**GENETIC ALGORITHM**

**Abstract**

Genetic algorithm is a kind of search algorithms which is basically based on the concept of natural selection and natural genetics. It is used for solving all types of optimization problems whether it is constrained and unconstrained one so that searching process can be better. It is based on the concept of Darwin’s theory of natural evolution specified in the origin of species and ‘survival of the fittest ‘amongst the species. The meaning of the survival of the fittest in the nature is that the fittest species remain intact in the nature while the unfit species are completely removed from the nature. Based on the similar concept amongst the large number of available solutions for a particular problem, only the more fit solutions are survived while the less fit solutions are discarded.

**Introduction**

Genetic Algorithms (GAs) are search based algorithms used for solving both the constrained an unconstrained optimization problems.GA is based on the concept of natural selection process and natural genetics that belongs to larger class of evolutionary algorithm.

In genetic algorithm, there is a population of possible solutions to a given problem. This initial population of possible solution forms the current population and is parents for the next generation. At each step, the genetic algorithm randomly selects individuals from this current population and undergoes recombination and mutation and produces the new children for the next generation. This process is repeated over various generations .The population evolves towards an optimal or near optimal solution over the successive generations. Each individual of the population is assigned a fitness value. The individual which is fitter are given a higher chance to mate and generate more fitter generations of population.

It is based on the concept of Darwin’s theory of natural evolution specified in the origin of species and ‘survival of the fittest‘. So in this way ,GA keep on evolving with better individuals or solutions over the generations till the stopping criterion is achieved.

The basic structure of a GA is as follows –

The basic genetic algorithm is started with initial population of possible solutions of a problem. This initial population is generated randomly. Now parents are selected from this randomly generated initial population. These selected parents are used for mating purpose. After that genetic operators like crossover and mutation are applied over these selected parents for the generation of new off-springs. Finally these newly generated off-springs replace the existing individuals in the population. This process is repeated till an optimal or near optimal solution is obtained.

**History/Evolution of the Approach with Related Work**

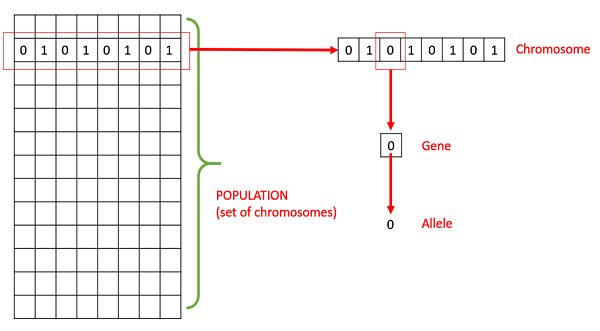
Genetic Algorithms were developed by John Holland and his students and colleagues at the University of Michigan, most notably David E. Goldberg and has since been tried on various optimization problems with a high degree of success. In 1975, Holland developed the idea of search algorithm for various optimization problems in his book “Adaptation in Natural and Artificial Systems”. He described how to apply the principles of natural evolution to optimizationproblems and built the first Genetic Algorithms. Holland’s theory has now been furtherdeveloped by leaps and bounds. Genetic Algorithms (GAs) now stand up as a powerful tool for solving search and optimization problems. Genetic algorithms are based on the basic principle of genetics and evolution. When various optimization problems were solved by using genetic algorithm then significantly good results were obtained. A heuristic is a part of an optimization algorithm. Heuristics uses the information that are currently collected by the algorithm and with the help of this gathered information heuristics decide which solution of the population should be tested next or how the next population can be produced. Genetic algorithms are guided random search and one of the most popular optimization techniques among evolutionary algorithms for multi-objective optimization problems. Genetic algorithms have been found to be capable of finding solutions for a wide variety of problems for which no acceptable algorithmic solutions exist. GA has been used for solving various NP Complete problems.GA attempts to arrive at optimal solutions through a process similar to biological evolution. To use a genetic algorithm, it is required to represent the solution of the problem as a genome (or chromosome). The genetic algorithm then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one. These operate on a population of potential solutions, applying the principle of survival of the fittest to generate improved estimations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness and breeding them together using genetic operators inspired by natural genetics. This process leads to the evolution of better populations than the previous populations. The GA consists of an iterative process that evolves a working set of individuals called a population toward an objective function, or fitness function. Genetic algorithms are typically implemented using computer simulations in which an optimization problem is specified.

