Genetic Algorithms

Introduction

In computer science and operations research, a genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.[1] John Holland introduced genetic algorithms in 1960 based on the concept of Darwin’s theory of evolution

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. This is in line with the Darwinian Theory of “Survival of the Fittest”.

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.

Advantages of GAs :

- Does not require any derivative information (which may not be available for many real-world problems).
- Is faster and more efficient as compared to the traditional methods.
- Has very good parallel capabilities.
- Optimizes both continuous and discrete functions and also multi-objective problems.
- Provides a list of “good” solutions and not just a single solution.
- Always gets an answer to the problem, which gets better over the time.

- Useful when the search space is very large and there are a large number of parameters involved.

Limitations of GAs:

- GAs are not suited for all problems, especially problems which are simple and for which derivative information is available.
- Fitness value is calculated repeatedly which might be computationally expensive for some problems.
- Being stochastic, there are no guarantees on the optimality or the quality of the solution.
- If not implemented properly, the GA may not converge to the optimal solution.

Key Terms :

Genetic Algorithm

A genetic algorithm (GA) characterizes potential problem hypotheses using a binary string representation, and iterates a search space of potential hypotheses in an attempt to identify the "best hypothesis," which is that which optimizes a predefined numerical measure, or fitness. GAs are, collectively, a subset of evolutionary algorithms.

Evolutionary Algorithm

An evolutionary algorithm (EA) is any type of learning method motivated by their obvious and intentional parallels to biological evolution, including, but not limited to, genetic algorithms, evolutionary strategies, and genetic programming.

Genetic Programming

Genetic programming is a specific type of EA which leverages evolutionary learning strategies to optimize the crafting of computer code, resulting in programs which perform optimally in a predefined task or set of tasks.

Population

In a GA, each iteration, or generation, results in a series of possible hypotheses for best approximating a function, and the population refers to the complete set or pool of these generated hypotheses after a given iteration.

Chromosome

In an obvious nod to biology, a chromosome is a single hypothesis of which many make up a population.

Gene

In a GA, potential hypotheses are made up of chromosomes, which are, in turn, made up of genes. Practically, in a GA, chromosomes are generally represented as binary strings, a series of 1s and 0s, which denote inclusion or exclusion of particular items represented by position in the string. A gene is a single bit within such a chromosome.

For example, the Hello World of genetic algorithms is often considered to be the knapsack problem. In this problem, there would be a set of N items which may or may not be included in a thief's knapsack, and these N items would be represented as a binary string (the chromosome) N characters long, with each position in the string representing a particular item and the positional bit (1 or 0; the gene) denoting whether the item is included in the particular hypothesis or not.

Population → all of the proposed solutions to the knapsack problem of the current generation (iteration of the algorithm)

Chromosome → a particular proposed solution to the knapsack problem

Gene → positional representation of a particular item (and its inclusion or exclusion) in the knapsack of a particular solution to the knapsack problem

Generation

In GAs, new sets of hypotheses are formed from previous sets of hypotheses, either by selecting some full chromosome (generally of high fitness) to move forward to a new generation unscathed (selection), by flipping a bit of an existing full chromosome and moving it forward to a new generation (mutation), or, most commonly, by breeding child chromosomes for the new generation by using an existing set's genes as parents.
A generation, then, is simply the full set of the results of a GA iteration.

Breeding

Breeding refers to what is generally the most common method of creating new chromosomes from an existing generation's set of hypotheses, which is using a pair of said chromosomes as parents and creating from them new child chromosomes, using the crossover method.

Selection

In a nod to natural selection, the concept of selection is ensuring that the best performing (highest fitness) chromosomes are ensured a higher probability of being used for breeding the next generation. Often highest performing chromosomes may be selected and pushed forward into the new generation without being used for breeding, ensuring that subsequent generations of hypotheses will minimally perform at the same level of the current generation.

Crossover

How are selected chromosomes used to breed subsequent generations? The crossover method, shown below, is the general choice. A pair of selected strings of N bits in length would be chosen, and a random integer c generated as point of crossover of some size (say, $0 < c < N$). The 2 strings are then independently split at this crossover point c and reassembled using the head of one string and the tail of the other, forming a pair of new chromosomes. The fitness of these new hypotheses would then be assessed in the following generation.

