



Feature Selection for Predictive Modelling

A Needle in a Haystack Problem

Munshi Imran Hossain

Sudipta Basu

Introduction

- Suppose someone is studying heart diseases.
- They want to find out what are the possible factors that may cause heart diseases.
- Let's try to imagine some of these factors from common sense ...

Features

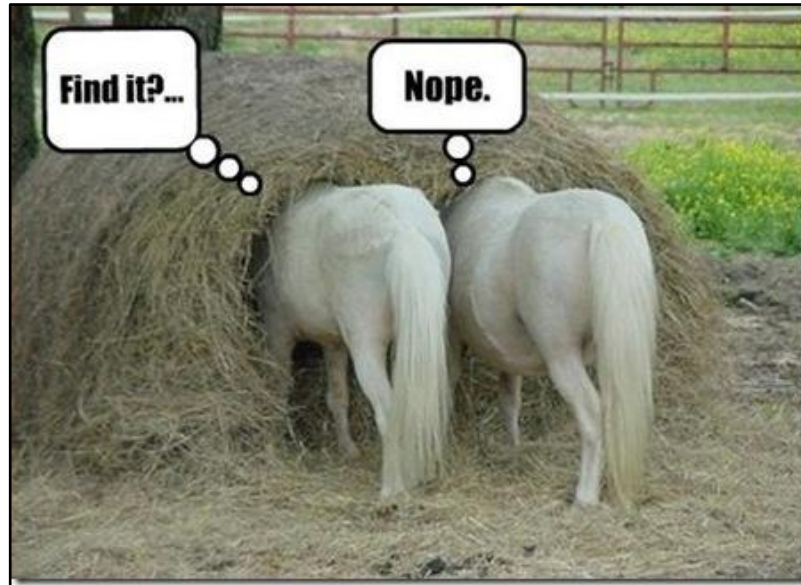
- Weight
- Smoking Habits
- Food Habits
- Drinking Habits
- Genetic Traits
- Bad Managers
- Extra Marital Affairs
- Irregular Sleep Patterns
- Stock Market Fluctuation
- Daughter's Boyfriend
- In Laws
- Unhappy Married Life
- Unsafe neighborhood
- Unemployment



An explosion of information

Needle in a Haystack

- Find relevant set of solutions.
- Solution space contains well over a trillion combinations. Finding a relevant set of solution is akin to finding a needle in a haystack!



- In an era of Big Data, this is a common problem for any field – banking, insurance, telecoms, manufacturing, healthcare, etc.

Predictive Modelling

- **Model Fitting:** A training data is used to fit a model. It is used to predict output on observations that it has not encountered.
- **Model Accuracy:** Test data is used to compute the accuracy of the model.
- **Features of the model:** Each such model has some independent variables.
- **Feature Selection:** This problem is also called the problem of Feature Selection.

Question

The question is:

Can we find a needle in a haystack in a time and cost effective manner ?

Answer

The answer is :

YES!

- There are lots of algorithms available which solve this problem
- One such group of algorithms: **Genetic Algorithms**

