MACHINE INTELLIGENCE

Introduction to Genetic Algorithms

Dr. Arti Arya

Department of Computer Science and Engineering

MACHINE INTELLIGENCE

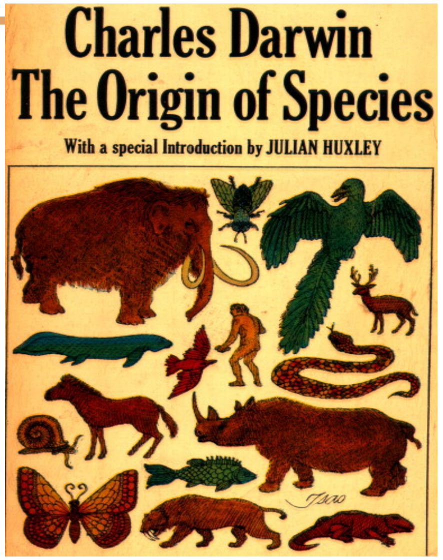
Introduction to Genetic Algorithms

Dr. Arti Arya

Department of Computer Science and Engineering

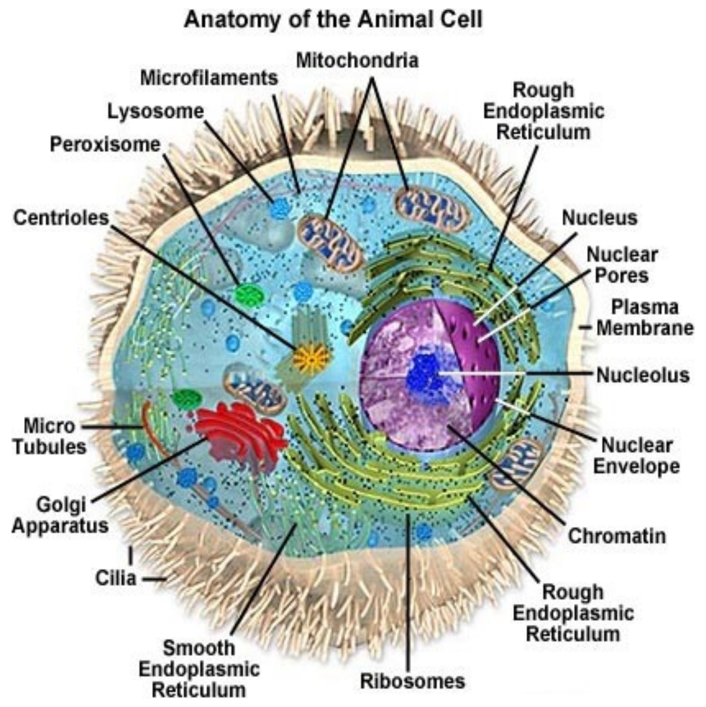
Genetic Algorithms- Introduction

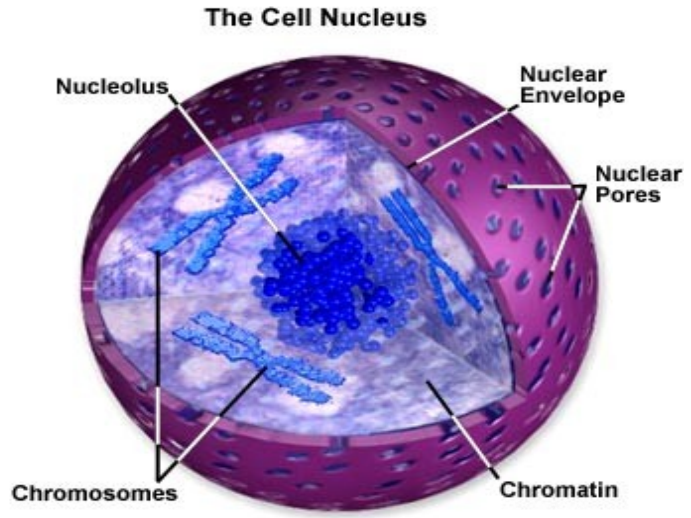
- Genetic Algorithm (GA) is a metaheuristic search-based optimization technique inspired by the principles of **Genetics and Natural Selection** *(Darwin's theory of evolution: survival of the fittest).*
- Idea was introduced by applying to problem solving by Professor John Holland in 1965.
- It is frequently used to find *optimal or near-optimal solutions* to difficult problems which otherwise would take a lifetime to solve.
- Multiple points in the solution space are simultaneously considered for optimality, ie global perspective of the solution space. This avoids local optima.
- For detailed information you can further refer to [2].