**Algorithm with Mathematical Notations and Diagrams**

**Basic terminology related to Genetic Algorithm**

Before beginning a discussion on Genetic Algorithms, it is essential to be familiar with some basic terminology which will be used throughout this tutorial.

* **Population** – Population is a subset of all the possible (encoded) solutions to the given problem.
* **Chromosomes** − A chromosome is one such solution to the given problem.
* **Gene** − A gene is one element position of a chromosome.
* **Allele** − It is the value a gene takes for a particular chromosome.
* **Genotype** − Genotype is the population in the computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system.
* **Phenotype** − Phenotype is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations.
* **Decoding and Encoding** − For simple problems, the **phenotype and genotype** spaces are the same. However, in most of the cases, the phenotype and genotype spaces are different. Decoding is a process of transforming a solution from the genotype to the phenotype space, while encoding is a process of transforming from the phenotype to genotype space. Decoding should be fast as it is carried out repeatedly in a GA during the fitness value calculation.

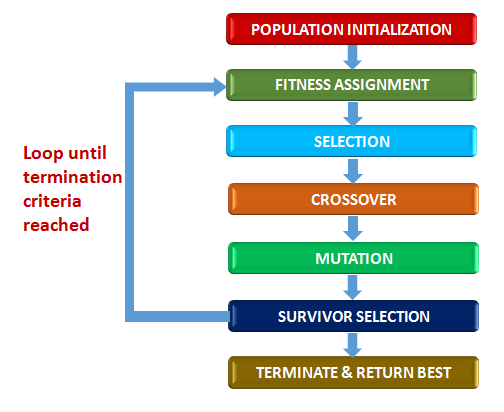


* **Fitness Function** − A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output. In some cases, the fitness function and the objective function may be the same, while in others it might be different based on the problem.
* **Genetic Operators** − these alter the genetic composition of the offspring. These include crossover, mutation, selection, etc.

## Basic Structure of Genetic Algorithm

The basic structure of a GA is as follows –

The basic genetic algorithm is started with initial population of possible solutions of a problem. This initial population is generated randomly. Now parents are selected from this randomly generated initial population. These selected parents are used for mating purpose. After that genetic operators like crossover and mutation are applied over these selected parents for the generation of new off-springs. Finally these newly generated off-springs replace the existing individuals in the population. This process is repeated till an optimal or near optimal solution is obtained. The basic structure is shown as below:



**Basic Structure of Genetic Algorithm**

**Algorithm: Basic Genetic Algorithm**

**Step1:[Start]**

Generate initial population randomly of n chromosomes/individuals (suitable and possible

solutions for the problem)

**Step2:[Fitness Evaluation]**

Evaluate the fitness f(x) of each chromosome/individual x in the population

**Step3:[Generate New population]**

Generate a new population of off-springs by repeating the following steps until the new population is complete

1. **Selection**

Select two parent chromosomes from a population according to their fitness ( the better fitness, the bigger chance to get selected).

1. **Crossover**

With a crossover probability, cross over the parents to form new offspring (children).

If no crossover was performed, offspring is the exact copy of parents.

1. **Mutation**

With a mutation probability, mutate new offspring at each locus (position in chromosome)

1. **Accepting**

Place new offspring in the new population.

**Step4:[Replace]**

Use new generated population for a further run of the algorithm.

**Step5:[Test]**

If the end condition is satisfied, stop, and return the best solution in current population.

**Step6:[Loop]**

Go to the second step for fitness evaluation.

**Example problem with the solution**

**Problem:**

There are four items and each item is associated with some weight (W) and value of each item (V).

|  |  |  |
| --- | --- | --- |
| **ITEM** | **WEIGHT** | **VALUE** |
| **A** | **5 kg** | **$12** |
| **B** | **3 kg** | **$ 5** |
| **C** | **7 kg** | **$ 10** |
| **D** | **2 kg** | **$ 7** |

There is a knapsack with limited capacity that can hold at most 12 kg.

****

Now the problem is which item should be kept in the knapsack so that it will maximize knapsack value without breaking knapsack.

**Solution:**

Given a problem the first step is to represent a solution to the problem through the process is called chromosome encoding.

**Step 1:**

**Chromosome Encoding**

The original definition of Genetic Algorithm defines that an instance of a solution to be encoded using binary strings and the binary strings are called genes. The collection of strings makes a chromosome.

