

Genetic Algorithm And Its Application In Mechanical Engineering

¹Mohammad Zahid Rayaz Khan*, ²Dr. A K Bajpai

1. M.Tech Student, Department Of Mechanical Engineering, Madan Mohan Malaviya Engineering college, Gorakhpur, U.P, INDIA

2. Professor, Department Of Mechanical Engineering, Madan Mohan Malaviya Engineering college, Gorakhpur, U.P, INDIA

Abstract:-In this paper we present the implementation of Genetic Algorithms (GA) for various field of Mechanical engineering & other engineering discipline. Genetic algorithms (Popularly Known as GAs) have now gained immense popularity in real-world engineering search and optimization problems all over the world. Genetic algorithms are computerized search and optimization methods that work very similar to the principles of natural evolution. This paper includes application of genetic algorithm in Mechanical Engineering, advantages and limitation.

Keywords: Genetic Algorithms, Optimization etc.

Introduction

Genetic Algorithms is an optimization and search technique based on the principles of genetics and natural selection. Some fundamental idea of genetic are barrowed and used artificially to construct search algorithms that are robust and required minimum problem information.

Genetic algorithms are inspired by Darwin theory about evolution –“**Survival Of The Fittest**”. In natural competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.[1]

The concept of Genetic algorithms was first introduced by John Holland [2] of the university of Michigan over the course of the 1960s and 1970s and finally popularized by one of his students. Thereafter he and his students have contributed much to the development in the field of GAs. He was the first to try to develop a theoretical basis for GAs through his schema theorem. The work of De Jong showed the usefulness of the GA for function optimization and made the first concerted effort to find optimized GA parameters. Goldberg has probably contributed the most fuel to the GA fire with his successful applications and excellent book [2].

Basic Genetics

Every living organism consist of cells, that describe how that organisms built. Each cell have same set of Chromosomes. A Chromosomes consist of genes and DNA blocks, hereditary factors that determine particular traits of an individual are encoded into the genes and forming DNA blocks which are stung along the length of chromosomes. Each traits of an individual encoded by some combination of DNA (A,T,G,C). Complete set of genetic material is called genome particular set of genes in genome is called genotype. The physical expression of the genotype (the organism itself after birth) is called the phenotype, its physical and mental characteristics, such as eye color, intelligence etc. This is called as **reproduction**.

When two phenotype (two organism) where mate they share their genes; the resultant offspring may end up having half the genes from one organism and half from other. This is called as **crossover**.

Mutations mean change in element of DNA. This change is mainly caused by errors in copying genes from parents.

Algorithm

An algorithm is any well defined computational procedure that takes some value or set of value as an input and produce some value or set of value as an output.

The word algorithm comes from the name of the 9th century Persian Muslim Mathematician “**Abu Abdullah Muhammad ibn Musa Al-Khwarizmi**”, whose name means “*the native of Khwarezm*”, a city that was part of the Greater Iran during his era and now is in modern day Uzbekistan. He wrote a treatise in the Arabic language during the 9th century, which was translated into Latin in the 12th century under the title *Algoritmi de numero Indorum*. This title means “Algoritmi on the numbers of the Indians”, where “Algoritmi” was the translator’s Latinization of Al-Khwarizmi’s name. Al-Khwarizmi was the most widely read mathematician in Europe in the late

Middle Ages, primarily through his other book, the Algebra. In late medieval Latin, *algorismus*, the corruption of his name, simply meant the "decimal number system" that is still the meaning of modern English algorism. In 17th century French the word's form, but not its meaning, changed to *algorithmi*. English adopted the French very soon afterwards, but it wasn't until the late 19th century that "Algorithm" took on the meaning that it has in modern English. The word algorism originally referred only to the rules of performing arithmetic using Hindu-Arabic numerals but evolved via European Latin translation of Al-Khwarizmi's name into algorithm by the 18th century. The use of the word evolved to include all definite procedures for solving problems or performing tasks.[5]

An algorithm is thus a sequence of computational steps that transform the input into output. A set of rule that precisely defines a sequence of operations. Let us follow an example to help us understand the concept of algorithm in a better way. Let's say that you have a friend arriving at the railway station, and your friend needs to get from the railway station to your house. Here are different ways (algorithms) that you might give your friend for getting to your home.

- **The taxi/auto-rickshaw algorithm:** Go to the taxi/auto-rickshaw stand.

Get in a taxi/auto-rickshaw. Give the driver my address

- **The call-me algorithm:** When your train arrives, call my mobile phone.

Genetic algorithms

An algorithms is developed which is analogues to the above basic genetics that is known as genetic algorithms.