Genetic Algorithms- Biological Background

- The nucleus contains the genetic information.





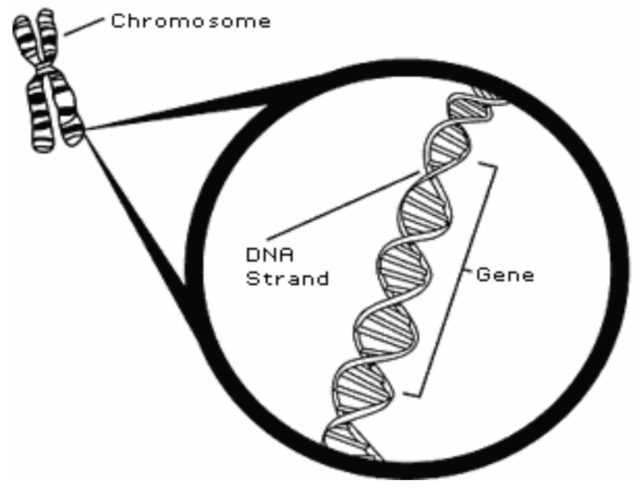


Figure from “ Soft and Evolutionary Computing” by N P Padhy.

Connection Of GA with Optimization

- The set of all possible solutions or values of the problem, make up the search space.

- This search space has a point or a set of points which gives the optimal solution.

- The aim of optimization is to find that point or set of points in the search space and *so is the objective of GAs.*

- GA use only objective function information, not derivatives or other auxiliary knowledge.

- Thus GA can deal with the non-smooth, non-continuous and non-differentiable functions which actually exist in a practical optimization problem.

Genetic Algorithms- Motivation

- Because Genetic Algorithms tend to find globally optimal solutions[1], which makes genetic algorithms attractive for solving optimization problems.

- **Need of GAs**

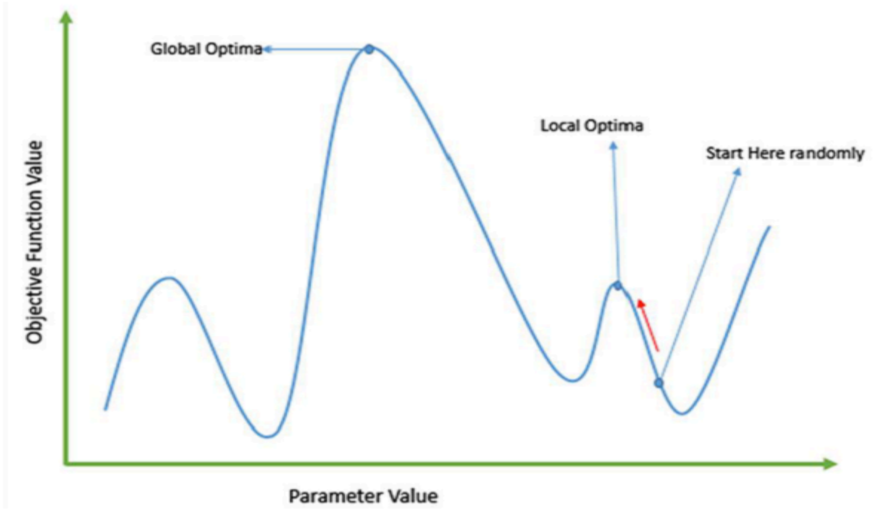
  1. Solving Difficult Problems

  - In computer science, there is a large set of problems, which are **NP-Hard**. For solving such problems, even the most powerful computing systems take a very long time (even years!) to solve that problem.

