Genetic Algorithm and its Applications - A BriefStudy

Joshi, Divya
diva1291@gmail.com

***Abstract*—This paper reviews and revisits the concepts, algo- rithm followed, the flow of sequence of actions and different op- erators used by Genetic Algorithm. GAs are the metaheuristic algorithm used for solving the searching problems. We will see that Genetic Algorithms has good searching properties which selects its operators depending upon the nature of the problem at hand, that is, if the problem has one optimal solution, Genetic Algorithm as well as Simulated Annealing can be used to solve itbut if a problem has more than one solution, then only Genetic Algorithm proves to be suitable and the better choice as itcreates several solutions for a problem.**

***Keywords*— Genetic Algorithm, working, components, mutation,selection, crossover, K-Point**

I. INTRODUCTION

Genetic Algorithm are metaheuristic algorithm that follows different approaches for solving a particular problem. Genetic Algorithm tends to be defined as population based algorithm. What we infer here is, [11]GA provides more than one solution, may be in hundreds, based on the size of the population.

In simpler words, Genetic Algorithm is said to support two prop- erties while solving a problem, that is, Exploration as well as Exploitation.

From the above explanation, we can infer that, as a matter of fact, GA does depend on the problem's nature. So its up to the problem's need of solution type that will decide which operator should the algorithm select. GAs tends to find better solutions for larger populations.[9]

II. GENETIC ALGORITHM

GAs was proposed in 1975 by Holland, John. Genetic Algorithms (GA) are based on search based optimization technique. GAs are metaheuristics that are based on principles of Natural Selection and Genetics, specifically based on the principle of 'survival of the fittest'[5]. GAs commonly are used for generating high quality and better solutions for problems of search and optimization. That is, GAs are frequently used to search and provide optimal or near optimal solutions to problems which might might take a lot of time. BasicGAs have frequent usage in the fields of Soft Computing, Machine Learning, Operations Research among others.

GA procedures are iterative in nature and maintains a population of individuals. These individuals are known as candidate solutions for a problem to be solved. Basically GAs can be said as composed of two main processes, first is 'Selection' for producing next generation and the second process is is called as 'Manipulation', which manipulates the selected individuals in order to create next generation by different techniques, called as crossover technique and mutation technique[8] . In GA, each iteration is called 'a generation'. So, on every generation, individuals from the current population of solutions are rated in accordance with their 'effectiveness' as being the solution to the problem at hand[11]. Taking these ratings into consideration, a population of new candidate solutions is formed using some operators such as mutation, crossover and selection[4] which are biologically inspired[3].

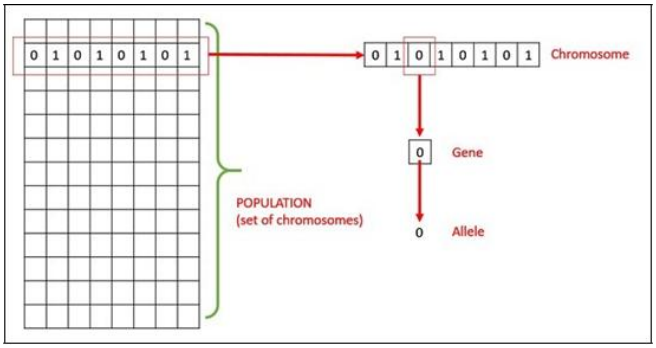

Fig. 1. GA Terminology

In GA, an individual is displayed as a string, otherwise called, 'chromosome'. They can also be considered as a solution given to a problem [10]. These strings contains characters or called 'genes' which holds specific values called 'allele'. The order of these genes on chromosome plays an important role. These specific positions on the chromosome are also called, 'loci'.

A. Components

Genetic Operators - GA searches for the best solutions using genetic operations like selection, fitness, crossover and mutation operations.

1. Fitness: This component is a positive measure of utility set for each individual in the population. The fitness value measures similarity between two individuals, quantitatively.
2. Selection: Individual in the population are given with a number of copies in the mating pool that gets used up to construct a completely new population. Therefore, the higher the fitness value of the individual, the higher the chance it receives more copies in the mating pool.
3. Crossover: Individuals are made to recombine to create new individuals, called offspring or children. A common recombination technique is one - point crossover among others.
4. Mutation: Mutation is a method to maintain diversity in the population. Each individual is changed or

mutated with a minute or very small probability, like lessor than 1.0.