Mutation

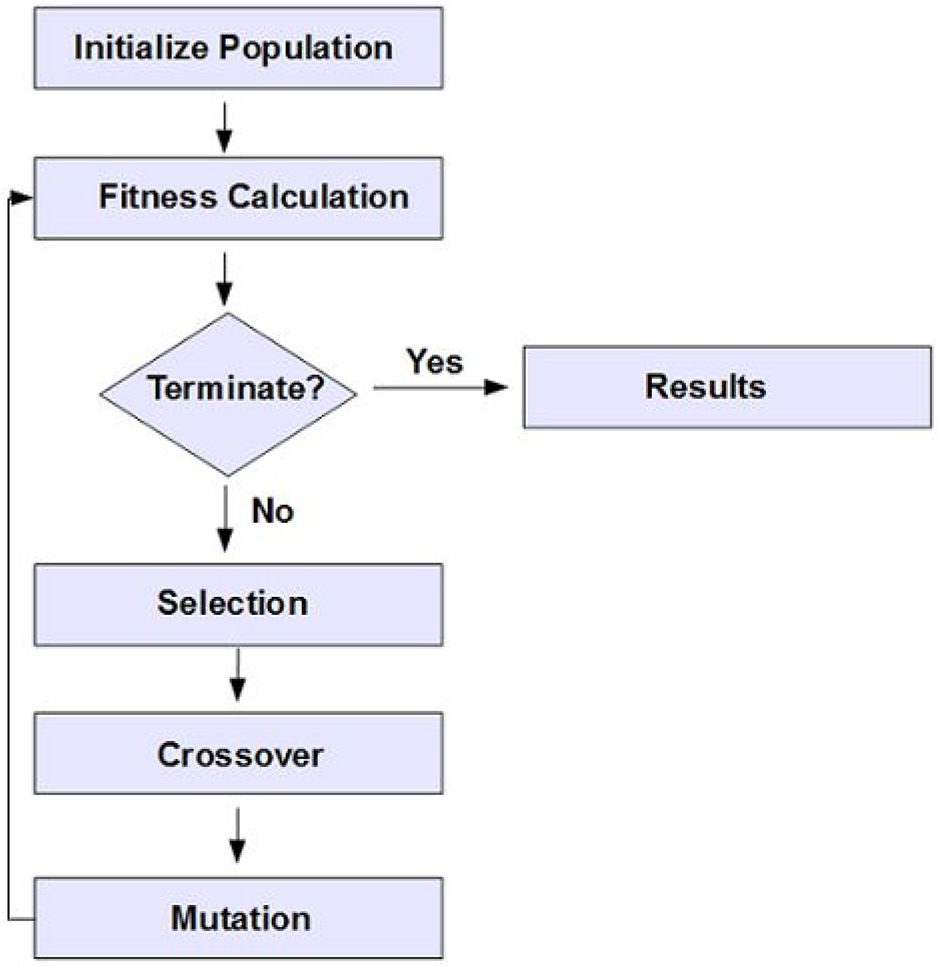
Just like in biological terms, mutation is used in GAs in order to push hypotheses toward optimal. Generally used sparingly, mutation would simply flip the bit of a random gene and push the entire chromosome forward to the subsequent generation, a strategy for escaping potential local minima.

Fitness

We need some metric to measure the best fit of a hypothesis. Some fitness function is used to evaluate each chromosome, and best fits can be identified and more heavily relied upon in order to create new generational chromosomes. The fitness function is heavily task-dependent.

Psuedo-code

```
START
Generate the initial population
Compute fitness
REPEAT
    Selection
    Crossover
    Mutation
    Compute fitness
UNTIL population has converged
STOP
```



Parent Selection Techniques :