Genetic algorithms begins with set of solution called population of solution like set of chromosomes in human being genetics. Best Solution from population of solution is taken and used to form new population by using genetics operators (reproduction , crossover , mutation) by the possibility that the new population will better than the older one. Solution are selected according to their fitness to form new solution (offspring). The above priocess is repeated until some condition is satisfied .

Genetic Algorithms work with coding of the parameters set ,not the parameters themselves like the coding of hereditary factors that determine particular traits (character) of individual into DNA (A,T,G,C). GAs use probabilistic transition rule not deterministic rule .

So in GAs among the population of solution the most fittest solution is selected and go for producing new population of solution which is more fittest solution than its parent solution (older one). Fittest is survived and unfit is died out .[6]

Genetic Algorithms vaccubulary

Genetic Algorithms	Explanation
Chromosome (string , individual)	Solution (coding)
Genes	Part of solution
Locus	Position Of Genes
Alleles	Value of gene
Phenotype	Decoded solution
Genotype	Encoded solution

The Genetic Algorithms mimics the process of natural evolution by combining the survival of the fittest among solution structures with a structured, yet randomized, information exchange and creates offspring. The offspring displaces weak solutions during each generation. The idea of survival of the fittest is of great importance to genetic algorithms. There are three basic operators found in every genetic algorithm: reproduction, crossover and mutation.

- **Reproduction:** The reproduction operator allows individual strings to be copied for possible inclusion in the next generation. The chance that a string will be copied is based on the string's fitness value, calculated from a fitness function. For each generation, the reproduction operator chooses strings that are placed into a mating pool, which is used as the basis for creating the next generation.
- **Crossover:** crossover in biological terms refers to the blending of chromosomes from the parents to produce new chromosomes for the offspring. The analogy carries over to crossover in GAs. The GA selects two strings at random from the mating pool. The GA then calculates whether crossover should take place using a parameter called the crossover probability. If the GA decides not to perform crossover, the two selected strings are simply copied to the new population. If crossover does take place, then a random splicing point is chosen in a string, the two strings are spliced and the spliced regions are mixed to create two (potentially) new strings. These child strings are then placed in the new population.

As an example, say that the strings 10000 and 01110 are selected for crossover (Fig. 1) and the GA decides to mate them. The GA selects a splicing point of 3. The following then occurs: The newly created strings are 10010 and 01100. Crossover is performed until the new population is created. Then the cycle starts again with selection. This iterative process continues until any user specified criteria are met.

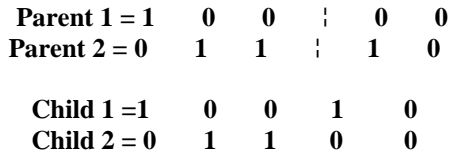


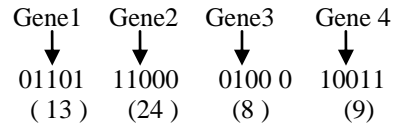
Fig 1 Crossover operation

- Mutation:** In genetics, a **mutation** is a change of the nucleotide sequence of the genome of an organism, virus, or extra chromosomal genetic element. Mutations result from unrepaired damage to DNA or to RNA genomes (typically caused by radiation or chemical mutagens), from errors in the process of replication, or from the insertion or deletion of segments of DNA by mobile genetic elements. In genetic algorithms of computing, **mutation** is a genetic operator used to maintain genetic diversity from one generation of a population of algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search. [7][9]

The purpose of mutation in GAs is preserving and introducing diversity. Mutation should allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. This reasoning also explains the fact that most GA systems avoid only taking the fittest of the population in generating the next but rather a random (or semi-random) selection with a weighting toward those that are fitter.

Examples: Maximize the function $f(x) = x^2$, $x \in (0,31)$

X will be represented as five digit using integer. So we will select randomly generated preset of solution as.[8]



We show one generational cycle done by hand .