Following section learns more about these genetic operators -

1) Selection Operation

Selection operation selects the elite individuals for parent chromosomes in the population from which we can generate offspring. In order to judge if an individual is an elite, **Fitness Values** are provided to each one of them. Assigning fitness value is just one of the many selection methods, which are tournament selection, roulette wheel selection, rank selection, Boltzman selection, elitism selection, steady-state selection, and others. This paper discusses three of these below:

- **Roulette Wheel Selection Method** - By this method, parent chromosome are selected based on their fitness values. The better their fitness value is, the better are chances of these chromosomes to be picked as parents. Chromosome with larger fitness area on a roulette wheel will be picked more often.Figure 2 [1]

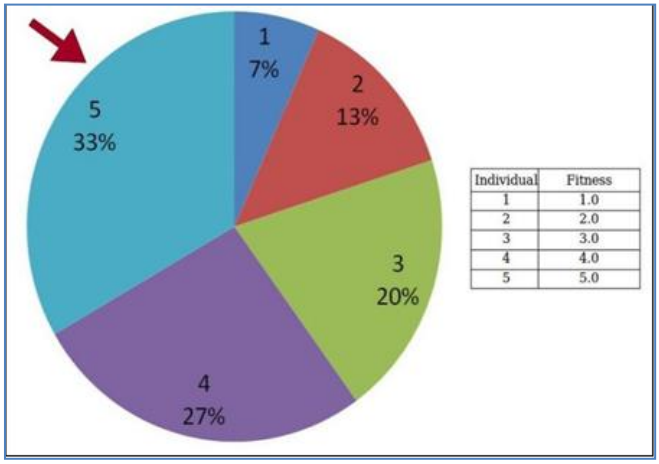

Fig. 2. Roulette Wheel Selection

Limitation: The limitation of Roulette Wheel Selection Method, that is, fitness values can differ by great measure. For example, if the fitness value of best chromosome is about 90% on the complete wheel, then the other chromosomes wont have any or very less chance of getting picked.

- **Rank Selection** - The Rank Selection Method first sorts out all the chromosomes from the population by fitness and then each of them get fitness according to this ranking. Thus, the worst chromosome will be awarded fitness equals to 1, similarly next worst as 2 and so on. Continuing this sequence, the best one will be awarded with fitness N (total number of chromosomes in the population). The probability of selecting a chromosome becomes proportional to its rank in the sorted list, and not in accordance with the fitness.

Limitation: This method can also lead convergence to slow down, because the worst chromosome do not differ that much from best chromosomes.

- **Elitism Selection** - In this method, first the best chromosomes (or a list) are copied to new population. The rest follows the classical method. As Elitism averts losing the best solution (as in the case of crossover and mutation method), it helps increase performance of GA rapidly

2) Crossover Operations

The two most common opera- tors used for the generation of offspring in a GA are crossover and mutation 3 [12]. The crossover operator, while copying selected bits from each parent produces two new offspring from two parent strings. The bit at position i in any of the two parents is copied to the bit at the position i in each offspring. The choice of which parent will be selected for contribution of the bit for i position is decided by another additional string, the crossover mask. Figure 4 [7] shows crossover operator. There are basically three crossover operators, namely uniform crossover, two-point and single-point.[6]

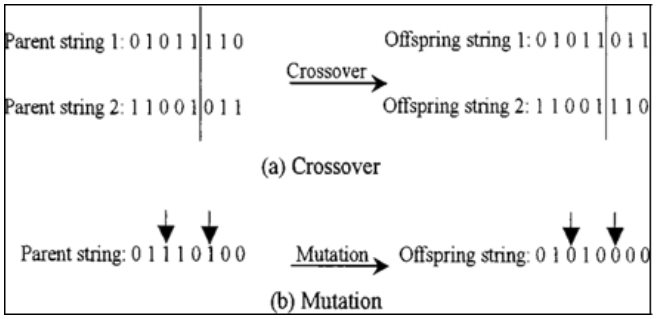

Fig. 3. Crossover and Mutation Operations

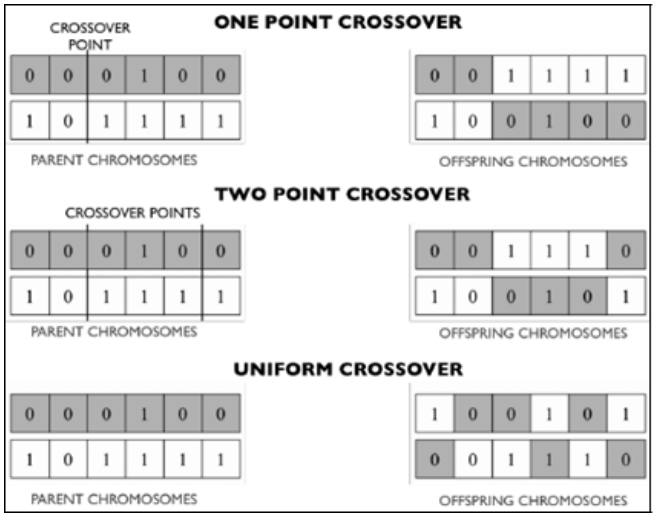

Fig. 4. Crossover Operations

3) Mutation Operations

The mutation operator produces offspring using just one parent chromosome. It creates minimal random changes to bit string by selecting one bit at random and then its value is changed. Mutation is often seen performed after Crossover Operation as shown in Figure 3.

