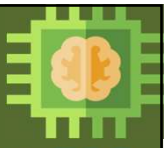
Elective Course
Course Code: CS4103
Autumn 2025-26



Lecture #20

Artificial Intelligence for Data Science

Week-6: Introduction to Genetic Algorithm (GA) [Part-I]

Course Instructor:

Dr. Monidipa Das
Assistant Professor
Department of Computational and Data Sciences
Indian Institute of Science Education and Research Kolkata, India 741246

Genetic Algorithms (GAs)

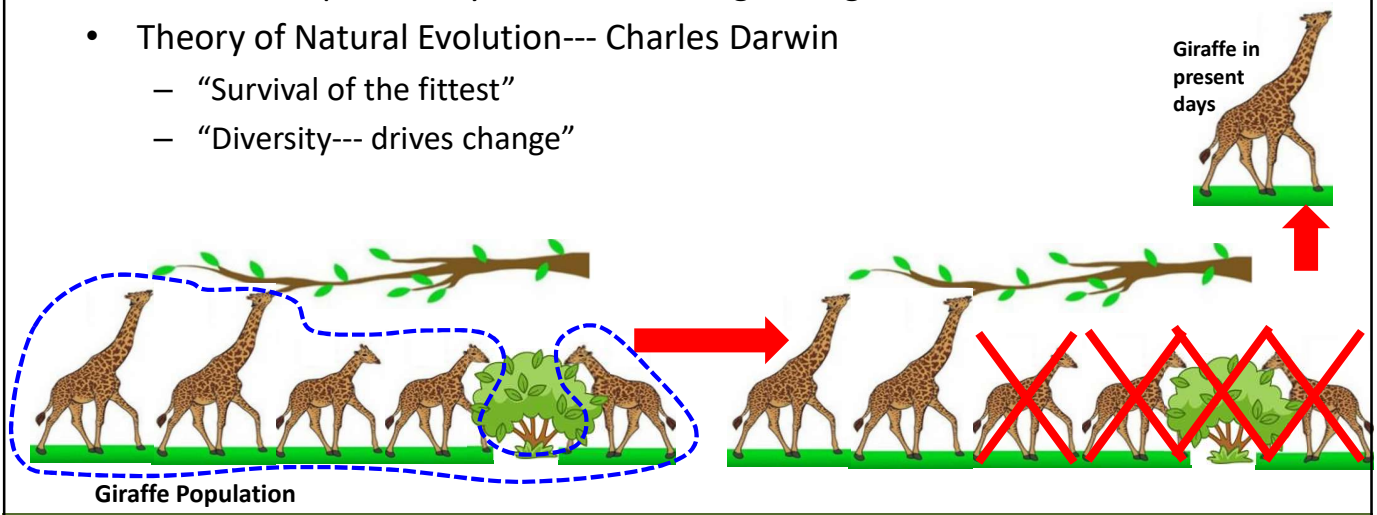
- Genetic Algorithms are the ***heuristic search and optimization techniques*** that mimic the process of ***natural evolution***.
- Introduced and developed by **John Holland** (1975)
- Belong to the family of **Evolutionary Algorithms** (EAs)

Topics of Discussion

- What is Natural Evolution?
- Where is the similarity between natural evolution and optimization problems?
- General structure of Evolutionary Algorithm
- Overview of Genetic Algorithm

Natural Evolution

- The natural process by which the biological organisms evolve
- Theory of Natural Evolution--- Charles Darwin
  - "Survival of the fittest"
  - "Diversity--- drives change"



Optimization Problem

- Problem of finding the best solution from a set of all feasible solutions
- General form

$$\min/\max \; f(\boldsymbol{x}) \qquad \boldsymbol{x} = (x_1, x_2, \ldots x_n) \in \mathbb{R}^n$$

