©2017 IJSRST | Volume 3 | Issue 8 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

A Review on Genetic Algorithm and Its Applications

Anju Bala
Research Scholar, Department of computer science and applications, M. D. University, Rohtak, Haryana, India

ABSTRACT

The wide spread use of Artificial Intelligence makes the most useful genetic algorithm as a heuristic search method. Genetic algorithms are one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work in any search space. Genetic algorithm will be able to create a high quality solution. Genetic algorithm use the principles of selection and evolution to produce several solutions to a given problem Genetic algorithm (GA) is a searching technique used in computing to find true or approximate solutions to optimization and search problems. Genetic algorithm (GA) are categorized as global search heuristics. GA is a good heuristic search for combinatorial problem like TSP, pen movement of a plotter, real world routing of school buses, delivery trucks and posted carriers. In this paper we present the genetic algorithm, its evolutionary cycle, basic principles., basic operator, working mechanism, algorithm, advantages and its applications.

Keywords : genetic algorithm, evolutionary cycle, principles, operators, advantages and its applications.

I. INTRODUCTION

Genetic algorithms are one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work in any search space. Genetic algorithm will be able to create a high quality solution. Genetic algorithm use the principles of selection and evolution to produce several solutions to a given problem.

A GA is heuristic, which means it estimates a solution. In fact, most real-life problems are like that: you estimate a solution rather than calculating it exactly.

For most problems you don't have any formula for solving the problem because it is too complex, or if you do, it just takes too long to calculate the solution exactly. An example could be space optimization - it is very difficult to find the best way to put objects of varying size into a room so they take as little space as possible. The most feasible approach then is to use a heuristic method.

Genetic algorithms are different from other heuristic methods in several ways. The most important difference is that a GA works on a population of possible solutions, while other heuristic methods use a single solution in their iterations. Another difference is that GAs are probabilistic (stochastic), not deterministic.

In the computer science field of artificial intelligence, a **genetic algorithm (GA)** is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problem. Genetic algorithms belong to the larger class of evolutionary algorithm(EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation , selection , and crossover[1,2].

Genetic algorithms find application in bioinformatics, phylogenetics, computational science, engineering ,economics, chemistry, mathematics, manufacturing, physics and other fields.

GA directed search algorithms based on the mechanics of biological evolution. GAs use concept of "natural **selection" and "genetic inheritance" (Darwin 1959). It is Developed by John Holland, university of Michigan (1970)** in U.S.A. It Provide efficient effective techniques for optimization and learning applications. It Widely used in business, scientific and engineering circle. GA class of probabilistic optimization algorithm. A genetic algorithm (GA) is a search techniques used in computing to find true or approximate solution to optimization and search problem. Genetic algorithms are categorized as global search heuristic or GA is good heuristic search for combinatorial problems[3]. Ex.TSP, pen movement of a plotter, real world routing of school buses, delivery trucks and posted carriers.

Genetic algorithm tend to thrive in an environment in which there is a very large set of candidate solutions and in which the search space is uneven and has many hills and valleys. Genetic algorithm will do well in any environment.

There are some key terms used in GAs:

- ✓ Individual - any possible solution
- ✓ Population – group of all individuals
- ✓ Search space – all possible solutions to a problem
- ✓ Chromosomes – blue print for an individual
- ✓ Trait – possible aspect of an individual
- ✓ Allele – possible setting for a trait
- ✓ Locus – the position of a gene on the chromosomes
- ✓ Genome – collection of all chromosomes for an individual

Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover (also called recombination). Traditionally, solutions are represented in binary as string of 0's and 1's, but other encoding are also possible[4,5].

The evolution starts from set of solutions of randomly generated solution to a problem and happens in the generations. In each generation, the fitness of every solution in the set of solutions is evaluated. Multiple solutions are selected from the current set of solution (based on their fitness) and modified (recombined and possibly mutated) to form a new solution. The new solution is then used in next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or satisfactory fitness level has been reached for the solutions[8].