4) Fitness Function

It defines the standard for ranking possible hypotheses and selecting them (by probability) in order to include them in next gen. Often other standards may be considered, like for example, the generality or the complexity of the rule.

B. Working Principle

1. Start: A random population with 'n' chromosomes is gener- ated.
2. Fitness: A fitness function, f(x), is calculated for each chro- mosome, 'x'.
3. New Population: Starts iterating over generation of new population until a stopping criteria is encountered.
4. Selection: Two parent chromosomes are selected from the population in accordance to their fitness rate calculated above.

   Note: The higher the f(x) rate, probability of selection increases.
5. Crossover: Here cross over between selected parent chromo- some is done to create new child or offspring.

   Note: If this step is skipped, the child is exact copy of the parents.
6. Mutation: Each offspring is mutated at each *loci* point.
7. Accepting: The newly created offspring are placed in the newset of population.
8. Replace: The newly generated population is used for further execution of the algorithm.
9. Test: If the end condition or stopping criteria is met, the execution is halted and the optimal solution is returned from the current population.
10. Loop: Else, if the stopping criteria is not satisfied, the execution returns to second step to evaluate fitness for newlycreated chromosomes.

C. Algorithm

GENETIC ALGORITHM[11]

- *begin*
- creating initial population of size P
- *repeat*
- select $parent_1$ and $parent_2$ from P.
- offspring = crossover($parent_1$,$parent_2$)
- mutation(offspring)
- update P
- **until** stopping criteria is met
- provide best result

end

The steps involved in a Genetic Algorithm are listed down in form of a flowchart [Figure 5] [2]

D. Working Example

For simplicity, in this paper we are taking Binary Coded Genetic Algorithm example to understand the working, selection , mutation and selection procedures in a better way.

Here we will select two parents, P 1 and P 2 first to create new children or chromosomes.

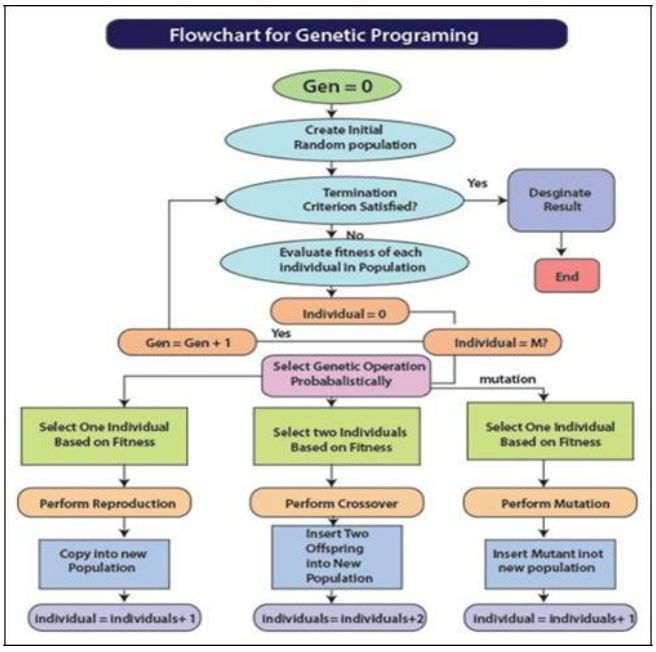

Fig. 5. GA Algorithm Flowchart

Chromosomes	Fitness Values
01100010	20
10101100	15
00011101	9
11001011	12

Fig. 6. Chromosomes Fitness Values

1) Selection

Following is the initial population taken for this example. Select parents according to their fitness level or value.

Parent	Before Cross Over							
P_1:	0	1	1	0	0	0	1	0
P_2:	1	0	1	0	1	1	0	0

K-Point

Fig. 7. Parents Before Crossover

2) Crossover

Following 'Single Point Crossover Schemes',now pick a random value, K-Point, say k. This single cross overpoint s selected on both the selected parents. All the data in both the strings, beyond this k point is swapped between two parents. In our example, we have selected K-point at **third bit**.

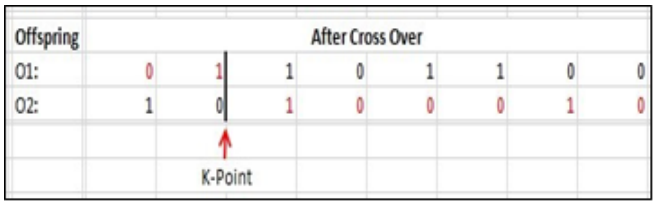

Fig. 8. Parents After Crossover

The strings generated after the cross over are the chromosomes of the offspring generated.

