Soft Computing - Module 5 - Genetic Algorithms

Theoretical Background

Genetic Algorithms(GAs) are **adaptive heuristic search algorithms** that belong to the larger part of evolutionary algorithms.

Genetic algorithms are based on A genetic algorithm is an adaptive heuristic search algorithm inspired by "**Darwin's theory of evolution in Nature.**" the ideas of natural selection and genetics.

They are commonly used to generate high-quality solutions for optimization problems and search problems.

It is one of the important algorithms as i**t helps solve complex problems** that would take a long time to solve.

Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to the next generation.

Genetic Algorithm Concepts:

Abstraction of real biological evolution:

Genetic algorithms are an abstraction of real biological evolution that is used in computer science and engineering to solve optimization problems.

The genetic algorithm simulates the process of natural selection, where advantageous traits are selected and passed on to the next generation, leading to the gradual optimization of the population. This abstraction of biological evolution is particularly useful for solving complex optimization problems that would be difficult or impossible to solve with traditional methods.

Focus on optimization

Genetic algorithms are a powerful optimization technique that are widely used in computer science, engineering, and other fields to solve complex problems. Unlike traditional optimization methods, which rely on analytical or mathematical techniques to find an optimal solution, genetic algorithms are based on the principles of natural selection and evolution.

The key focus of genetic algorithms is optimization, which involves finding the best possible solution to a problem within a given set of constraints. This could involve finding the optimal configuration of parameters for a machine learning model, the best route for a delivery truck to take, or the most efficient design for a building.

Solve complex problem

Genetic algorithms are a powerful optimization technique that are used to solve complex problems across a wide range of fields. They are particularly effective when the problem at hand is too complex to solve using traditional mathematical or analytical methods.

The main advantage of genetic algorithms is their ability to handle large, high-dimensional search spaces that would be difficult or impossible to explore exhaustively using traditional optimization methods. This makes them well-suited for solving complex problems in fields such as engineering, computer science, finance, and biology.

Survival of the fittest

In genetic algorithms, the principle of "survival of the fittest" is an essential concept that underpins the process of evolution and optimization. This principle refers to the idea that individuals with the best fitness or performance in each environment are more likely to survive and reproduce, passing on their traits to the next generation.

Terminologies:

Population: Population is the subset of all possible or probable solutions, which can solve the given problem.

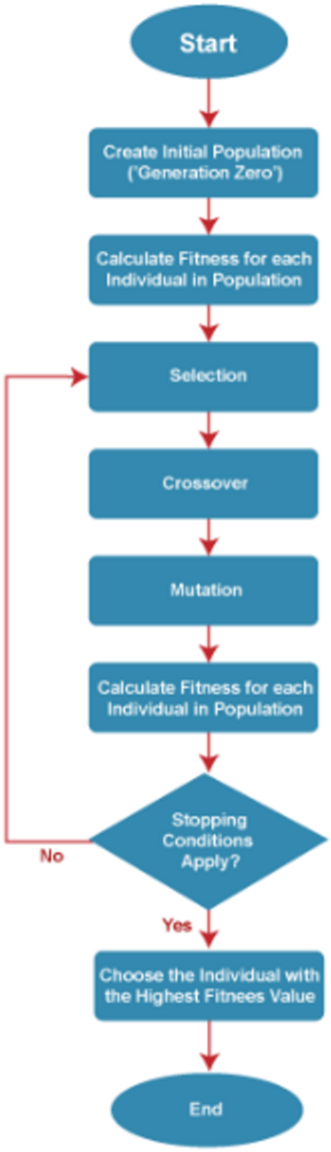
Chromosomes: A chromosome is one of the solutions in the population for the given problem, and the collection of gene generate a chromosome.

Gene: A chromosome is divided into a different gene, or it is an element of the chromosome.

Allele: Allele is the value provided to the gene within a particular chromosome.

Fitness Function: The fitness function is used to determine the individual's fitness level in the population. It means the ability of an individual to compete with other individuals. In every iteration, individuals are evaluated based on their fitness function.

Simple Genetic Algorithm:


Start
Create Initial Population ('Generation Zero')
Calculate Fitness for each Individual in Population
Selection
Crossover
Mutation
Calculate Fitness for each Individual in Population
Stopping Conditions Apply?
No
Yes
Choose the Individual with the Highest Fitnees Value
End

1. Select Np individuals from the previous population.
2. Create the mating pool randomly.
3. Perform Crossover.
4. Perform Mutation in offspring solutions.
5. Perform inversion in offspring solutions.
6. Replace the old solutions of the last generation with the newly created solutions and go to step 2

Search Space:

The population of individuals are maintained within search space.

Everyone represents a solution in search space for given problem.

Everyone is coded as a finite length vector (analogous to chromosome) of components.

These variable components are analogous to Genes.

Thus, a chromosome (individual) is composed of several genes (variable components).

Advantages of Genetic Algorithm

- The parallel capabilities of genetic algorithms are best.
- It helps in optimizing various problems such as discrete functions, multi-objective problems, and continuous functions.
- It provides a solution for a problem that improves over time.

Limitations of Genetic Algorithms

- Genetic algorithms are not efficient algorithms for solving simple problems.
- It does not guarantee the quality of the final solution to a problem.
- Repetitive calculation of fitness values may generate some computational challenges.

Difference between Genetic Algorithms and Traditional Algorithms

- A search space is the set of all possible solutions to the problem. In the traditional algorithm, only **one set of solutions is maintained**, whereas, in a genetic algorithm, s**everal sets of solutions in search space can be used**.
- Traditional algorithms **need more information** in order to perform a search, whereas genetic algorithms **need only one objective function to calculate the fitness of an individual.**

Traditional Algorithms **cannot work parallelly**, whereas **genetic Algorithms can work parallelly** (calculating the fitness of the individualities are independent).