Genetic Algorithms- Motivation

- **Failure of Gradient Based Methods**

- Traditional techniques efficient and works very well for single-peaked (unimodal) objective functions like the cost function in linear regression.
- More complex problems having multi-peaked (multimodal)objective functions come across, which causes such methods to fail, as they suffer from an inherent tendency of getting stuck at the local optima.



Genetic Algorithms- Motivation

- **Getting a Good Solution Fast**

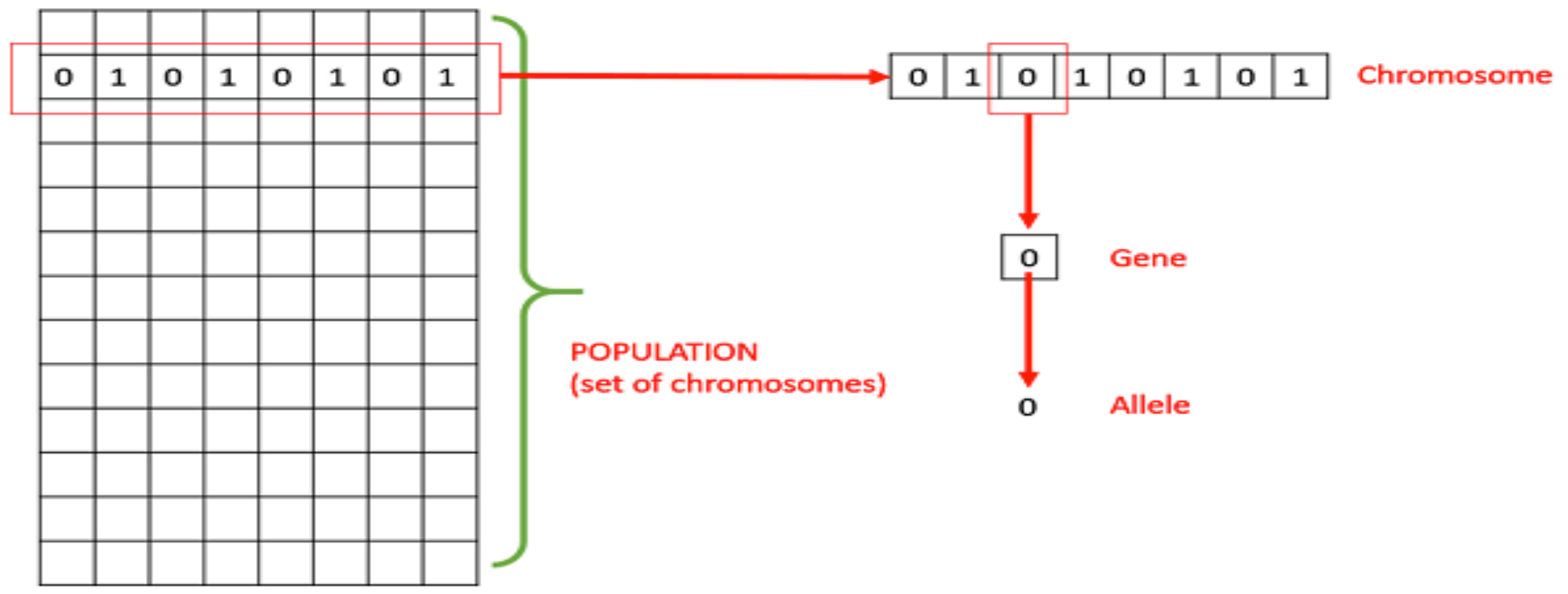
Some difficult problems like the Travelling Salesperson Problem (TSP), have real-world applications like path finding and VLSI Design.

