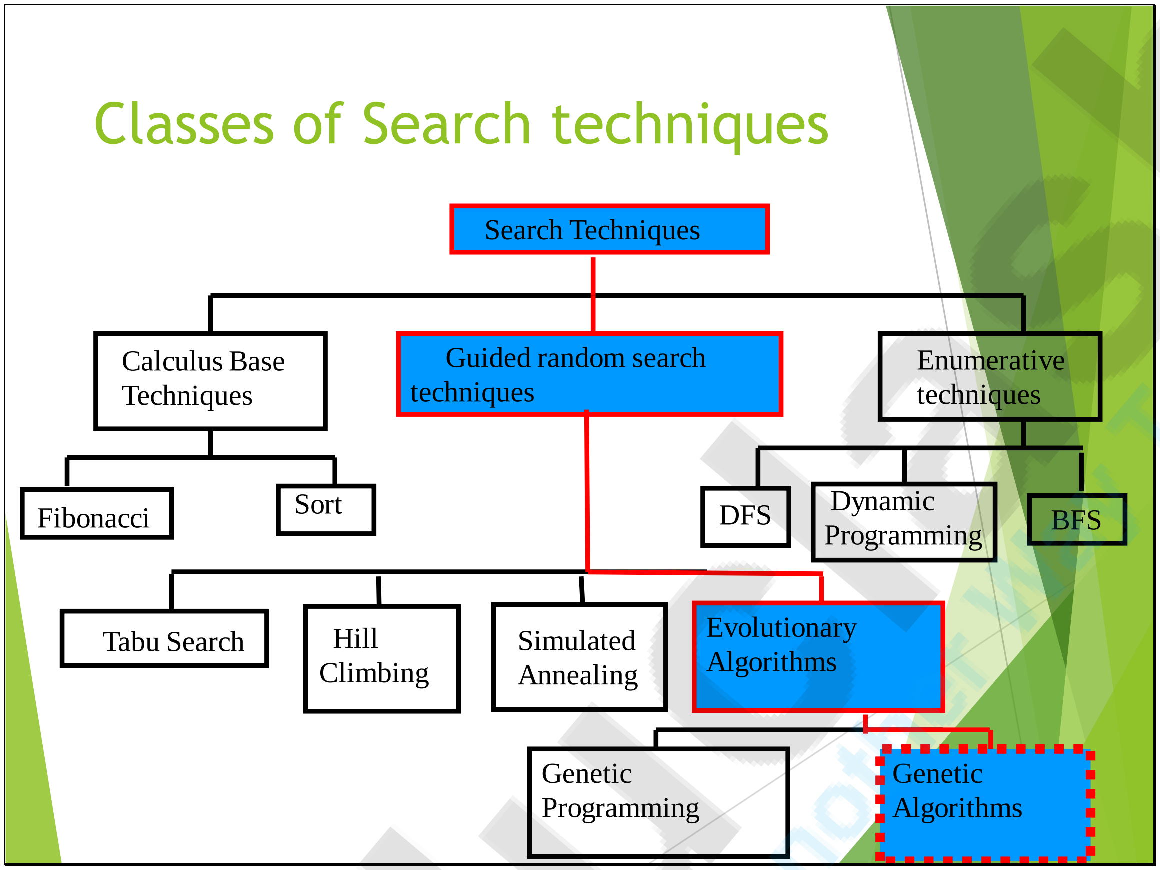
- The evolution usually starts from a population of randomly generated individuals and happens in generations.