Offspring	After Cross Over					
O1	1	0	1	1	0	0
O2	1	0	0	0	1	0

Fig. 9. Offspring After Mutation

3) Mutation

For this step, lets select offspring, O2 gen- erated above and apply mutation on it. When MP is Mutation Probability, then:

if

MP = 0: bit remains same MP = 1: bit is flipped

Result: MO = Mutated Offspring.

Offspring	After Mutation					
O2 Before Mutation	1	0	0	0	1	0
Mutation Probabilty	0	1	0	0	0	1
O2 After Mutation	1	1	0	0	1	1

Fig. 10. Offspring After Mutation

Final Mutated Offspring that consists of the most fittest value:

[110011] –

E. Advantages

GAs have various advantages which have made them immenselypopular. These include:

- They do not require a derivative information, which normallyisn't available in real world problems.
- It is much faster and much more efficient as compared to othertraditional algorithms.
- It has parallel capabilities.
- Both, continuous as well as discrete functions are optimizedby GAs.
- It also is able to optimize multi objective problems.
- Instead of just one solution, it provides with a list of generated"good" results or solutions.
- It tends to always reach an answer to a problem at hand, eventually, which also keeps getting better over generations.
- It is the approach most suitable and useful for large search spaces or/and large number of parameters are present or involved.

III. LIMITATIONS

Like any technique, GAs also suffer from a few limitations. These include:

1. Though GAs are well suited for most search optimization problems but there are some problems where GAs are not that suitable like the problems that are simpler in nature and for which derivative information is present.
2. Computationally Expensive: GAs can prove to be computa- tionally expensive for some of the problems as it needs to keep calculating 'Fitness Value'.
3. Since GAs are stochastic in nature, there is no guarantee that it will definitely provide good quality result or an optimal one.

GAs may not provide an optimal result at all, if they are not implemented with caution and properly.

IV. CONCLUSION

Genetic Algorithms proved to be better in finding areas of complex and real world problems. Genetic Algorithms are adaptive to their environments, as this type of method is a platform appearing in the changing environment. In Present these algorithms are more applicable. Several improvements must be made in order that GAs could be more generally applicable.

REFERENCES

[1] Elisa Amorim et al. "Comparison between Genetic Algorithms and Differential Evolution for Solving the History Matching Problem". In: vol. 7333. June 2012, pp. 635–648. ISBN: 978-3-642-31124-6. DOI: 10.1007/ 978-3-642-31125-3 48.

[2] Artificial Neural Network - Genetic Algorithm. URL: https://www.javatpoint.com/artificial-neural-network- genetic-algorithm.

[3] Genetic algorithm. 2021. URL: https://en.wikipedia. org/wiki/Genetic algorithm.

[4] David E. Goldberg. Genetic Algorithm in Search, Optimization and Machine Learning. 1989.

[5] Holland John H. "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence". In: (1975).

[6] Haldurai Lingaraj. "A Study on Genetic Algorithm and its Applications". In: (2016).

[7] Usama Mehboob et al. "Genetic Algorithms in Wire- less Networking: Techniques, Applications, and Is- sues". In: Soft Computing 20 (June 2016). DOI: 10 . 1007/s00500-016-2070-9.

[8] Noraini Razali and John Geraghty. "Genetic Algo- rithm Performance with Different Selection Strategies in Solving TSP". In: vol. 2. Jan. 2011.

[9] Sinan Salih. Simulated Annealing vs genetic algo- rithm. 2018. URL: https : / / www . researchgate . net/post/Simulated Annealing vs genetic algorithm/ 5c0f372a36d2356a0b5cc06b/citation/download.

[10] Chathurangi Shyalika. "An Insight to Genetic Al- gorithms". In: (2019). URL: https : / / medium . datadriveninvestor . com / an - insight - to - genetic - algorithms-part-i-a7f5a5d6d214.

[11] James T. Cain Theodore W. Manikas. Genetic Algo- rithms vs. Simulated Annealing: A Comparison of Ap- proaches for Solving the Circuit Partitioning Problem. 1996, pp. 2–7.

[12] Cheng-Xiang Yang et al. "Two-Stepped Evolutionary Algorithm and Its Application to Stability Analysis of Slopes". In: Journal of

Computing in Civil Engineer- ing - J COMPUT CIVIL ENG 18 (Apr. 2004). DOI: 10.1061/(ASCE)0887-3801(2004)18:2(145).

AUTHORS PROFILE

Divya Joshi have received her Bachelor's in Computer Science and Engineering from Kurukshetra University. She is currently pursuing her Master's from Chandigarh University in Artificial Intelligence and Machine Learning. Her research interest includes Natural Language Processing, Holography, Augmented Reality and Computer Vision.