MACHINE INTELLIGENCE

GA- Basic Terminologies

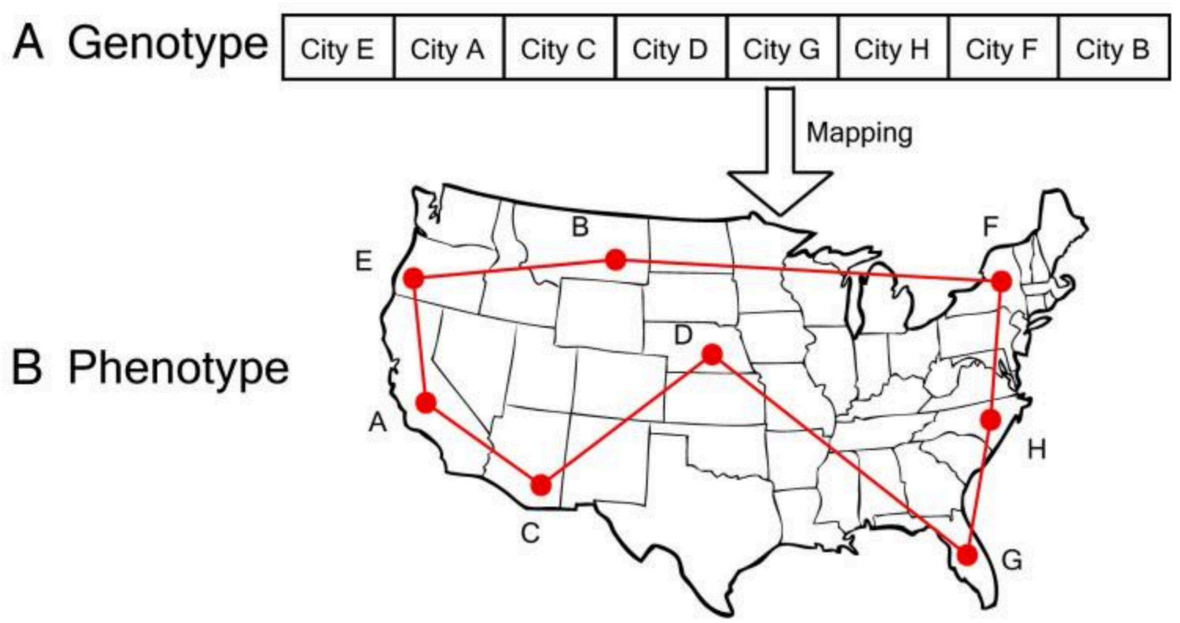
•**Population** – Subset of all the possible (encoded) solutions.
•**Chromosomes** – A chromosome is one such solution to the given problem.

•**Gene** – A gene is one element position of a chromosome.
•**Allele** – It is the value a gene takes for a particular chromosome.



GA- Basic Terminologies

- **Genotype** – The set of genes representing the chromosome in computation space.
- **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.



GA- Basic Terminologies

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.

- **Fitness Function –** A fitness function takes the solution as input and produces the suitability of the solution as the output.

- **Genetic Operators** – These alter the genetic composition of the offspring. These include crossover, mutation, selection, etc.