- 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

Why GA

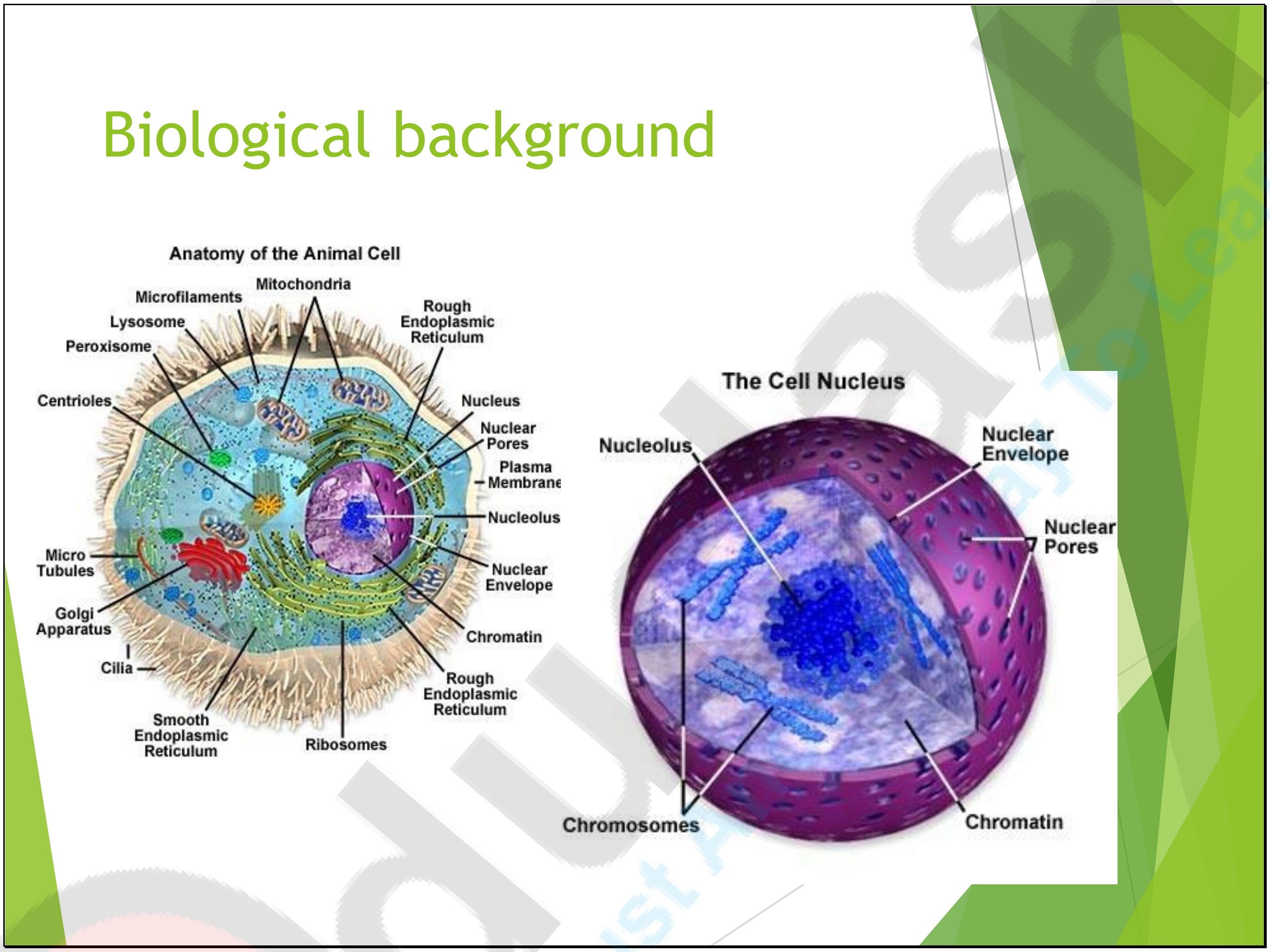
- Better than conventional algorithms
- More robust
- They do not break easily even if inputs are changed slightly or in the presence of reasonable noise
- In searching n- dimensional surface, GA offer significant benefits over more typical optimization techniques

- Classes of Search techniques



- Traditional Search Techniques
  - Gradient- Based Local Optimization Method
  - Random search
  - Stochastic Hill Climbing
  - Simulated Annealing
  - Symbolic Artificial Intelligence

Biological background



- **The cell:** Genetic information is contained in the cell nucleus
- **Chromosomes**
  - All the generic information gets stored in the chromosomes
  - It is build of DNA
  - Chromosomes are divided into several parts called genes
- **Genetics**
  - ***Genotype:***– Particular set of genes
  - ***Phenotype:***– Physical characteristic of the genotype (smart, beautiful, healthy, etc)