- One big difference in genetic Algorithms is that rather of operating directly on seeker results, **inheritable algorithms operate on their representations** (or rendering), frequently called as chromosomes.
- One of the big differences between traditional algorithm and genetic algorithm is that **it does not directly operate on candidate solutions**.

Traditional Algorithms **can only generate one result in the end**, whereas Genetic Algorithms can generate **multiple optimal results from different generations**.

- The traditional algorithm is not more likely **to generate optimal results**, whereas Genetic algorithms do not guarantee to generate optimal global results, but also there is a great possibility of getting the optimal result for a problem as it uses genetic operators such as Crossover and Mutation.

Traditional algorithms are **deterministic in nature**, whereas Genetic algorithms are **probabilistic in nature**

Operators in Genetic Algorithm:

Selection:

individual genomes are chosen from a population for later breeding

- Roulette wheel selection
- Rank-grounded selection
- Tournament Selection

Crossover

The genes of parents are exchanged among themselves until the crossover point is met. These newly generated offspring are added to the population. This process is also called crossover

Types of crossover styles available:

- One point crossover
- Multi-point crossover
- Uniform crossover

Mutation

The mutation operator inserts random genes in the offspring (new child) to maintain the diversity in the population. It can be done by flipping some bits in the chromosomes.

Mutation helps in solving the issue of premature convergence and enhances diversification. The below image shows the mutation process:

Types of mutation styles available,

- Flip bit mutation/Binary Mutation
- Swap Mutation
- Scramble Mutation
- Inversion Mutation

Encoding:

Encoding of chromosomes is t**he first step in solving the problem** and it depends entirely on the problem heavily.

The process of representing the solution in the form of a string of bits that conveys the necessary information.

Just as in a chromosome, each gene controls particular characteristics of the individual, similarly, each bit in the string represents characteristics of the solution.

Binary Encoding: Most common methods of encoding. Chromosomes are strings of 1s and 0s and each position in the chromosome represents particular characteristics of the solution.

Permutation Encoding: Useful in ordering such as the Travelling Salesman Problem (TSP). In TSP, every chromosome is a string of numbers, each of which represents a city to be visited.

Value Encoding: Used in problems where complicated values, such as real numbers, are used and where binary encoding would not suffice. Good for some problems, but often necessary to develop some specific crossover and mutation techniques for these chromosomes.

Convergence/Termination Criteria

1. Manual Checking: More prone to error
2. Solution found that satisfies objective criteria: Based on Objective criteria e.g Threshold
3. Fixed Number of Generation(Iteration): To go near Optimum Solution
4. Budget Limit Reached: Fix the convergence criteria as per time and space.

Problem solving using Genetic Algorithm

Define the problem: The first step is to define the problem you want to solve and determine the parameters that will be used to evaluate potential solutions. For example, if you want to optimize a function, you need to define the range of values that the function can take and the criteria that will be used to evaluate how well a solution performs.

Define the population: The next step is to create a population of potential solutions. This is typically done by randomly generating a set of initial solutions. Each solution should be represented as a set of parameters.

Define the fitness function: The fitness function is used to evaluate how well a solution performs. It takes a solution as input and returns a value that represents its fitness. The fitness function should be designed so that higher values indicate better solutions.

Select parents: The next step is to select the parents that will be used to create the next generation. This is typically done by using a selection method such as tournament selection or roulette wheel selection. The better a solution performs, the more likely it is to be selected as a parent.

Crossover: The crossover operator is used to combine the genes of two parents to create a new solution. This is typically done by selecting a random crossover point and swapping the genes before and after the point between the two parents.

Mutation: The mutation operator is used to introduce small random changes to a solution. This helps to prevent the algorithm from getting stuck in local optima. Mutation is typically done by randomly selecting a gene and changing its value.

Evaluate fitness: Once the new generation has been created, the fitness function is used to evaluate how well each solution performs.

Repeat: The process of selecting parents, applying crossover and mutation, and evaluating fitness is repeated until a stopping criterion is met. This could be a maximum number of generations, a minimum fitness level, or a combination of both.

Select the best solution: Once the algorithm has completed, the best solution found is selected based on its fitness value. This solution represents the optimal solution to the problem.

- **Objective Questions:**

Which of the following is NOT a characteristic of genetic algorithms?

a) Simulating natural selection and evolution

b) Handling problems with a large number of variables

c) Efficiently solving linear problems

d) Handling problems with non-linear relationships

What is the role of the fitness function in a genetic algorithm?

a) To generate the initial population of potential solutions

b) To evaluate the quality of solutions

c) To determine the crossover and mutation rates

d) To select the best solutions for the next generation

What is the purpose of crossover in a genetic algorithm?

a) To randomly mutate some individuals in the population

b) To introduce new individuals into the population

c) To select the best individuals from the previous generation

d) To combine genetic information from two or more individuals to create new individuals

Which of the following is NOT a benefit of using genetic algorithms?

a) Handling problems with a large number of variables

b) Robustness in handling noisy or incomplete data

c) Solving problems with linear relationships

d) Flexibility in application to various fields

How does the mutation operator in a genetic algorithm help to diversify the population?

a) By randomly selecting individuals to be replaced

b) By selecting the fittest individuals for reproduction

c) By introducing small random changes in the genetic information of individuals

d) By combining the genetic information of two or more individuals to create new individuals

- **Short Answer Questions:**

What is a genetic algorithm, and how does it work?

What is a fitness function, and why is it important in genetic algorithms?

What is crossover in a genetic algorithm, and how does it work?

What is a mutation in a genetic algorithm, and how does it work?

What are some common applications of genetic algorithms?