II. EVOLUTIONARY CYCLE

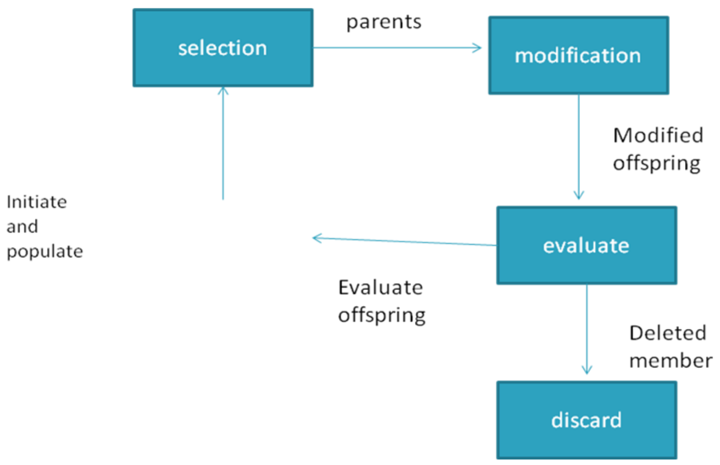


Figure 2. Evolutionary cycle of GA.

1. The selection chooses the fittest individuals. You will need a method to calculate this fitness. Let's use the space optimization example. The fitness method simply calculates the amount of free space each individual/solution offers. The best are selected for further iteration.
2. The cross-over is the method for combining those selected individuals into new individuals. Remember that the individuals are simply

strings of values. The cross-over splits up the "parent" individuals and recombines them. Here's an example of how two "parents" cross over to make two "children":

3. The mutation simply adds some noise to "genes" of the individuals (usually the "children"). It is a way of varying the "gene pool" to provide some protection against "in-breeding[11]."

III. BASIC PRINCIPLES

- ✓ Coding or representation String with all parameters
- ✓ Fitness function Parent selection
- ✓ Reproduction Crossover Mutation
- ✓ Convergence When to stop

IV. BASIC OPERATOR

Encoding : The process of representing the solution in the form of a string that conveys the necessary information

Fitness function : A fitness function quantifies the optimality of a solution so that that particular solution may be ranked against all other solution[11].

Recombination : The process that determines which solution are to be preserved and allowed to reproduce and which ones deserve to discard.

Cross Over : It is the process in which two chromosomes(strings) combine their genetic material(bits)to produce a new offspring which possesses both their characteristics.

Mutation : it is the process by which a string is deliberately changed so as to maintain diversity in the population set[12].

In computer mutation means change in code/class that effect testing.

eg. Point mutation : changes in one or few code.

-substitution

- insertion

- deletion

V. WORKING MECHANISM OF GAS

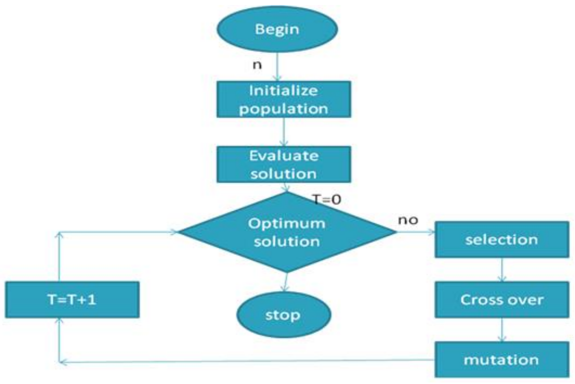


Figure 3. Flow chart of GA.

VI. BASICS OF GENETIC ALGORITHMS

The most common type of genetic algorithm works like this: a population is created with a group of individuals created randomly. The individuals in the population are then evaluated. The evaluation function is provided by the programmer and gives the score based on how well they perform at the given task. Two individuals are then selected based on their fitness, the higher the fitness , the higher the chance of being selected . These individuals then "reproduce" to create one or two offspring, after which the offspring are mutated randomly. This continues until a suitable solution has been found or a certain number of generations have passed, depending on the need s of the programmer[4,9].