- **Natural selection:** Plays major role in survival process

Key terms

- **Individual** - Any possible solution
- **Population** - Group of all *individuals*
- **Search Space -** All possible solutions to the problem
- **Chromosome** - Blueprint for an *individual*
- **Trait** - Possible aspect (*features)* of an *individual*
- **Allele** - Possible settings of trait (black, blond, etc.)
- **Locus** - The position of a *gene* on the *chromosome*
- **Genome** - Collection of all *chromosomes* for an *individual*

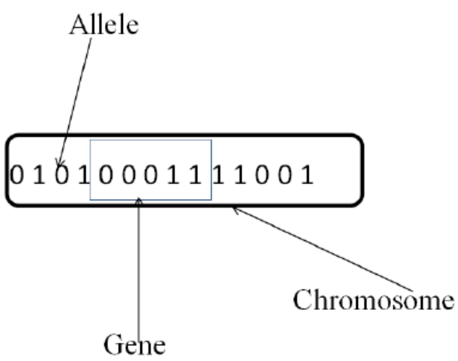
Genetic Algorithm vs Traditional Algorithm

- GA's work with coded versions of the problem parameters rather than parameters themselves, i.e., GA works with the coding of solution set and not with the solution itself
- GA's work with population of points instead of single point
- Usually, the size of the population is in the range from 20 to 200 or 300 and majority of traditional optimization methods explores 1, 2, or 3 points in the search space on each iteration
- In GA's previously found good information is emphasized using reproduction operator and propagated adaptively through crossover and mutation operators
- GA does not require any auxiliary information except the objective function values
- GA uses the probabilities in their operators while conventional methods for continuous optimization apply deterministic transition operates i.e. GA does not use deterministic rules
- GA uses fitness function for evaluation rather than derivatives
- They can be applied to any kind of continuous or discrete optimization problem

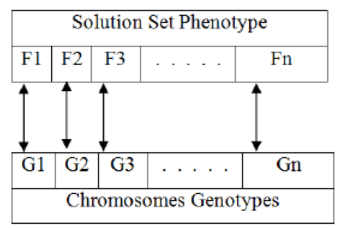
Basic Terminologies in GA

Individuals

- It is a single solution
- It groups together two forms of solutions
  - Chromosome
  - Phenotype
- Chromosome should contain information about the solution that it represents
- Morphogenesis function associates each genotype with its phenotype
- Each chromosome must define one unique solution
- But each solution can be encoded by more than one chromosome
- All the candidate solutions of the problem must correspond to at least one possible chromosome
- When different chromosomes can encode the same solution, the representation is said to be degenerated
- A slight degeneracy is not so worrying
- But too important degeneracy could be a more serious problem
- It can badly affect behavior of the GA