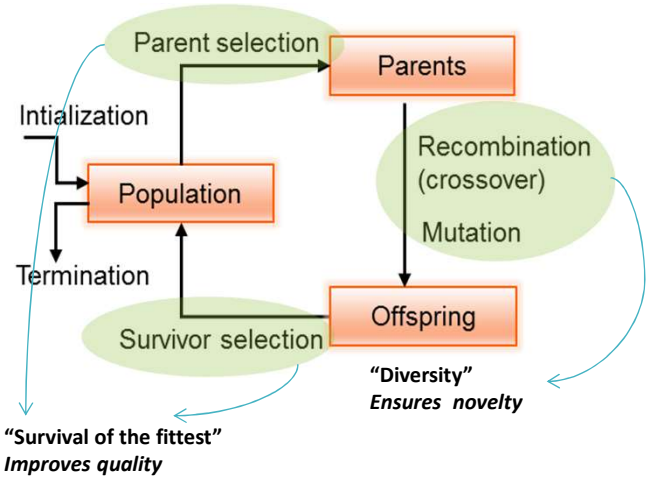
$$\text{subject to } g_j(\boldsymbol{x}) \geq 0, \quad j = 1, 2, \cdots, J$$

$$h_k(\boldsymbol{x}) = 0, \quad k = 1, 2, \cdots, K$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \cdots, n$$

Evolutionary Algorithm (EA)

- EAs fall into the category of "generate and test" algorithms
- These are stochastic, population-based algorithms
- Variation operators (recombination and mutation) create necessary diversity and facilitate novelty
- Selection reduces diversity and acts as a force pushing quality



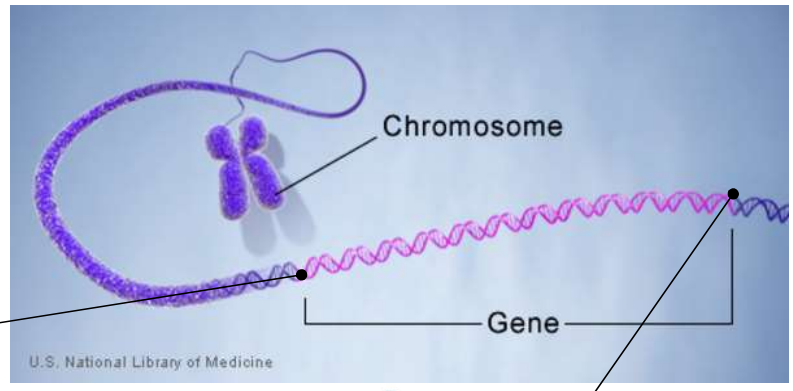
Genetic Algorithm: An Overview

- Randomized heuristic search algorithms
- Not random....Historical information directs the search
- Based on evolutionary ideas
  - **Survival of the Fittest**
  - **Natural Genetics**

Natural Genetics

- **Heredity**
  - Heredity is the transfer of characteristics (or traits) from parent to offspring through genes
  - Information needs to be passed on from one generation to the next

Gene, Chromosome



- Gene:
  - Unit of heredity
  - Key factor behind the natural evolution
- Chromosome:
  - Coiled genes

Genetic Variation

- There has to be differences in the characteristics of individuals in order for change to occur
- The primary mechanisms of achieving genetic variation are:
  - Mutation
  - Crossover (reproduction)

MUTATION

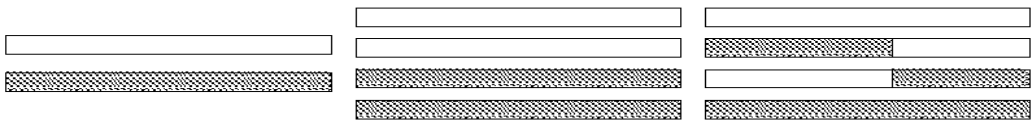
11

- It is a random change within gene
- It can be beneficial, neutral or harmful to the organism
- Not all mutations matter to evolution

Crossover

12

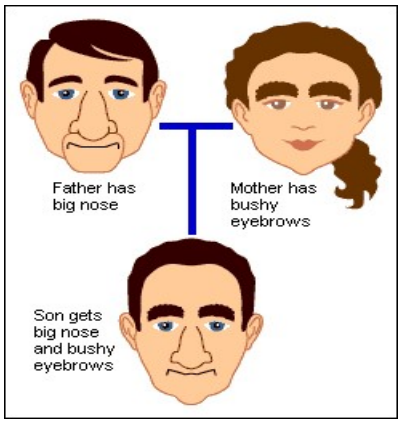
- First, the crossover operator randomly chooses a crossover point where two parent chromosomes “break”,
- Then exchanges the chromosome parts after that point. As a result, two new offspring are created.



