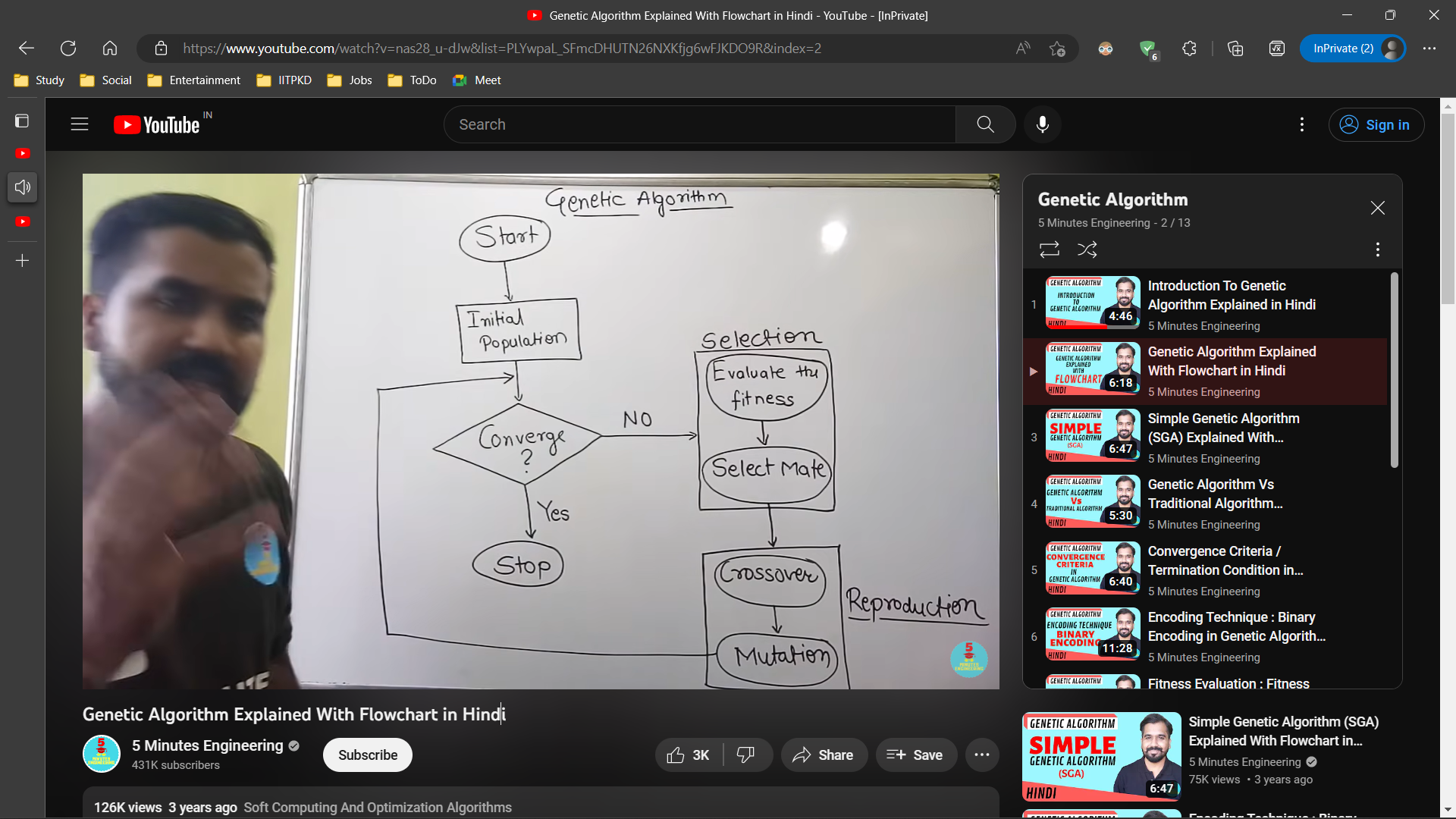
Genetic Algorithm

[Mayank Singla](mailto:111901030@smail.iitpkd.ac.in)

* An evolutionary algorithm
* Adaptive Heuristic Search Algorithm
* **Based on Genetics and Natural Selection**
* To generate a high-quality solution for the optimization problem
* Population (group of individuals maintained within a search space) and Individual (possible solution for a given problem)
* Operators of GA
  + Selection (Select the fittest for the mating process)
  + Crossover (Reproduction of the fittest to generate a new offspring)
  + Mutation (Ultra sum of the gene)
  + Encoding (On initial population)

**Flowchart:**

****

Initial Population - To generate a new generation population / Kind of Random Solution

Converge - Termination Condition (optimal solution)