Genetic Algorithms

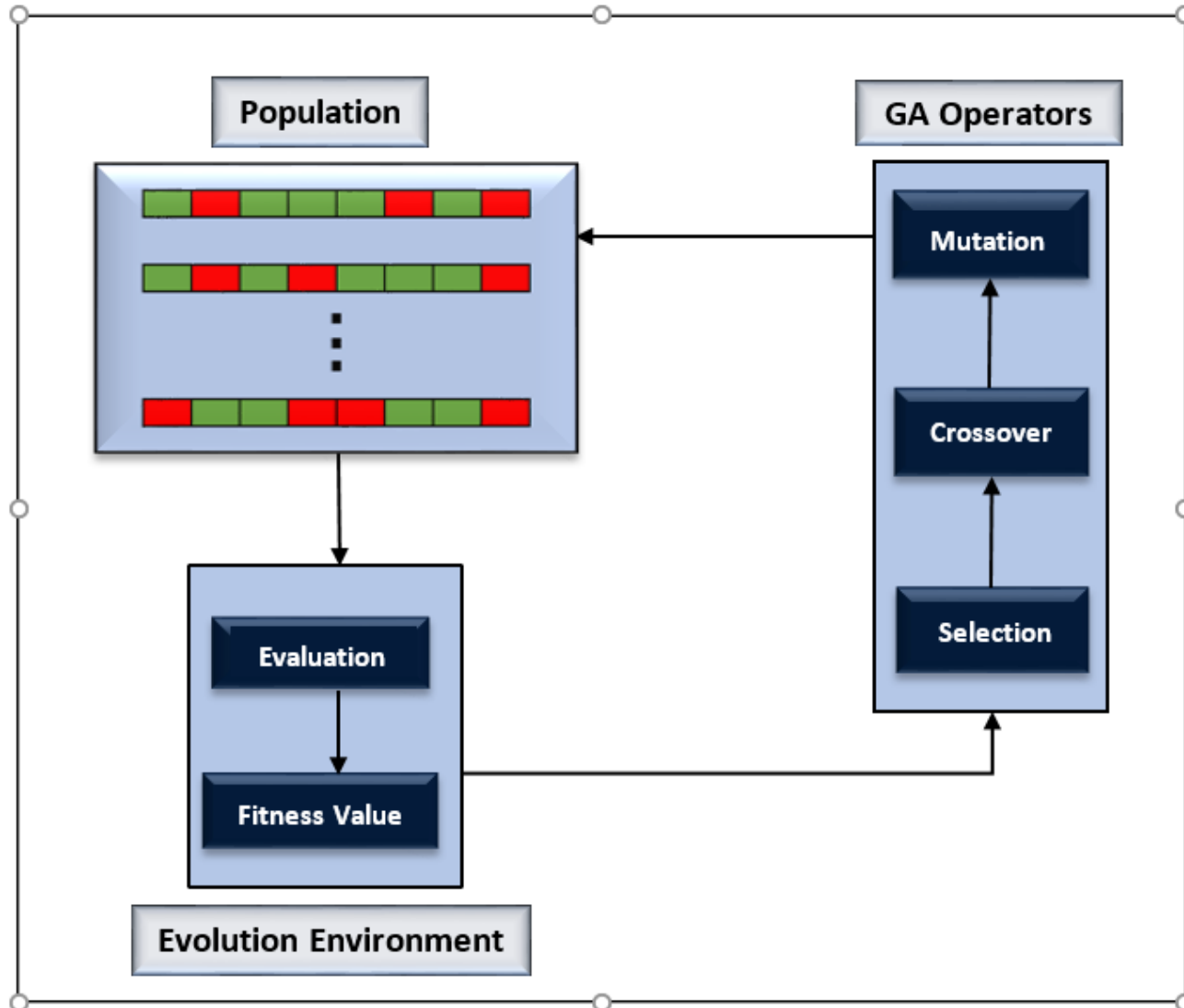
- Genetic algorithms are numerical optimization algorithms that are inspired by ideas from **Natural Selection** and **Evolutionary Biology**.
- The method is quite generic, which means that it can be used to solve optimization problems of a wide range.

Application Areas

Applicable to wide variety of problems. Some areas are –

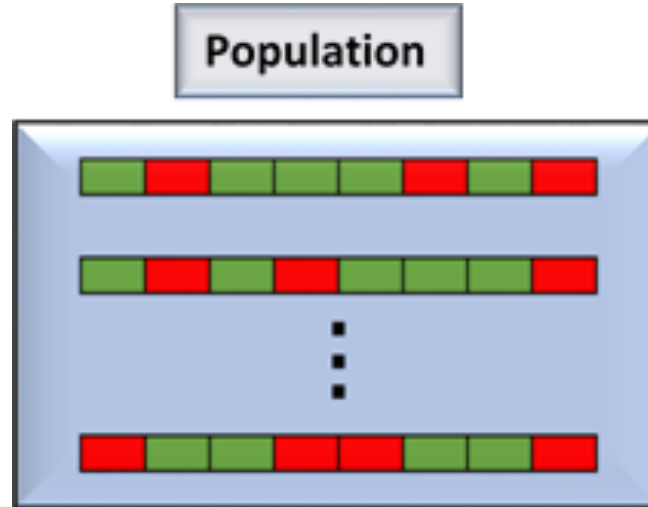
1. Prediction of three dimensional protein structure
2. Automatic evolution of computer software
3. Training and designing artificial neural networks
4. Image processing
5. Job shop scheduling

A Schematic Genetic Algorithm



Initialization

- Algorithm begins with a population of **N** solutions.