Reproduction

13

- This type of producing young can introduce new gene combinations through genetic shuffling



Genetic Algorithm: An Overview

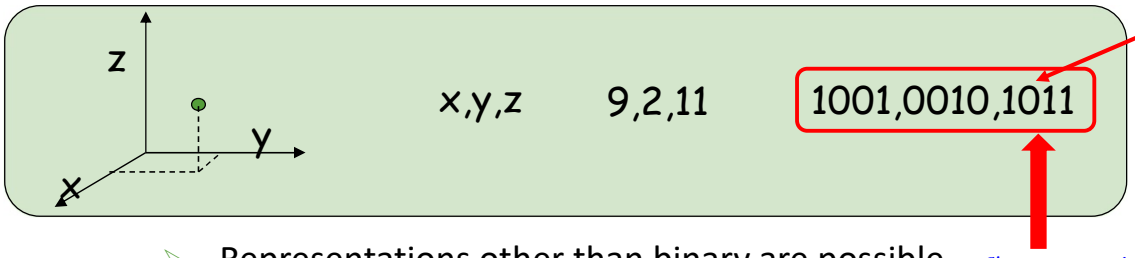
14

- **Step-1:** Start with a randomly generated population of **individuals**.
  The population size is n.
- **Step-2:** Calculate the **fitness** of each individual in the population.
- **Step-3:** Repeat the following steps until n children or offspring are created.
  - **Select** pair of parents from the population
  - With probability P_c perform **crossover** on these selected parents to generate two offspring
  - **Mutate** offspring with probability P_m
- **Step-4:** Replace the old population with this new population
- **Step-5:** Go to step 2 until the termination condition is reached.

Basic Concepts

- **Individual**
  - **Candidate solution**– Chromosome
  - The typical candidate representation is a binary string. This string can be thought of as the genetic code of a candidate – thus the term "genetic algorithm"!

Candidate representation example



- Representations other than binary are possible
- but they make crossover and mutation harder.

E.g.: $\max f(x) = 15x - x^2$

$1 \le x \le 15$

Integer	Binary code	Integer	Binary code	Integer	Binary code
1	0 0 0 1	6	0 1 1 0	11	1 0 1 1
2	0 0 1 0	7	0 1 1 1	12	1 1 0 0
3	0 0 1 1	8	1 0 0 0	13	1 1 0 1
4	0 1 0 0	9	1 0 0 1	14	1 1 1 0
5	0 1 0 1	10	1 0 1 0	15	1 1 1 1

Candidate representation example

- Let's say we want to represent bikes as the individuals
  - Make (Bridgestone, Cannondale, Nishiki, or Gary Fisher)
  - Tire type (knobby, treads)
  - Handlebar type (straight, curved)
  - Water bottle holder (*Yes, No*)
- We can encode this as a binary string, where each bit represents whether a value is accepted.

Make				Tires		Handlebars		Water bottle	
B	C	N	G	K	T	S	C	Y	N

*Example from "Genetic Algorithm", Richard Frankel, Stanford University

Candidate representation example

- Any bike that is made by Bridgestone or Cannondale, has treaded tires, and has straight handlebars can be represented as 1100011011:

Make				Tires		Handlebars		Water bottle	
1	1	0	0	0	1	1	0	1	1
B	C	N	G	K	T	S	C	Y	N

Genetic Algorithm: An Overview

- **Step-1:** Start with a randomly generated population of **individuals**.
  The population size is n.
- **Step-2:** Calculate the **fitness** of each individual in the population.
- **Step-3:** Repeat the following steps until n children or offspring are created.
  - **Select** pair of parents from the population
  - With probability P_c perform **crossover** on these selected parents to generate two offspring
  - **Mutate** offspring with probability P_m
- **Step-4:** Replace the old population with this new population
- **Step-5:** Go to step 2 until the termination condition is reached.

Fitness Function

- The fitness function is analogous to a heuristic that estimates how close a candidate is to being a solution.

E.g.: $f(x) = 15x - x^2$

Chromosome/Individual	Decimal value (x)	Fitness (f(x))
1100	12	36
0101	5	50

Genetic Algorithm: An Overview

- **Step-1:** Start with a randomly generated population of **individuals**.
  The population size is n.
- **Step-2:** Calculate the **fitness** of each individual in the population.
- **Step-3:** Repeat the following steps until n children or offspring are created.
  - **Select** pair of parents from the population
  - With probability P_c perform **crossover** on these selected parents to generate two offspring
  - **Mutate** offspring with probability P_m
- **Step-4:** Replace the old population with this new population
- **Step-5:** Go to step 2 until the termination condition is reached.

To be discussed in the next lecture

Questions?