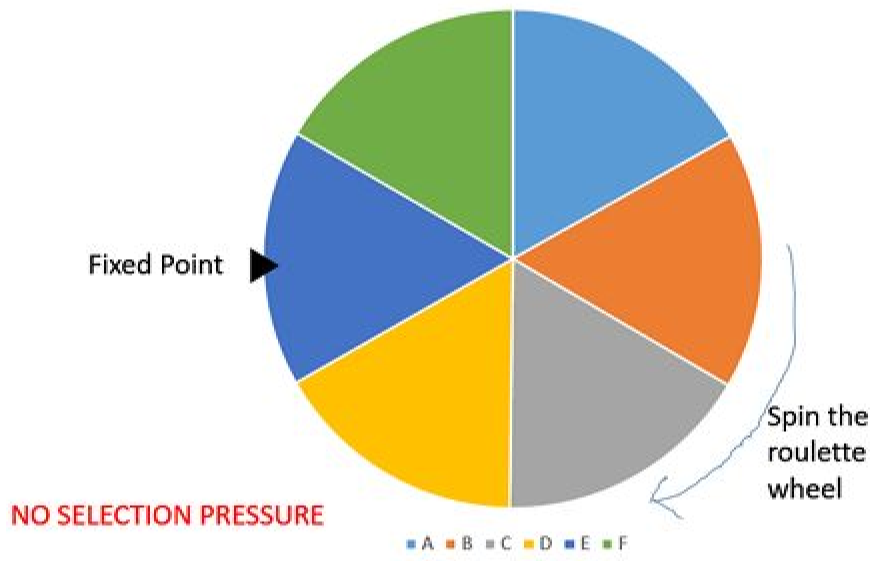
1] Roulette Wheel Selection

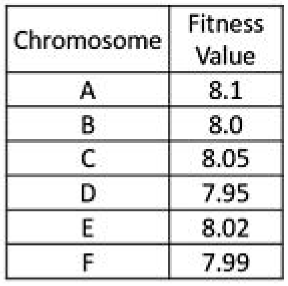
In a roulette wheel selection, the circular wheel is divided as described before. A fixed point is chosen on the wheel circumference as shown and the wheel is rotated. The region of the wheel which comes in front of the fixed point is chosen as the parent. For the second parent, the same process is repeated.

It is clear that a fitter individual has a greater pie on the wheel and therefore a greater chance of landing in front of the fixed point when the wheel is rotated. Therefore, the probability of choosing an individual depends directly on its fitness.

Implementation wise, we use the following steps −

a] Calculate S = the sum of a finesses.
b] Generate a random number between 0 and S.
c] Starting from the top of the population, keep adding the finesses to the partial sum P, till P<S.
d] The individual for which P exceeds S is the chosen individual.