- These solutions are also called chromosomes.
- First Generation** – The first round of solutions to the problem at hand

Initialization

Evaluation

Selection

Crossover

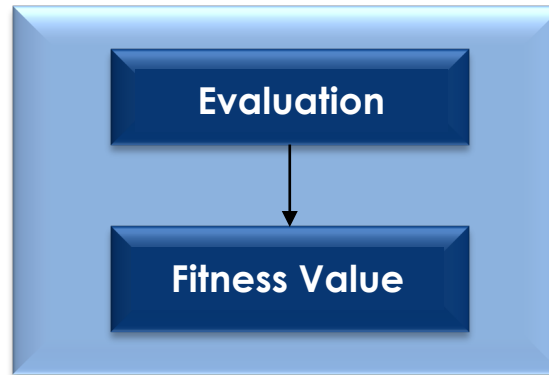
Mutation

Convergence

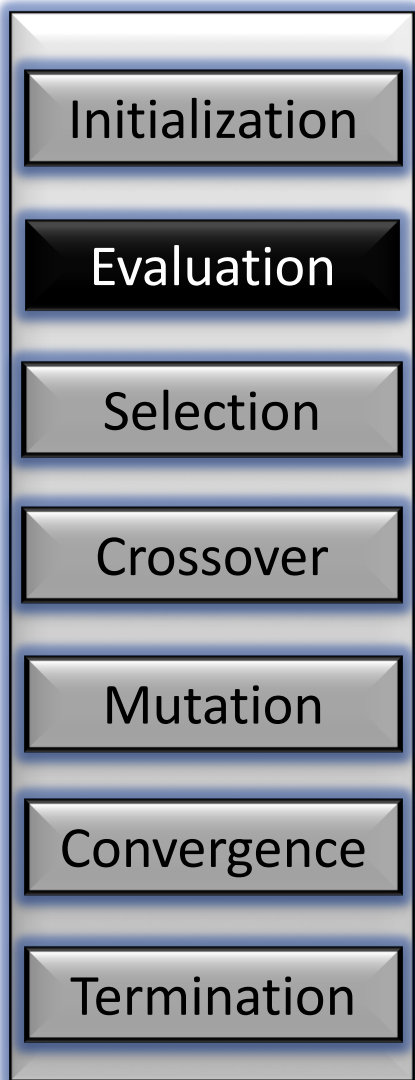
Termination

Evaluation

Evolution Environment

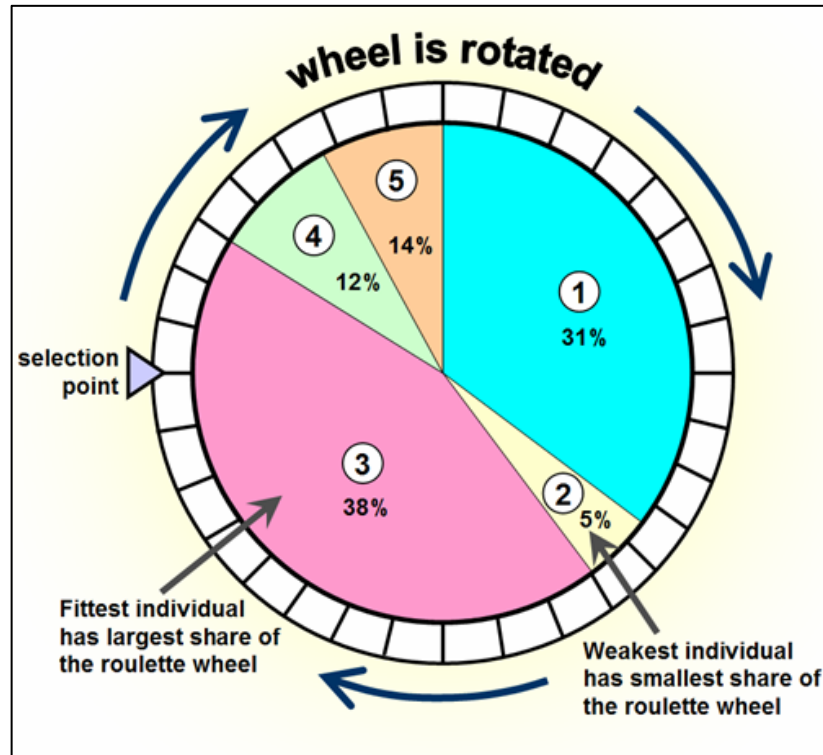


- Each solution is applied on the problem .
- A fitness value is evaluated for the solution.
- In finding root of eqn. E :
$$\text{Fitness} = |E(\text{Guess Solution}) - \text{Actual Values}|$$



Selection

- Select n best chromosomes for the next stage using a stochastic process, say “**Roulette Wheel Sampling**”.