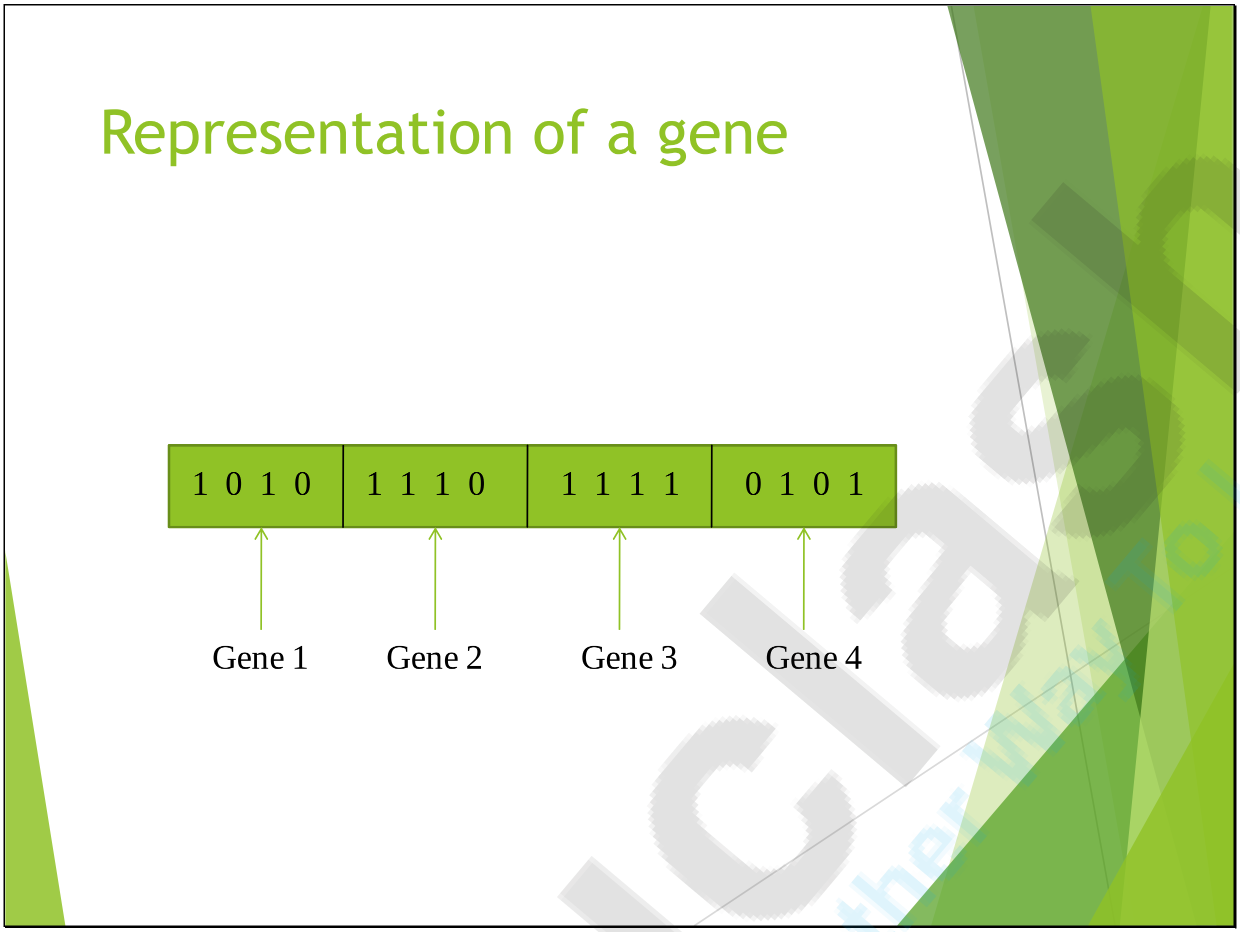
Representation of a chromosome



Representation of a genotype and phenotype

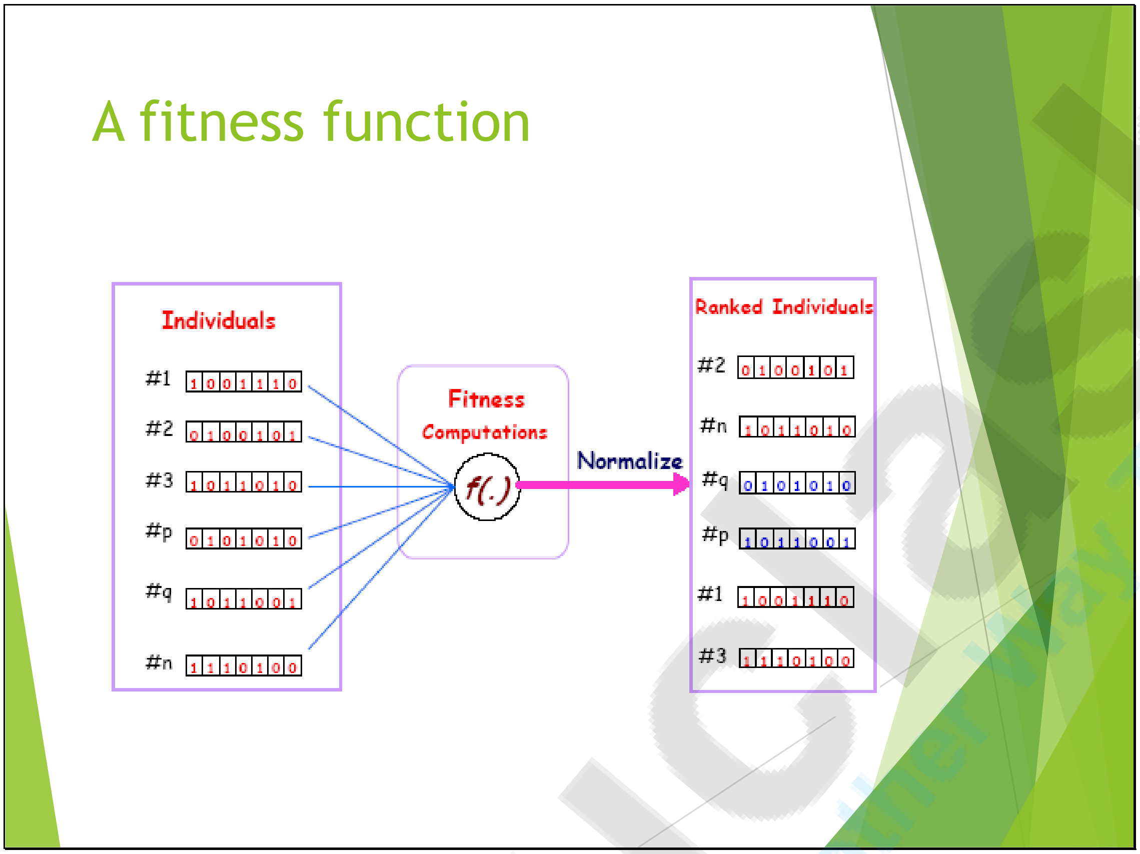
Genes

- Genes are the basic instructions for building a GA
- A chromosome is a sequence of genes
- A gene is a bit string of arbitrary lengths
- It is the GA's representation of a single factor for a control factor
- Each factor in the solution set corresponds to a gene in the chromosome
- Control factor must have an upper bound and a lower bound
- A bit string of length n can represent $(2^n - 1)$ intervals
- The structure of each gene is defined in a record of phenotype parameters
- Phenotype parameters are instructions for mapping between genotype and phenotype
- This mapping is necessary to convert solution sets from model into form that GA can work with (i.e. encoding solution set into chromosome) and for converting new individuals from GA into a form that model can evaluate(i.e. decoding chromosome to a solution set)