A better way to represent solution is to consider each gene within this chromosome as an indication whether an item is going to be present or absent in the knapsack. Presence of item in the knapsack is represented by 1 and absence of item is represented by 0

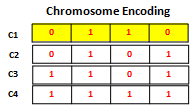
**Step 2:**

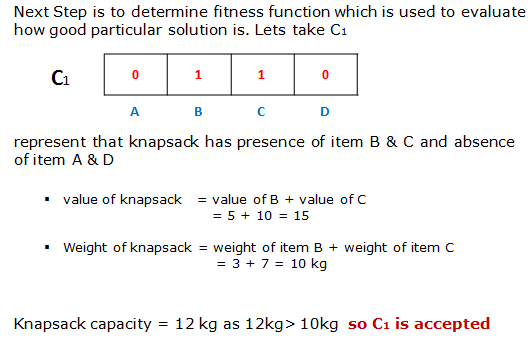
**Fitness Evaluation**

Next step is to evaluate a fitness function. For this population of chromosome from the first step is considered and evaluation is done to find out measure of goodness of a particular solution of the population .

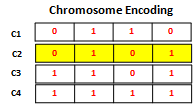
Let’s us consider first chromosome of generation1

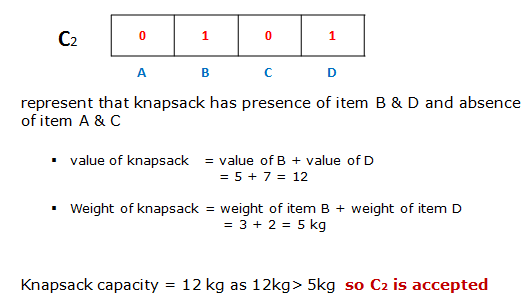
Consider C1 and itsvalue is shown below:



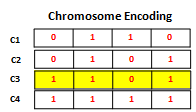
****

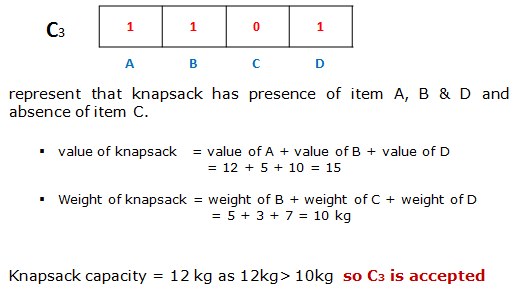
**Now Let’s take C2**

****

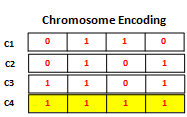
****

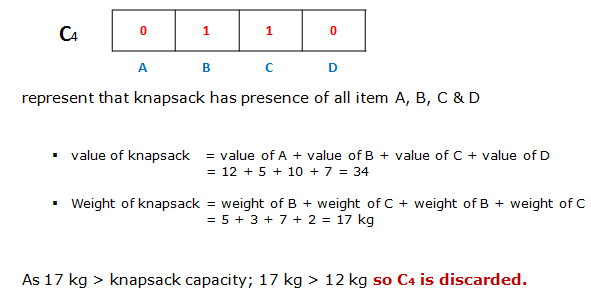
**Now take C3**

****

****

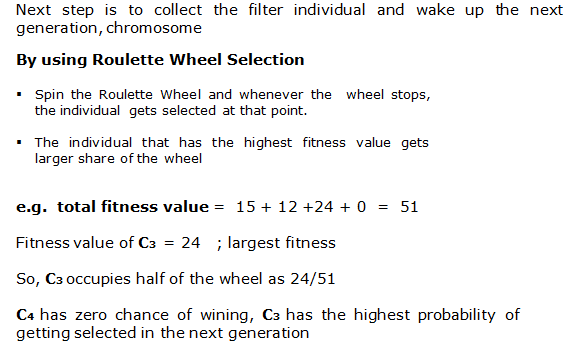
**For C4**

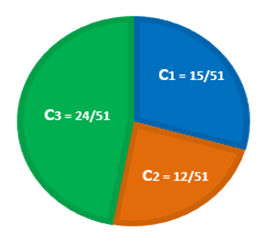
****

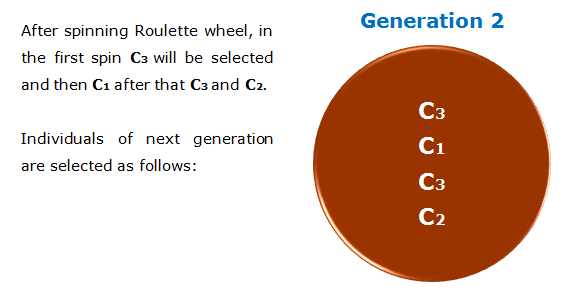
****

**Step 3:**

**Selection**

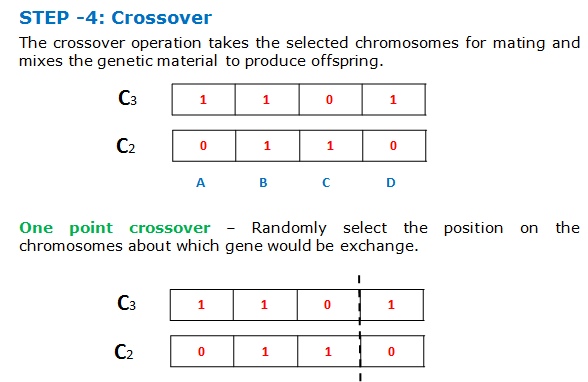
****

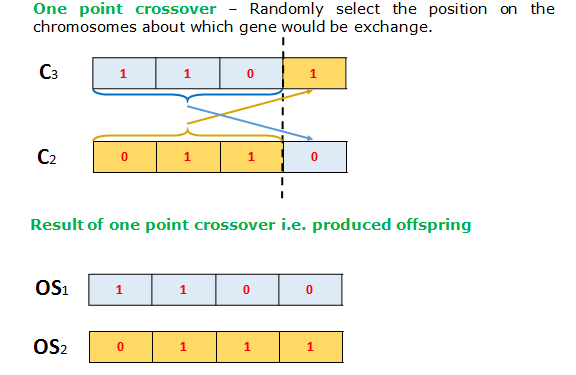
****

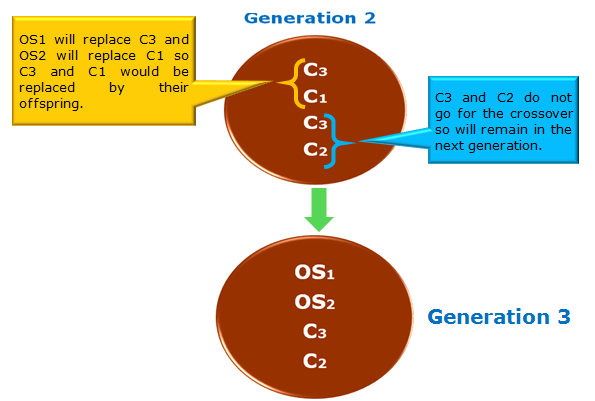
****

**Step 4:**

**Crossover**

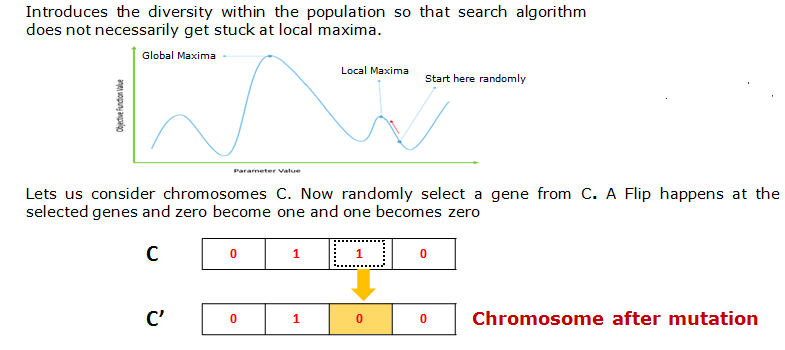
****

****

****

**Step 4:**

**Mutation:**

****

**Applications**

**Applications of Genetic Algorithm**

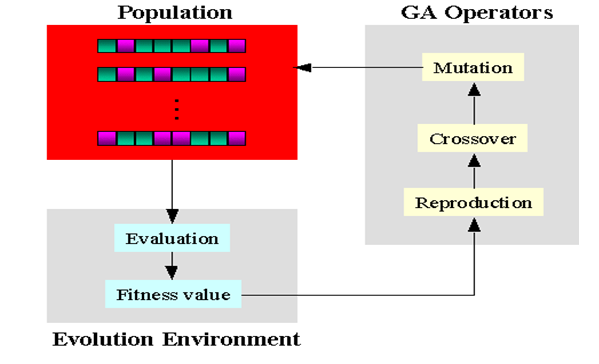
Genetic Algorithms are basically used for solving various types of constrained and unconstrained optimization problems. Genetic algorithm have been applied for solving various difficult problems like NP-hard problems, for machine learning and also for develop gradually simple problems, for evolving arts, pictures and music. Some applications of Genetic Algorithm are as follows:

* **Optimization** − Genetic Algorithms are most commonly used in wide variety of optimization problems, including numerical optimization, and combinatorial optimization problems such as traveling salesman problem (TSP), circuit design, job shop scheduling and video & sound quality optimization, telecommunication routing, state assignment problem, time tabling problem, traffic and shipment routing etc.
* **Business:** Genetic Algorithms have been used for solving various types of business problems.It has been used in various functional areas of bussiness such as finance, marketing, information systems, and production/ operations. Within these functional areas, Genetic Algorithms has performed a variety of applications such as tactical asset allocation, job scheduling, machine-part grouping, and computer network design.
* **Economics** – Genetic Algorithms have been used in optimization of economic field to characterize various economic models like the cobweb model, game theory equilibrium resolution, asset pricing, etc.
* **Neural Networks** – Genetic Algorithms are also used to train neural networks, particularly recurrent neural networks.
* **Parallelization** – Genetic Algorithms can be parallelized and also have very good parallel capabilities. This feature of genetic algorithm can be helpful in solving certain problems.
* **Image Processing** – Genetic Algorithms are used for various digital image processing (DIP) tasks as well like dense pixel matching. With medical X-rays or satellite images, there is requirement to align two images of the similar field, taken at different times. Now by comparing a random sample of points on the two images, a Genetic Algorithm can efficiently be applied to solve such problem.
* **Vehicle routing problems** –Genetic algorithms can be used to solve various vehicle routing problems.
* **Scheduling applications** − GAs are used to solve various scheduling problems as well, particularly the time tabling problem.
* **Machine Learning** –Genetic algorithms are used for machine learning applications like prediction, protein structure prediction etc. They are also used to design neural networks, to evolve rules for learning classifier systems and symbolic production systems.
* **Robot Trajectory Generation** – Genetic Algorithms have been used in the field of Robotics. In case of robot’s design, it is depended on the type of job the robot is intended to do. A range of optimal designs and components can be searched with the help of genetic algorithms to plan the path which a robot arm takes by moving from one point to another. It can also be used for each specific task and return entirely new type of robots
* **Parametric Design of Aircraft** − GAs have been used to design aircrafts by varying the parameters and evolving better solutions.
* **DNA Analysis** − GAs have been used to determine the structure of DNA using spectrometric data about the sample.
* **Multimodal Optimization** − GAs are obviously very good approaches for multimodal optimization in which we have to find multiple optimum solutions.
* **Traveling salesman problem and its applications** − GAs have been used to solve the TSP, which is a well-known combinatorial problem using novel crossover and packing strategies
* **Automatic Programming**: Genetic algorithms are used to evolve computer programs for specific tasks and to design other computational structures as in Cellular automata and sorting networks.
* **Design**: They are also used to optimize the structure and operational design of buildings, factories, machines etc. They are used to design heat exchangers, robot gripping arms, flywheels, turbines etc.
* **Evolvable Hardware**: Genetic algorithms are used develop computer models that use stochastic operators to evolve new configurations from old ones so as develop new electronic circuits that can be termed as evolvable hardware.
* **Game Playing**: Genetic algorithms are also applied in game theory and so they are widely used in developing computer games, simulated environments.
* **Encryption and code breaking**: Genetic algorithms can be used for both the creation as well as to break those codes for encryption of sensitive data.

**Discussion**

The basic principle behind GAs is that they create and maintain a population of individuals represented by chromosomes. Chromosomes are essentially a character string analogous to the chromosomes appearing in DNA. The solution of the problem is represented in the form of chromosomes by the genetic algorithm. These chromosomes are typically encoded solutions to a problem. The chromosomes then undergo a process of evolution according to rules of selection, reproduction and mutation. The fitness of the chromosomes is evaluated and the solution which is more fit is selected for the reproduction using crossover operator. The process of crossover involves two chromosomes swapping chunks of data. The diversity amongst the population is maintained by the mutation operator and it introduces slight changes into a small proportion of the population and is representative of an evolutionary. The solution (chromosome) which is more fit is selected and replaces the less fit solution (chromosome).This process continues till the optimal or near optimal solution is obtained on the basis of some pre-specified criteria.