Initialization

Evaluation

Selection

Crossover

Mutation

Convergence

Termination

Selection

Initialization

Evaluation

Selection

Crossover

Mutation

Convergence

Termination

First Generation



Selected Chromosomes

Crossover (Recombination)

Initialization

Evaluation

Selection

Crossover

Mutation

Convergence

Termination

- Selected chromosomes are used for crossover.
- It is a process in which information is exchanged between two parent chromosomes to generate new chromosomes.
- In our schematic, chromosomes are strings of binary numbers.
- Create offspring by cleaving the chromosomes at a common location.
- Continue till the new generation has **N** chromosomes.

Crossover (Recombination)



Initialization

Evaluation

Selection

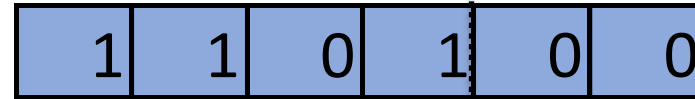
Crossover

Mutation

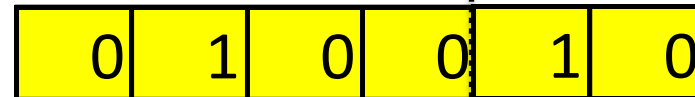
Convergence

Termination

Parent A



Parent B



Offspring 1

Offspring 2

Mutation

- New offspring chromosomes are subjected to mutation with a low probability.

- Flip *1* at a particular location in a chromosome to a *0*



- **Maintain Genetic Diversity** – Keep sufficient diversity in the population for generating new solutions in future generations.

Initialization

Evaluation

Selection

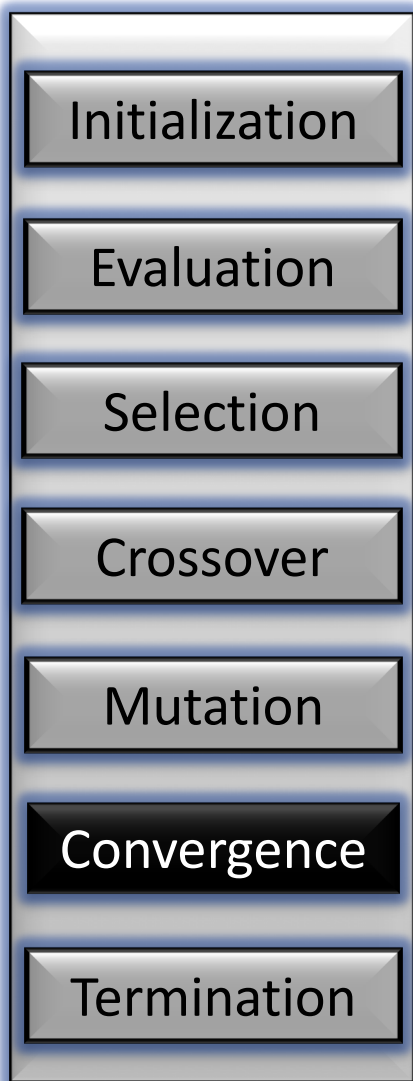
Crossover

Mutation

Convergence

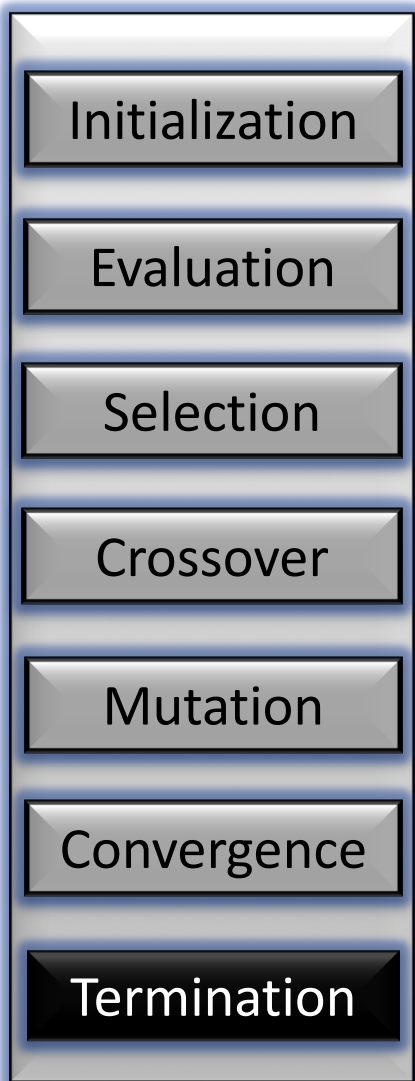
Termination

Convergence



- **Second Generation:** The processes of selection, crossover and mutation result in new chromosomes that belong to **Second Generation**.
- Compare the highest fitness value of the **Second Generation** with the **First generation** for Convergence.