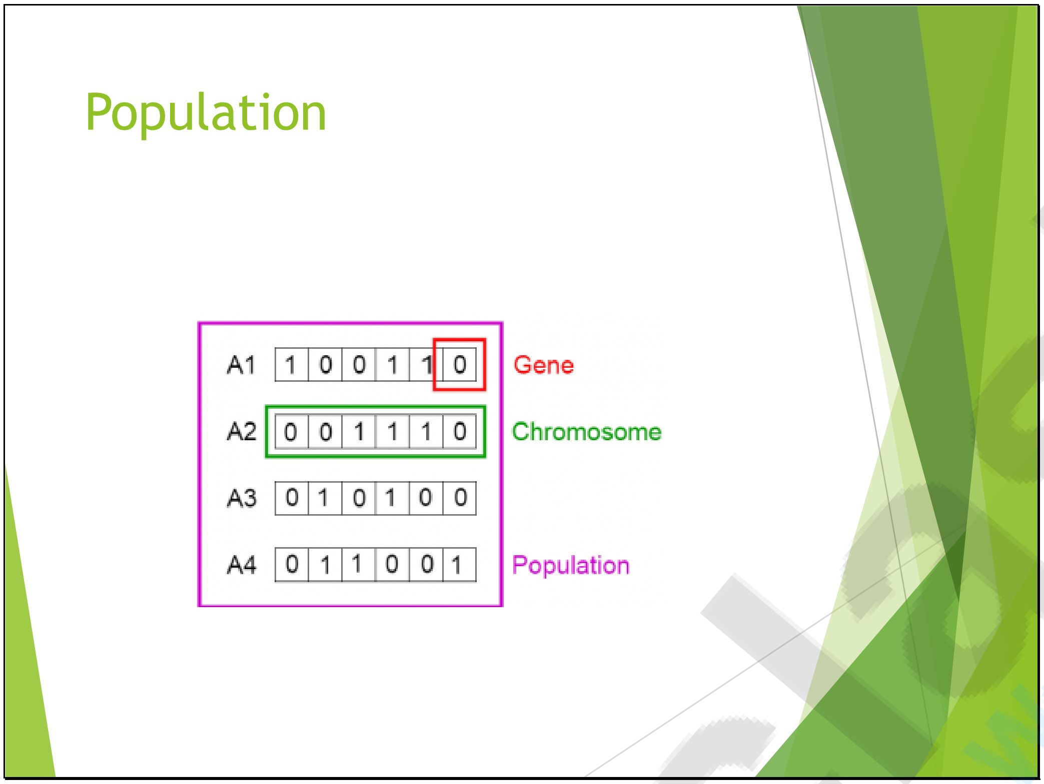


Fitness

- Fitness of an individual in a GA is the value of an objective function for its phenotype
- For calculating fitness,
  - The chromosome has to be first decoded
  - The objective function has to be evaluated
- Fitness indicates
  - How good the solution is
  - How close the chromosome is to the optimal one
- In the case of multicriteria optimization, the fitness function is definitely difficult to determine
- In this case, there is often a dilemma as how to determine if one solution is better than another
- Sometimes a fitness function is obtained by a simple combination of the different criteria if it gives good result
- For more advanced problems, some ideas from multicriteria optimization theory can be used



- Populations
  - It is a collection of individuals
  - It consists of
    - A number of individuals being tested
    - The phenotype parameters defining the individuals
    - Some information about search space
  - Two important aspects of population in GA
    - The initial population generation
    - The population size



- **Population initialization**

  - In most of the cases, population is initialized randomly
  - However there may be instances where the initialization of population is carried out with some known good solutions
  - While initializing the population two important points should be considered to find global optimal solutions
    - Population should have a gene pool as large as possible in order to be able to explore the whole search space
    - Mean fitness of the population should be high

- **Population size**

- Population size will depend on the complexity of the program
- Population size raises few problems
- The larger the population is, the easier it is to explore the search space
- A large population is quite useful
- However it requires much more computational cost memory and time
- Practically, a population size of around 100 individuals is quite frequent
- Goldberg has shown that GA efficiency to reach global optimum instead of local ones is largely determined by the size of the population

Working principle of genetic algorithm

- In the genetic algorithm a solution, i.e., a point in the search space is represented by a finite sequence of zeros and ones, called a chromosome
- First, a population of strings representing the decision variables is randomly generated
- The size of the population depends on the string length and the problem being optimized
- Each string is then evaluated using the objective function value i.e. fitness
- Once the strings are evaluated, the three genetic operators reproduction, crossover and mutation are applied to create a new population
- Reproduction, crossover and mutation operators are applied repeatedly until global optimum solution is found

- Pseudo-code of working of Gas

Initialize a population of strings at random
Evaluate each string in the population
Repeat
Reproduction
Crossover
Mutation
Evaluation of the population
Until (termination criterion)

Reproduction

- The selection of individuals to produce successive generations plays an important role in GA
- Reproduction comprises forming a new population, usually with the same total number of chromosome, by selection from members of the current population, following a particular scheme
- The higher the fitness, the more likely it is that the chromosome will be selected for the next generation
- There are several strategies for selecting the individuals
- (e.g) roulette wheel selection, ranking methods and tournament selection
- After the reproduction phase is over, the population is enriched with good strings
- Reproduction makes copies of good strings, but does not create any new string

Crossover

- A cross over operator is used to recombine with the hope of creating a better string
- An overall crossover probability is assigned to the crossover process, which is the probability that given two parents, the crossover process will occur
- This probability is often in the range of 0.6 to 0.9

Mutation

- Mutation acts as a background operator
- It is used to search the unexplored search space by randomly changing the values at one or more positions of the selected chromosome

Stopping criteria

- These are the conditions under which the search process will terminate

- **Maximum generations:** GA stops when the specified number of generations has evolved
- **Elapsed time**: Genetic process will end when a specified time has elapsed
- **No change in fitness**: GA stops if there is no change to the population's best fitness for specified number of generations
- **Stall generations**: The algorithm stops if there is no improvement in the objective function for a sequence of consecutive generations of length Stall generations
- **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
- **Best individual**: A best individual convergence criterion stops the search once the minimum fitness in the population drops below the convergence value
- **Worst Individual**: Worst individual terminates the search when the least fit individuals in the population have fitness less than the convergence criteria
- **Sum of fitness**: in this scheme, the search is considered to have satisfaction converged when sum of the fitness in the entire population is less than or equal to the convergence value in the population
- **Median fitness**: 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