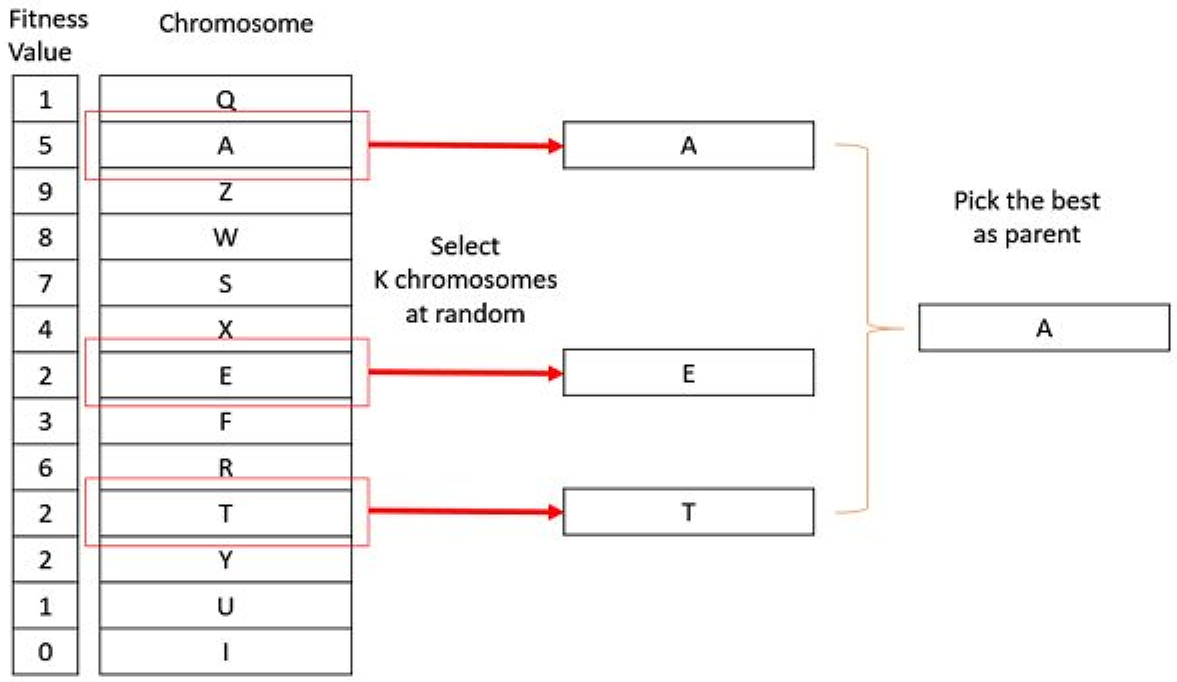




Chromosome	Fitness Value
A	8.1
B	8.0
C	8.05
D	7.95
E	8.02
F	7.99

2] Tournament Selection

In K-Way tournament selection, we select K individuals from the population at random and select the best out of these to become a parent. The same process is repeated for selecting the next parent. Tournament Selection is also extremely popular in literature as it can even work with negative fitness values.



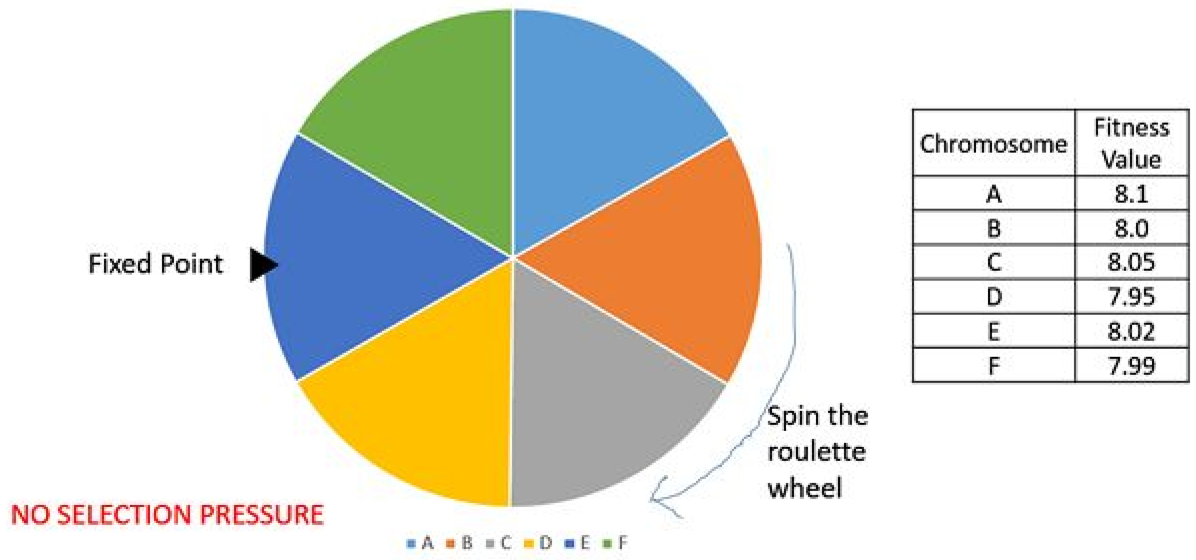
3] Rank Selection

Rank Selection also works with negative fitness values and is mostly used when the individuals in the population have very close fitness values (this happens usually at the end of the run). This leads to each individual having an almost equal share of the pie (like in case of fitness proportionate selection) as shown in the following image and hence each individual no matter how fit relative to each other has an approximately same probability of getting selected as a parent. This in turn leads to a loss in the selection pressure towards fitter individuals, making the GA to make poor parent selections in such situations.

In this, we remove the concept of a fitness value while selecting a parent. However, every individual in the population is ranked according to their fitness. The selection of the parents depends on the rank of each individual and not the fitness. The higher ranked individuals are preferred more than the lower ranked ones.

Chromosome	Fitness	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

Values normalized using ranks instead of fitness.
Used when most fitness values are close to each other,this would lead to random selection
By using ranks as weights we ensure fitter individuals are given more preference



Chromosome	Fitness Value
A	8.1
B	8.0
C	8.05
D	7.95
E	8.02
F	7.99

4] Random Selection

In this strategy we randomly select parents from the existing population. There is no selection pressure towards fitter individuals and therefore this strategy is usually avoided.