Termination



- Repeat the steps from **Evaluation** to **Convergence** by creating subsequent generations until termination
- Some common termination conditions are:
 - **Convergence:** Highest fitness values of two subsequent generations remain the same (within a certain tolerance)
 - Fixed number of generations reached.
 - Allocated budget (computation time/money) reached.
 - Combinations of the above

Example: Problem Statement

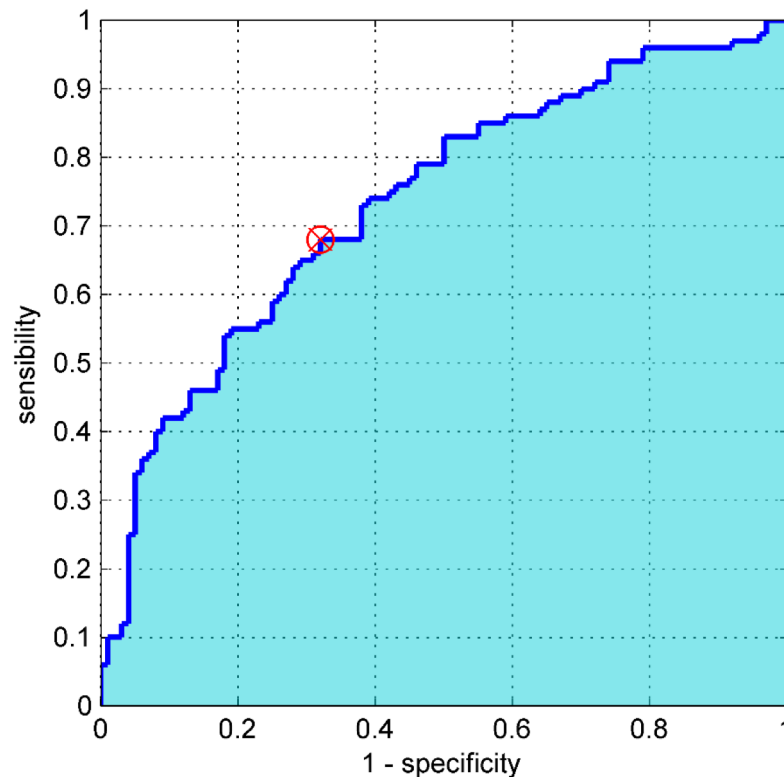
- **Data:** We generated data from a device that is used to monitor breathing.
- **Problem Statement:** A classification problem – find the set of features that will be used to build a linear discriminant analysis (LDA) model.
- **Objective:** Determine whether the breathing action of a subject is normal or labored.

Example: Simulation Parameters

- It consists of measurements on more than a **100** health parameters of subjects. Number of subjects is a little over **1500**.
- Data was split into training and test sets in a ratio of **80:20**. **10,000** such splits were made randomly.