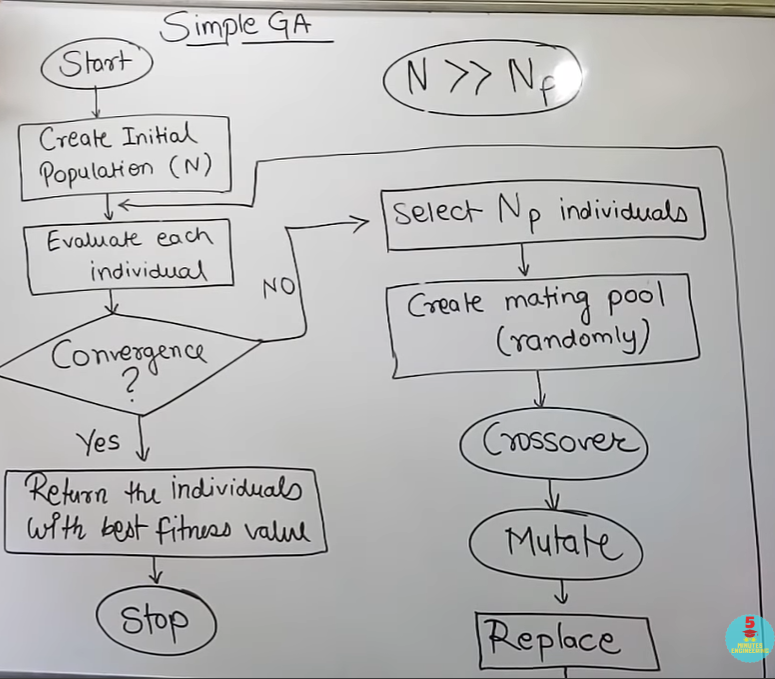
Fitness - Using the fitness score

Crossover - Consider k points

- Generating the offspring using selected mates

Mutation - Alter the genes of the offspring reproduced to get new offspring

**Simple GA:**

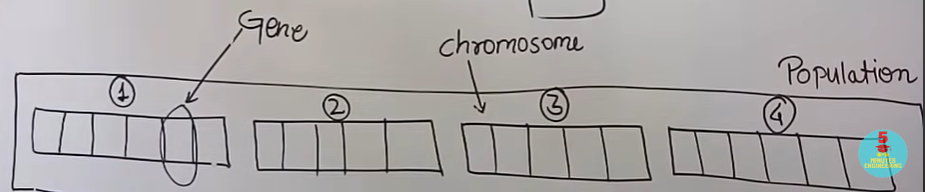
****

* Evaluate fitness score
* Select Np individuals -> With Replacement Policy i.e. we can select one individual multiple times
* Create a mating pool by selecting pair of parents and sending them for reproduction

**Convergence Test / Termination Condition:**

* Manual Checking (erroneous)
* Solution found that satisfies objective criteria
* Fixed number of generations (no. of iterations)
* Budget limit reached (time to run)
* Not getting better results / same results / convergence

**Encoding / Spawning:**

****

* Can be used to generate the initial population
* Each individual can be treated as a chromosome and chromosomes have many genes
* **Binary Encoding:** Individuals/Chromosome as a binary string and genes as bits

**Genotype vs Phenotype**

* Genotype → Encoding Data
* Phenotype → Data Expression

**Fitness Evaluation / Fitness Function / Objective Function:**

* Some criteria need to be defined
* Ex. distance in TSP, profit in 0-1 knapsack

**Crossover:**

Take two parents(chromosomes) and generate new offspring using their genes(bits), only consider unique offspring

* **Single Point Crossover:**
  + Have a single point in both the binary strings, and break them into two parts
  + Swap their parts to generate new offspring
* **Two-Point Crossover:**
  + Swap only the middle parts
* **Multi-Point Crossover:**
  + Swap alternate parts
* **Uniform Crossover:**
  + Deciding for each bit of parent if we want to swap those bits or not by tossing a coin for each bit probabilistically
* **Uniform Crossover with Crossover Mask:**
  + Crossover Mask is a randomly generated binary string of the same length
  + To produce offspring, create some rules for each offspring and produce them

**Mutation:**

* Alter genes of offspring more to have more genetic variation b/w offspring and parent
* Ex. Flipping with mutation probability for each bit (again a binary string and decide for each bit)
* Ex. Interchanging any two randomly picked bits

**Selection:** Selection ɑ Fitness

* **Tournament Selection:**
  + We need to pick Np parents
  + Each time we will randomly pick x individuals and conduct a tournament between them to find the winner to get one parent
  + This way we will have Np tournaments to get Np parents
  + If the winner is based on fitness score, the max scorer will be the winner always
* **Roulette Wheel Selection:**
  + Selecting by giving weights to the probabilities of each individual which is directly proportional to their fitness score (score / sum of score)
* **Rank-Based Selection:**
  + Give rank 1 to the least fittest and rank N to the fittest
  + Find out the percentage value for each individual (rank / sum or ranks)
  + Select by taking these as weights for probabilities
  + The roulette wheel is more biased as compared to this, so this is better

**Hard Constraints:**

* ✅All lectures of a course must be scheduled in distinct periods
* ✅Faculty conflict - Two classes of the same faculty in the same period
* ✅No student group should have one class at a time
* ✅Room conflict - Two lectures in the same room in the same period
* ✅Room Capacity and Features - Enough capacity and normal/lab rooms

**Soft Constraints:**

* ✅Time to travel between the campuses
* ✅Room Stability - All lectures of a course in the same room
* Lectures evenly distributed
* Faculty Slot Preferences

START

Generate the initial population

Compute fitness

REPEAT

Selection

Crossover

Mutation

Compute fitness

UNTIL population has converged

STOP

**JS:**

[**p5.js Web Editor | NOC 9.4 Genetic Algorithm - Shakespeare (p5js.org)**](https://editor.p5js.org/codingtrain/sketches/PqRSmKLQU)

[**p5.js Web Editor | NOC 9.5 Genetic Algorithms - Smart Rockets (p5js.org)**](https://editor.p5js.org/codingtrain/sketches/BOTCxBDbO)