GA- A generalized pseudo-code

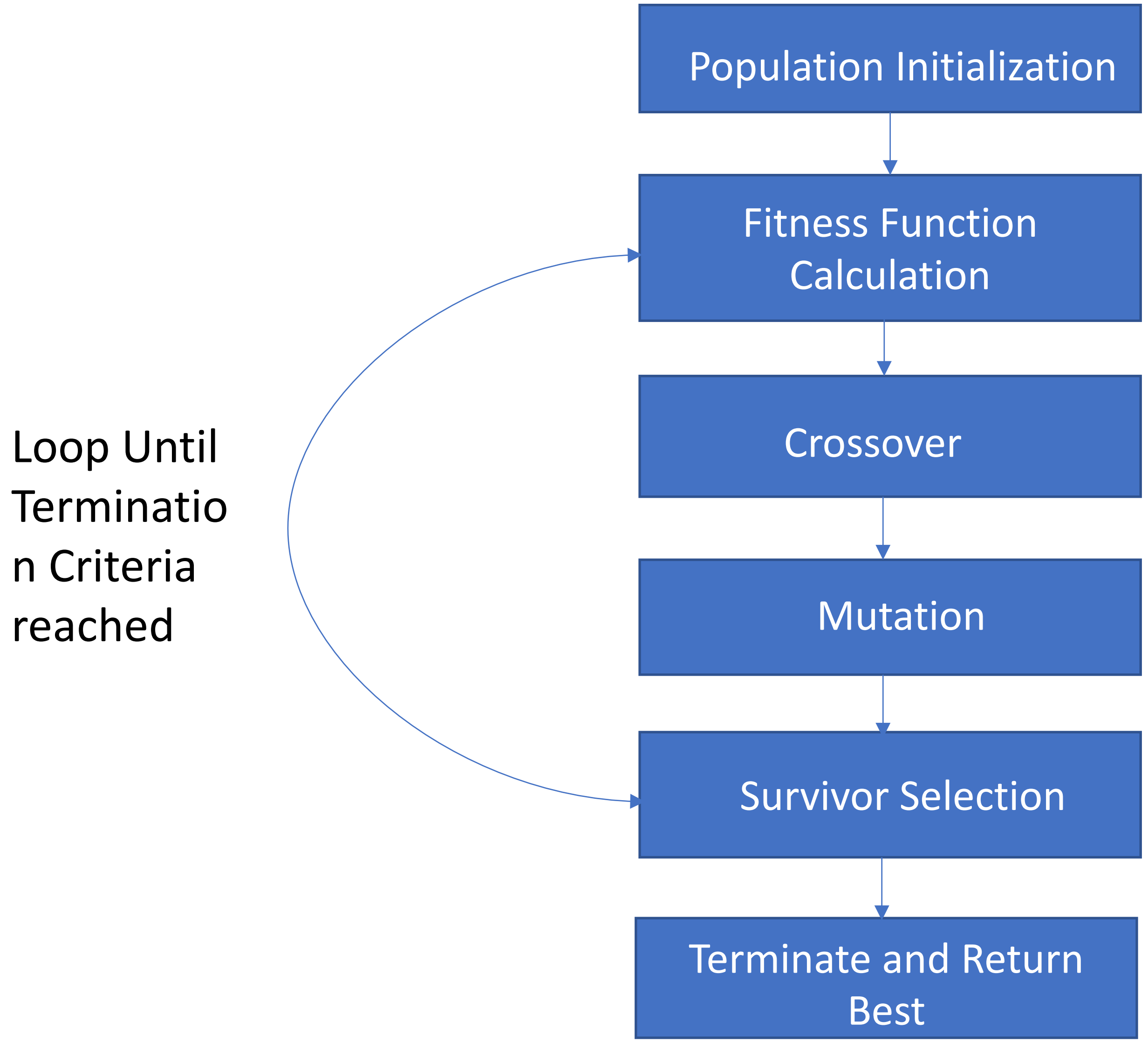
```
GA()
      initialize population
      find fitness of population

while (termination criteria is reached) do
      parent selection
      crossover with probability pc
      mutation with probability pm
      decode and fitness calculation
      survivor selection
      find best
return best
```

Steps in GA……

- In GAs,
  - A pool or a population of possible solutions to the given problem is considered.
  - These solutions then undergo recombination and mutation (like in natural genetics), producing new offsprings.
  - And the process is repeated over various generations.
  - Each individual (or candidate solution) is assigned a fitness value and the fitter individuals will get a higher chance to mate and yield more "fitter" individuals.
  - In this way solutions "evolving" over generations, till a stopping criterion is met.

GA- Basic Structure


Population Initialization
Fitness Function Calculation
Crossover
Mutation
Survivor Selection
Terminate and Return Best
Loop Until Terminatio n Criteria reached

Genetic Representation(Encoding)

- *Encoding* is a process of representing chromosomes and thus genes.
- GAs algorithms do not deal with the solutions directly but with encoded representation of it.
- Chromosomal representation of a problem can be done by any of the following ways:
  - Binary string representation
  - Real number coding
  - Integer coding
  - Permutation Representation
  - Random key Representation
  - Representing genes in arrays, trees, lists and other objects

Hypotheses Representation in GA

- Most of the time hypotheses in GAs are represented by bit strings.
- Bit strings are used for encoding bcoz it becomes easy to use genetic operators like crossover and mutation.
- Just as a simple example how bit strings would be used:
- Consider the classical example of “ Play Tennis” where Outlook takes 3 values ‘rainy’, ’sunny’ and ‘overcast’.
- Say Outlook attribute is represented by a binary string of length 3 then
    - 100 represents Outlook has value “rainy”.
    - 110 represents Outlook can acquire “rainy or sunny”.