Reproduction:

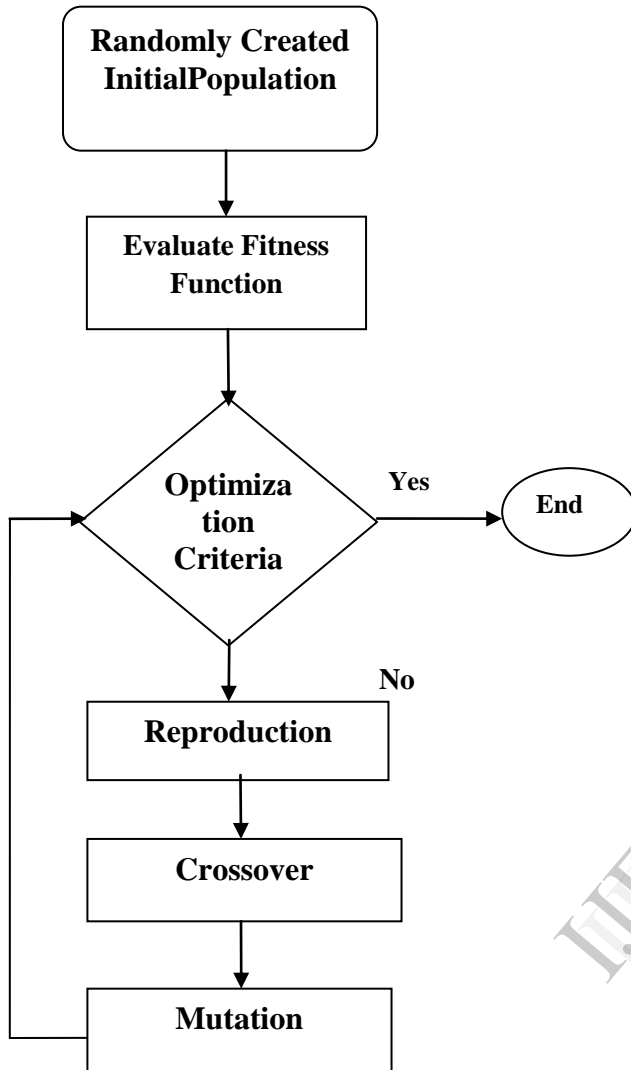
String No	Initial Population	Value of x	Fitness Function $f(x)=x^2$	Probability y	Expected count	Actual count
1.	01101	13	169	0.14	0.58	1
2.	11000	24	576	0.49	1.97	2
3.	01000	8	64	0.06	0.22	0
4.	10011	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Maximum			576	0.49	1.97	2

Crossover:

String No	Mating Pool	Cross over Point	Offspring After Crossover	Value of x	Fitness Function $f(x)=x^2$
1.	0110 1	4	01100	12	144
2.	1100 0	4	11001	25	625
3.	01 000	2	11011	27	729
4.	10 011	2	10000	16	256
Sum					1754
Average					439
Maximum					729

Mutation :

String No	Offspring After Crossover	Offspring After Mutation	Value of x	Fitness Function $f(x)=x^2$
1.	01100	<u>1</u> 1100	28	784
2.	11001	11001	25	625
3.	11011	11011	27	729
4.	10000	10 <u>1</u> 00	20	400
Sum				2538
Average				634.5
Maximum				784



Flow chart of Genetic Algorithms

Need Of Genetic Algorithms

Many optimization techniques like goal programming, multistage dynamic programming, linear programming, geometric programming, branch and bound algorithm etc are used for optimization purpose but all of them face great difficulties when the number of variables increases, because the problem becomes combinatorial explosive and hence computationally complex. Every traditionally optimization algorithm is designed to solve a specific type of problem. Different researchers used different techniques to optimize process parameters but all of those techniques have their own limitations. The solution to the optimization problems, which includes real value variables, can be obtained using numerous

methods. There is no efficient all-purpose optimization method available for nonlinear programming problems like process parameter optimization. The computational time and cost involved in the determination of optimal parameters commonly depends on the complexity or simplicity of the model. Some models can produce accurate solutions by making rigorous computation, which is not economic in terms of computation time and cost. Sometimes the solution from these models may not be optimal. Some other models may develop solutions far from the optimum in a fast manner. Therefore a compromise between the high accuracy of a rigorous solution and low accuracy of an oversimplified solution should be made. Genetic Algorithms (GAs) are robust search algorithms that are based on the mechanics of natural selection and natural genetics. They combine the idea of "survival of the fittest" with some of the mechanics of genetics to form a highly effective search algorithm. GAs are promising methods for solving difficult technological problems, and for machine learning. More generally, GAs are part of a new movement in computer science that is exploring biologically inspired approaches to computation. Genetic Algorithms overcame the limitations of other optimization techniques

- Genetic Algorithms is better than other optimization technique ; it is most robust.
- Unlike other optimization technique the GAs do not break easily even if the input changed slightly, or in the present of reasonable noise.
- While performing search in large state – space, or multi model state-space, or n-dimension space a genetic algorithms often significant benefits over many other logical search optimization technique .
- Genetic algorithms work with a coding of solution set, not the solutions themselves.
- Genetic algorithms search from a population of solutions, not a single solution.
- Genetic algorithms use probabilistic transition rules, not deterministic rules.
- GA works with a population of solutions instead of just a single solution.