Example: Area Under ROC Curve

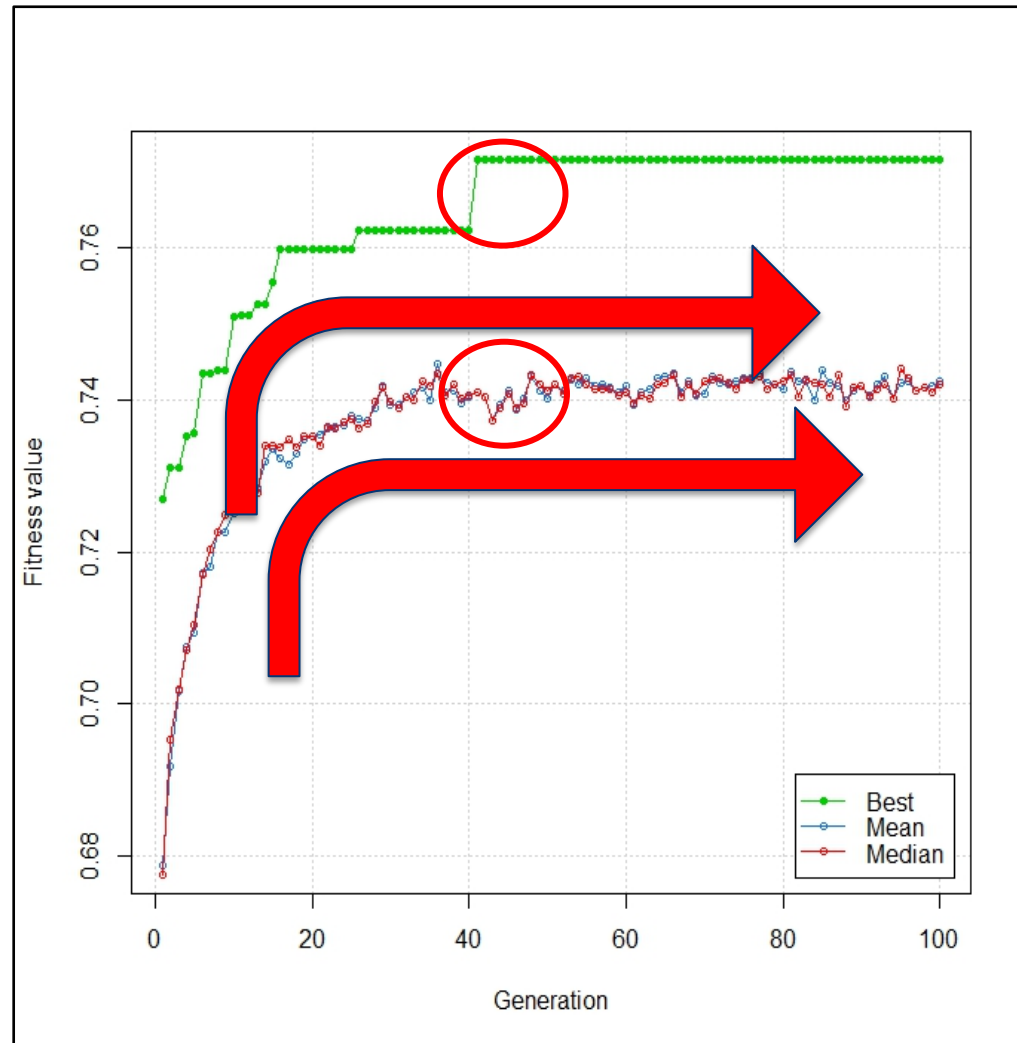
- The **LDA** model was then used to predict the outcomes on the test data and **AUC** was computed.



Example: GA Operators

GA Parameters	Values
Population size	100 chromosomes
Number of generations	100
Evaluation	Median AUC over 10,000 simulations
Selection	Roulette-Wheel Sampling
Crossover	Single Point Crossover
Mutation	1 / 27

Example: Performance



Advantages

Independent of Calculus:

- Conventional methods of optimization are based on calculus.
- They get trapped in local optima.
- They are also based on the existence of derivatives. This condition is difficult to satisfy for objective functions for many problems.
- In problems where calculus-based optimization methods are not suitable, GA can be useful for optimization.

Advantages

Flexibility:

- In addition to the main operators above, other heuristics may be employed to make the calculation faster or more robust.
- The **Speciation** heuristic penalizes crossover between candidate solutions that are too similar. This encourages population diversity and helps prevent premature convergence to a less optimal solution.

Limitations and Solutions

Repeated Fitness Function Evaluation:

- Repeated fitness function evaluation for complex problems can be prohibitive.
- In real world problems such as structural optimization problems, a single function evaluation may require **several hours to several days** of complete simulation.

Solution:

- Forgo an exact evaluation and use computationally efficient **approximated fitness**.
- Amalgamation of **approximate models** may be one of the most promising approaches to convincingly use GA to solve complex real life problems.

Limitations and Solutions

Scalability:

- GA do not scale well with complexity.
- If the number of features is large, there is exponential increase in search space size.
- Extremely difficult to use it on problems such as designing an engine, a house or plane.

Solution:

- Break the problem down into the simplest representation possible.
- Encode designs for fan blades instead of engines, building shapes instead of detailed construction plans, and airfoils instead of whole aircraft designs.

Conclusion

- In the era of big data, conventional optimization methods are sometimes not agile enough to solve the problem.
- With higher computing power from cloud computing infrastructures, genetic algorithms can be applied for solving problems in reasonable time.



Found It !

Any Questions ?



Thank You!