Hypotheses Representation in GA

- Consider a rule's precondition

  (Outlook= Overcast ∨ Rain)∧(wind=strong)

- This can be encoded as

  101 10 (How?)

Consider a rule:

If. Wind=Strong Then PlayTennis=yes

Encoding:

111 10 10 (Pls Explain how?)

Genetic Representation-Examples

Binary Encoding

The equivalent value for any n-bit string can be obtained by

$$X_i = X_i^l + \frac{X_i^u - X_i^l}{(2^{n_i} - 1)} \times (decoded\ value\ of\ string)$$

Each variable has both the upper and lower limit which can be put in the form as

$$X_i^l \le X_i \le X_i^u$$

An 'n'-bit string can be represented by an integer between 0 to 2^n - 1, which consists of 2^n integers

n = 4, X_i^l=4 , X_i^u=25, $string$ =1010

Decoded value of string = 1010 = 2^3 x 1 + 2^2 x 0 + 2^1 x 1 + 2^0 x 0 = 10

Genetic Representation-Examples

The decoded value of 4 bit string is

$$X_i = 4 + \frac{21}{15} * 10 = 18$$

If 'p' is the precision required for a continuous variable then the string length 'S_l' should be equal to

$$S_i = \log_2 \left[\frac{X^u - X^l}{p} \right]$$

Example : Knapsack problem [4]

The problem: There are things with given value and size. The knapsack has a given capacity. Select things to maximize the value of things in knapsack, but do not extend knapsack capacity.

Encoding: Each bit says, if the corresponding thing is in knapsack.

Genetic Representation

Real Valued Representation

For problems where we want to define the genes using *continuous* rather than discrete variables, the real valued representation is the most natural[3]. The precision of these real valued or floating point numbers is however limited to the computer.

0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9

Integer Representation

For discrete valued genes, we cannot always limit the *solution space to binary 'yes' or 'no'.* For example, if we want to encode the four distances – North, South, East and West, we can encode them as **{1,2,3,4}**. In such cases, integer representation is desirable.

1	2	3	4	3	2	4	1	2	1

Genetic Representation

Permutation representation (*Path Representation* or *Order representation)*

For example, a tour of 8-city Travelling Salesman Problem(TSP)

3-5-8-1-4-2-6-7 can be represented as [3 5 8 1 4 2 6 7]

Random keys representation

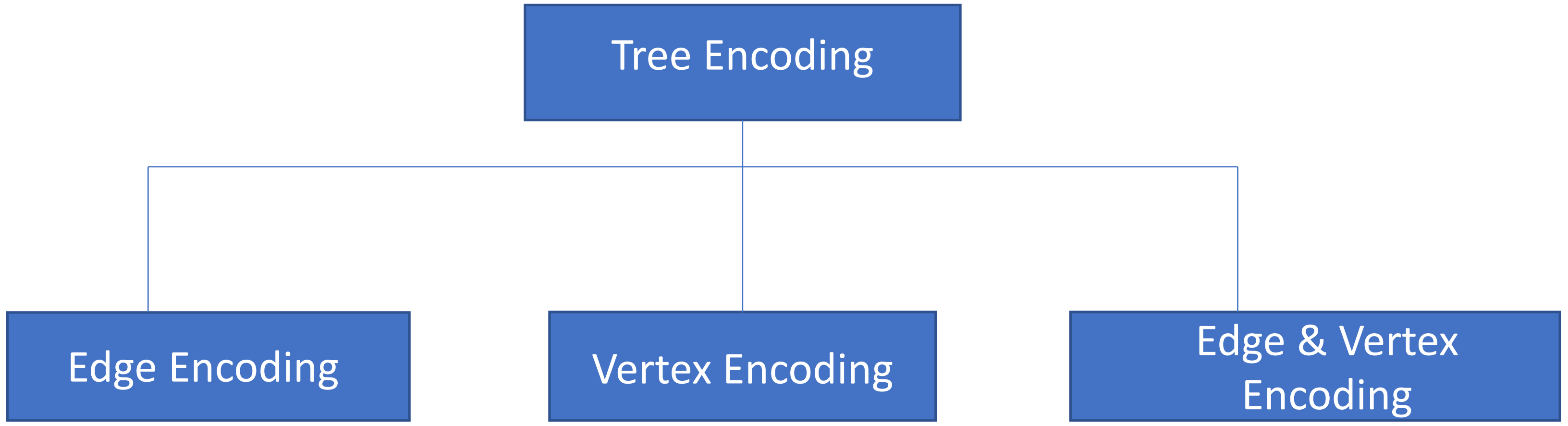
Solution is encoded with random numbers from (0, 1) - 8 city TSP