Genetic algorithm is based on population of possible solutions or multiple points to the given problem as compared to traditional approaches which are based on single point. The main benefit of genetic algorithm is that it can be used in such type of conditions where there is failure of numerical or mathematical models .As it is an evolutionary algorithm so the view of the progress can be obtained within each iteration. GA can be used in a number of application areas such as optimization, design, robotics, image processing, machine learning, automatic programming, etc.



**Genetic Algorithm Evolution Approach**

**Advantages of Genetic Algorithm**

Genetic Algorithms have various advantages which have made them quite popular. The advantages of Genetic Algorithms are as follows:–

* The main advantage of the Genetic algorithm is its parallelism property. Most of the search techniques start from one point and continue until with a single point in each iteration until a final solution is reached. Therefore a problem of local maxima may exist in them, while the starting solution space in Genetic Algorithm is having multiple points in search space and hence the problem of local maxima generally does not exist.
* The Genetic Algorithm is much easier to implement as compared to other techniques as it requires no knowledge or gradient information about the response surface. The advantage of the Genetic Algorithm approach is the ease with which it can handle arbitrary kinds of constraints and objectives; all such things can be handled as weighted components of the fitness function, making it easy to adapt the Genetic Algorithm scheduler to the particular requirements of a very wide range of possible overall objectives.
* Genetic Algorithm can be used when no algorithms or heuristics are available for solving a problem. A Genetic Algorithm based system can be built as long as a solution representation and an evaluation scheme can be worked out. Since it only requires the description of a good solution and not how to achieve it, the need for expert access is minimized.
* Optimization problems in which the constraints and objective functions are non-linear and/or discontinuous are not amenable to solution by traditional methods such as linear programming. Genetic Algorithm can solve such problems. It does not guarantee optimal solutions, but produce near optimal solutions which are likely to be very good.
* Solution time with Genetic Algorithm is highly predictable. It is determined by the size of the population, time taken to decode and evaluate a solution and the number of generations of population.
* Genetic Algorithm use simple operations, but are able to solve problems which are found to be computationally prohibitive by traditional algorithmic and numerical techniques. One example is the TSP problem.
* It is faster and more efficient as compared to the traditional methods of optimizing a problem.
* Provides a list of “good” solutions and not just a single solution to a problem.
* It always gets an answer to the problem, which gets better over the time.
* It is useful when the search space is very large and there are a large number of parameters involved in a particular problem.

**Limitations of Genetic Algorithm**

Genetic Algorithms suffers from a few limitations. These limitations are as follows:

* One of the major problems occurs while implementing Genetic Algorithm is identification of the fitness function. As the optimal solution heavily depends on the fitness function, so it must be determined accurately. No standard techniques are available to define a fitness function. The responsibility of the user is to define fitness function.
* The diversity in the population is one of the major objectives of Genetic Algorithm and when sometimes premature convergence occur then diversity in the population is lost .
* Another problem is related with the choosing of various parameters like the size of the population, mutation rate, crossover rate, the selection method and its strength.
* The termination criteria are also not standardized. No effective single termination criteria have been identified till now.
* Genetic Algorithm only look at the fitness value of each chromosome rather than knowing what the fitness value actually means. As a result, their capability to explain why a particular solution was arrived at is practically very poor. So, Genetic Algorithms are blind to the optimization process.
* Although Genetic Algorithm are moderately scalable – an increased number of variables can be accommodated by increasing the length of the chromosome – a longer chromosome also makes finding the solution more time consuming. The longer the chromosome, the larger the population needs to be since there are more potential combinations of genes. This result in more time required for decoding and fitness evaluation.
* In general, GA does not require extensive access to data. But some applications may require access and process data from the organization’s databases to be able to evaluate the fitness of solutions. For these applications, the quality and quantity of data is important.
* Genetic Algorithms 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.