Application of Genetic Algorithm in Mechanical Engineering

Genetic algorithm is an optimization technique which is widely used for solving optimization problem related to Mechanical Engineering. It is used in following ways:

- Genetic Algorithms are used in optimization of process parameter in

advance machining process like ECM, EDM, USM etc.

- Genetic algorithms are used in operation sequencing & machining parameters selection.
- Practical applications of Genetic algorithms are numerous. They are used in more unexpected areas such as designing airplane wings.
- In the shape optimization of a double-chamber muffler with an extended tube, Genetic Algorithm is used as the optimizer.
- For the planning of periodic preventive maintenance of mechanical components Genetic algorithms are used as tool.
- Genetic Algorithms are used in solving the problem of distributing a plant or facility involves calculating an as effective as possible physical distribution of the resources needed by a company to carry out its productive activities
- Genetic Algorithms are used in optimization of production planning & control activity and production scheduling activity in manufacturing.
- Genetic algorithms are ideally suited for the fixture layout optimization problem.
- Flow shop sequencing problem or assembly line problem are optimized by Genetic Algorithms.
- Genetic Algorithms are used for Assembly Planning of mechanical components.
- Genetic Algorithms are used in designing Automobile suspension system.

Process Planning Using Genetic Algorithms

Process planning is an engineering task that determines the detailed manufacturing requirements for transforming a raw material into a completed part, within the available machining resources. Process planning is the activity which is done next to product design. The output of process planning generally includes operations, machine tools, cutting tools, fixtures, machining parameters, etc. Process planning is an increasingly important part of the interface between the design and manufacturing engineering processes. The process planning for a component is modeled in a network by simultaneously considering the selection of operations, machines and operation sequence, as well as the constraints imposed by the precedence relationships between operations and available machining resources.

There are two approaches to process planning; manual-based method and

computer-aided process planning (CAPP). Manual-based method requires knowledgeable and experienced process planners to do the job. It is time consuming and the plans developed over a period of time may not be consistent. Computer-aided process planning has been developed to overcome these problems. A few benefits can be achieved by using computers in process planning, such as; accurate and consistent process plans can be produced, reduce the cost and lead time of process planning, increased productivity of planners, reduce the needs of skill planners, and can be integrated with other applications.[10]

In computer-aided process planning Genetic algorithm is used as optimization technique to generate a population of process plan. Genetic algorithm is developed to find the optimal plan. Genetic algorithm was successfully applied in a robust and general way to the process plan optimization problem. Using Genetic algorithm, the user can get various machining sequences and finally the optimized process plan in much less time. In Genetic algorithm, a search begins with an initial population. The population consists of a number of chromosomes. Each chromosome is represented by the sequence of job orders, the sequence of operations and the set of machines to be used to accomplish the operation. Among the lot of process plan generated by reproduction operator of genetic algorithm fittest will survive and when the optimum process plan is achieved.

Benefits of GAs

The advantage of the GA approach is the ease with which it can handle arbitrary kinds of constraints and objective. Genetic Algorithms are easy to apply to a wide range of problems, from optimization problems like traveling salesperson problem, to inductive concept learning, scheduling, and layout problems. Some advantages of genetic algorithms are as follows:

- It can solve every optimization problem which can be described with the chromosome encoding.
- It solves problems with multiple solutions.
- Since the genetic algorithm execution technique is not dependent on the error surface, we can solve multi-dimensional, non-differential, non-continuous, and even non-parametrical problems.
- Structural genetic algorithm gives us the possibility to solve the solution structure and solution parameter problems at the same time by means of genetic algorithm.

- Genetic algorithm is a method which is very easy to understand and it practically does not demand the knowledge of mathematics.
- Genetic algorithms are easily transferred to existing simulations and models.