[0.45 0.68 0.91 0.11 0.62 0.34 0.74 0.89] can be represented as 3-5-8-1-4-2-6-7

(machine scheduling- resource allocation- vehicle routing-quadratic assignment problem)

Genetic Representation-Examples

Tree Representation

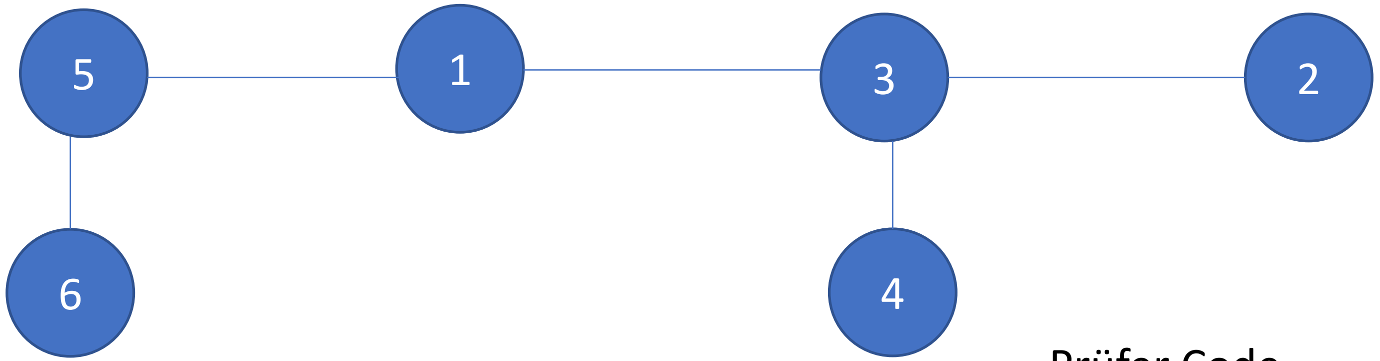


For GA's, one method of encoding input is through Tree Representation i.e. a chromosome may be represented through a tree.

Genetic Representation-Examples

Vertex Encoding

Prufer's Code: Represents a tree in a particular way.



Prüfer Code

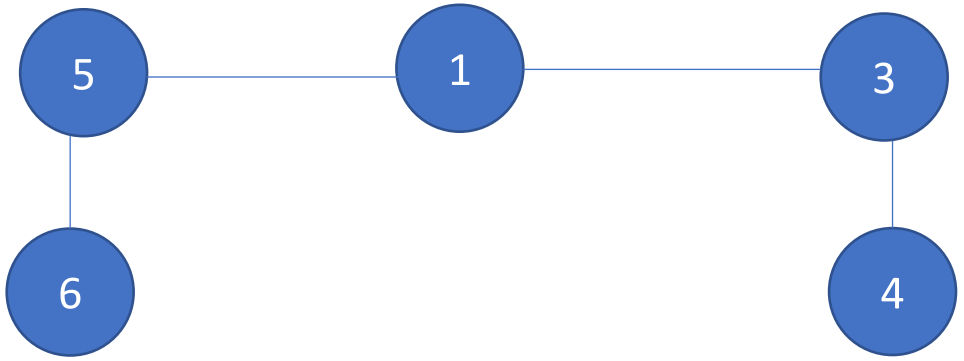
Generate Prüfer Code for the above tree.