**Landmark Research papers for Genetic Algorithm**

* Garrett m. Morris,David s. Goodsell,Robert s. Halliday,Ruth huey,William e. Hart,Richard k. Belew,Arthur j. Olson1,” Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function” Journal of Computational Chemistry, Vol. 19, No. 14, 1639]1662 1998, John Wiley & Sons, Inc.
* [K. Deb](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.K.%20Deb.QT.&newsearch=true), [A. Pratap](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.A.%20Pratap.QT.&newsearch=true) , [S. Agarwal](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.S.%20Agarwal.QT.&newsearch=true) ,[T. Meyarivan](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.T.%20Meyarivan.QT.&newsearch=true),”A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, [IEEE Transactions on Evolutionary Computation](http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=4235),Volume: 6, [Issue: 2](http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=21497), Apr 2002
* [J. Horn](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.J.%20Horn.QT.&newsearch=true), [N. Nafpliotis](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.N.%20Nafpliotis.QT.&newsearch=true),[D.E. Goldberg](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.D.E.%20Goldberg.QT.&newsearch=true),”A Niched Pareto Genetic Algorithm for Multiobjective Optimization” IEEE Conference on Evolutionary Computation,27-29 June 1994
* [Gareth Jones](http://www.sciencedirect.com/science/article/pii/S0022283696908979),  [Peter Willett](http://www.sciencedirect.com/science/article/pii/S0022283696908979), [Robert C Glen](http://www.sciencedirect.com/science/article/pii/S0022283696908979),  [Andrew R Leach](http://www.sciencedirect.com/science/article/pii/S0022283696908979),  [RobinTaylor](http://www.sciencedirect.com/science/article/pii/S0022283696908979),”Development and Validation of a Genetic Algorithm for Flexible Docking”,[Journal of Molecular Biology](http://www.sciencedirect.com/science/journal/00222836),[Volume 267, Issue 3](http://www.sciencedirect.com/science/journal/00222836/267/3), 4 April 1997, Pages 727–748

**Resources**

**Books:**

The following books can be referred for getting knowledge of Genetic Algorithms:

* Genetic Algorithms in Search, Optimization and Machine Learning by **David E. Goldberg**.
* Foundations of Genetic Algorithms by **Christopher R.Stephens Marc Toussaint Darrell Whitley Peter F.Stadler (Eds)**
* Genetic Algorithms + Data Structures = Evolutionary Programs by **Zbigniew Michalewicz**.
* Practical Genetic Algorithms by **Randy L. Haupt** and **Sue Ellen Haupt**.
* Multi Objective Optimization using Evolutionary Algorithms by **Kalyanmoy Deb**.

**Tutorial Links:**

[**https://www.tutorialspoint.com/genetic\_algorithms**](https://www.tutorialspoint.com/genetic_algorithms)

[**https://www.codeproject.com/Articles/607791/ga-knapsack**](https://www.codeproject.com/Articles/607791/ga-knapsack)

**Video Lectures on YouTube**

[**https://www.youtube.com/watch?v=Z\_8MpZeMdD4&t=145s**](https://www.youtube.com/watch?v=Z_8MpZeMdD4&t=145s)

[**https://www.youtube.com/watch?v=uI5viW4r5ic&t=114s**](https://www.youtube.com/watch?v=uI5viW4r5ic&t=114s)

[**https://www.youtube.com/watch?v=zwYV11a\_\_HQ&t=70s**](https://www.youtube.com/watch?v=zwYV11a__HQ&t=70s)

[**https://www.youtube.com/watch?v=9kbzMeEBvUY**](https://www.youtube.com/watch?v=9kbzMeEBvUY)

**References**

**[1]. “**Evolutionary Approaches for Real World Applications in 21st Century” .Nikhil

Padhye Massachusetts Institute of Technology Cambridge, MA 02139, USA

[npdhye@mit.edu](mailto:npdhye@mit.edu)

[2]. “Efficient Parallel Genetic Algorithms: Theory and Practice,” Erick Cant\_u

Paz,David E. Goldberg

[3]. “A Survey of Parallel Genetic Algorithms,Erick Cantú-Paz”,Department of

Computer Science and Illinois Genetic Algorithms Laboratory,University of Illinois at

Urbana-Champaign.

[4]. “A Transaction and QoS-Aware Service Selection Approach Based on Genetic

Algorithm”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS:

SYSTEMS, ZhiJun Ding, JunJun Liu, YouQing Sun, ChangJun Jiang, and MengChu

Zhou, Fellow, IEEE

[5]. “A Genetic-Based Methodology for Evaluating Requested Outages of Power

Network Elements”, IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 26, NO.

4, NOVEMBER 2011, Julio Cesar Stacchini de Souza, Senior Member, IEEE, Milton

Brown Do Coutto Filho, Senior Member, IEEE, and Marcio Leonardo Ramos Roberto