Limitations

- Certain optimization problems (they are called variant problems) cannot be solved by means of genetic algorithms. This occurs due to poorly known fitness functions which generate bad chromosome blocks in spite of the fact that only good chromosome blocks cross-over.
- There is no absolute assurance that a genetic algorithm will find a global optimum. It happens very often when the populations have a lot of subjects.
- Like other artificial intelligence techniques, the genetic algorithm cannot assure constant optimization response times. Even more, the difference between the shortest and the longest optimization response time is much larger than with conventional gradient methods. This unfortunate genetic algorithm property limits the genetic algorithms' use in real time applications.
- Genetic algorithm applications in controls which are performed in real time are limited because of random solutions and convergence, in other words this means that the entire population is improving, but this could not be said for an individual within this population. Therefore, it is unreasonable to use genetic algorithms for on-line controls in real systems without testing them first on a simulation model.
- Machine and robot learning: GAs have been used for many machine-learning applications, including classification and prediction, and protein structure prediction. GAs have also been used to design neural networks, to evolve rules for learning classifier systems or symbolic production systems, and to design and control robots.
- Economic models: GAs have been used to model processes of innovation, the development of bidding strategies, and the emergence of economic markets.
- Immune system models: GAs have been used to model various aspects of the natural immune system, including somatic mutation during an individual's lifetime and the discovery of multi-gene families during evolutionary time.
- Ecological models: GAs have been used to model ecological phenomena such as biological arms races, host-parasite co-evolutions, symbiosis and resource flow in ecologies.
- Population genetics models: GAs have been used to study questions in population genetics, such as "under what conditions will a gene for recombination be evolutionarily viable?" Interactions between evolution and learning: GAs have been used to study how individual learning and species evolution affect one another.
- Models of social systems: GAs have been used to study evolutionary aspects of social systems, such as the evolution of cooperation, the evolution of communication, and trail-following behavior in ants.[11]

Other application of Genetic Algorithms

Genetic algorithms are applied to many scientific, engineering problems, in business and entertainment, including:

- Optimization: GAs have been used in a wide variety of optimization tasks, including numerical optimization, and combinatorial optimization problems such as traveling salesman problem (TSP), circuit design [11], job shop scheduling and video & sound quality optimization.
- Automatic Programming: GAs have been used to evolve computer programs for specific tasks, and to design other computational structures, for example, cellular automata and sorting networks.

Conclusion:-

Genetic Algorithm are applicable to very wide range of problem from optimization problem like travelling salesman problem, scheduling and layouts problems. This is very easy to implement. It has its own advantages and limitation as explained above. Genetic Algorithms are good tool for optimizing Mechanical Engineering problem like optimization of process parameter of wire EDM, ECM, MAF, and other cutting process, job scheduling, automatic aircraft conflict resolution, optimization of plant layout. It is also applicable on another engineering discipline problem.

If the conception of a computer algorithms being based on the evolutionary of organism is surprising, the extensiveness with which this algorithms is applied in so many areas is

no less than astonishing. These applications, be they commercial, educational and scientific, are increasingly dependent on this algorithms, the Genetic Algorithms. Work is going on for development of Genetic Algorithm for machining of cylindrical component on lathe .Its usefulness and gracefulness of solving problems has made it the a more favorite choice among the traditional methods, namely gradient search, random search and others.

References

1. Deb Kalyanmoy, Kanpur Genetic algorithms Laboratory (KanGal). Department of Mechanical Engineering, IIT Kanpur, india .
2. Holland John, Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.
3. Lautenberger .C, Rein G and . Fernandez-Pello .C, Fire Safety Journal 41 (2006), pp. 204–214.
4. Chakraborty .R .C , Fundamentals Of Genetic Algorithms, AI Course Lecture 39-40, June 01,2010.
5. Wikipedia (<http://en.wikipedia.org/wiki/Algorithm>)
6. Mathew V.Tom , Genetic Algorithm ,Assistant Professor, Department of Civil Engineering, Indian Institute of Technology Bombay, pp.1-8.
7. Goldberg E. David, Genetic algorithms in search, optimization & Machine learning, ISBN 81-7758-829-X, pp. 10-14, 61-63.
8. Goldberg E. David, Genetic algorithms in search, optimization & Machine learning, ISBN 81-7758-829-X, pp. 16-20.
9. Long-Jyi Yeh, Ying-Chun Chang, and Min-Chie Chiu ,Journal of Marine Science and Technology, Vol. 12, No. 3, pp. 189-199 (2004).
10. Ales Slak, Joze Tavcar & Joze Duhovnik , Application of Genetic Algorithm into Multicriteria Batch Manufacturing Scheduling.
11. Mathew V.Tom, Genetic Algorithm ,Assistant Professor, Department of Civil Engineering, Indian Institute of Technology Bombay, pp.15-16.
12. Kumar. Raj E, K.Annamalai , Nature Inspired Algorithm For Computer Aided Process Planning , International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 ,Vol. 2, Issue 5, September- October 2012, pp.481-484.
13. Belavendram. N , International Journal of automotive and Mechanical Engineering (IJAME) ISSN: 1985-9325 (Print); ISSN: 2180-1606 (Online); Volume 2, pp. 211-220, July-December 2010 .