The length of Prüfer's Code is given by **n-2,** where n is number of nodes/vertices in the tree.

Step 1: Look for the leaf nodes with **the smallest label**. So, its **2** and edge incident from **node 2 is to node 3**. So **remove node 2** from the tree and **add 3** at the first position in the Prüfer code.

3			

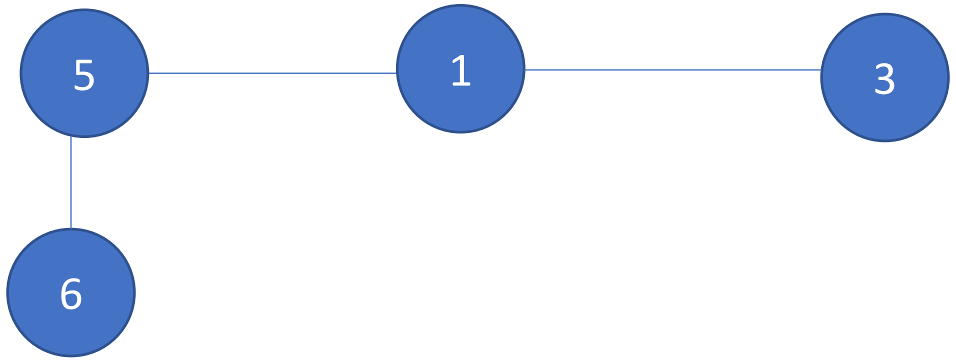
Genetic Representation-Examples



Prüfer Code

3	3		

Step 2: Now look for the leaf nodes with the **smallest label** again and **its node 4** and **edge incident from node 4 is to node 3**. So **remove 4** from the tree and add 3 at the second position in the Prüfer code.



Continuing like this till just one edge is left and we will have Prüfer's Code of length 4.

So,

3	3	1	5

is the final Prüfer's Code corresponding to given tree.

Representing Hypotheses

- Hypothesis in GAs are often represented by bit strings, which are easily manipulated by crossover and mutation.

- Consider sets of if-then rules that can easily be represented by bit strings, by choosing an encoding of rules that allocate specific substrings for each rule precondition and postcondition.

Representing Hypotheses- Example

- Attribute: **Outlook=Sunny, Rainy, Overcast**

- Use a bit string of length 3, where each position corresponds to one of the values. Placing a 1 in some position indicates that the attribute is allowed to take on the corresponding value.

- So 001 represents **Outlook = Overcast** and 011 represents **Outlook = Overcast or Rainy**

Id code	outlook	temp	humidity	windy	play
a	Sunny	high	high	strong	no
b	s	h	h	weak	n
c	overcast	h	h	s	y
d	Rainy	mild	h	s	y
e	r	cool	normal	s	y

More Representations

- Conjunctions of constraints can be represented by concatenation.

  Eg. **011 10** represents **Outlook = Rainy or Overcast and Wind = Strong**

- Postconditions can be represented in the same way **111 10 10** represents **If Wind = Strong then PlayTennis = Yes**. Notice that **111** represents the "don't care" condition on Outlook

- Sets of rules can be represented by concatenating single rules, but may not be fixed length!

- Some GAs represent hypothesis as symbolic descriptions rather than bit strings.

References

[1] https://www.solver.com/products-overview
[2] Holland JH. Genetic algorithms. Scientific american. 1992 Jul 1;267(1):66-73.
[3] https://www.tutorialspoint.com/genetic_algorithms/
[4] Hristakeva M, Shrestha D. Solving the 0-1 knapsack problem with genetic algorithms. In Midwest instruction and computing symposium 2004 Apr 16.

THANK YOU

Dr. Arti Arya
Department of Computer Science
artiarya@pes.edu
+91 9972032451 Extn 029