Simple genetic algorithm:

```
Simple_Genetic_Algorithm( )
    {
Initialize the population;
Calculate fitness function;
While(Fitness Value != Optimal Value)
    {
Selection;
Crossover;
Mutation;
Calculate Fitness Function;
    }}
```

The Algorithms

1. Randomly generate an initial population M(0)
2. Compute and save the fitness u(m) for each individual m in the current population M(t)
3. Define selection probabilities p(m) for each individual m in M(t) so that p(m) is proportional to u(m)
4. Generate M(t+1) by probabilistically selecting individuals from M(t) to produce offspring via genetic operators
5. Repeat step 2 until satisfying solution is obtained.

Nature of computer mapping:

Nature	**Computer**
Population	Set of solution
Individual	Solution to a problem
Chromosome	Encoding for a solution
Gene	Part of the encoding of a solution
Reproduction	Cross Over

VII. ADVANTAGES OF GA

1. Global search method: GAs search for the function optimum starting from a point of the function domain not a single one. This characteristics suggest the GAs are global search.
2. Blind search method: GAs only use the information about the objective function. They don't require other information.
3. GAs use probabilistic transition rules during iteration, unlike the traditional method that use fixed transition rules. This make them more robust and applicable to a large range of problem.
4. Concept is easy to understand.
5. Always an answer; answer gets better with time[13].

Herein, we will examine GAs as a number of different things:

- ✓ GAs as problem solvers
- ✓ GAs as challenging technical puzzle
- ✓ GAs as basis for competent machine learning
- ✓ GAs as computational model of innovation and creativity
- ✓ GAs as computational model of other innovating systems
- ✓ GAs as guiding philosophy

VIII. WHO CAN BENEFIT FROM GA

Nearly everyone can gain benefits from Genetic Algorithm , once he can encode solutions of a given problem to chromosomes in GA , and compare the relative performance(fitness) of solutions. An effective GA representation and meaningful fitness evaluation are the keys of success in GA applications. The appeal of GAs comes from their simplicity and elegance as robust search algorithms as well as from their power to discover good solutions rapidly for difficult high-dimensional problems[13]. GAs are useful and efficient when

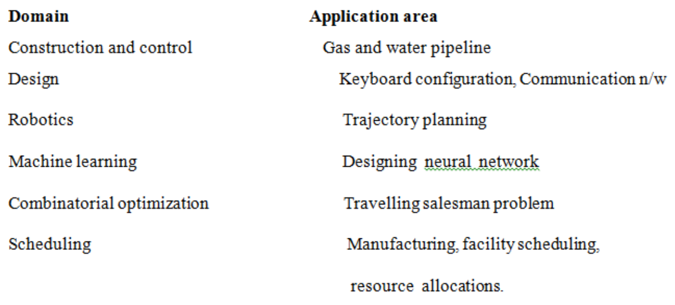
- ✓ The search space is large, complex or poorly understood.
- ✓ Domain knowledge is scarce or expert knowledge is difficult to encode to narrow the search space.
- ✓ No mathematical analysis is available.
- ✓ Traditional search method fails.

IX. APPLICATION OF GENETIC ALGORITHM

- ✓ Optimization: GAs have been used in a wide variety of optimization tasks, including numerical optimization, and combinatorial optimization problems such as travelling salesman problem(TSP) circuit design, 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..
- ✓ 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 behaviour in ants..

Application area of GA:



Domain	Application area
Construction and control	Gas and water pipeline
Design	Keyboard configuration, Communication n/w
Robotics	Trajectory planning
Machine learning	Designing neural network
Combinatorial optimization	Travelling salesman problem
Scheduling	Manufacturing, facility scheduling, resource allocations.

X. CONCLUSION

In recent trend automatic generating the test cases attracts many researchers by using genetic algorithm. Testing the object oriented programs has been addressed from different view point by many researches, most of them concerned with the problem related to the generation of automatic test cases. By which human and cost effort are minimized. Using Evolutionary Genetic Algorithm, we can create optimal valid test cases. We can find out the fitness value of program. Genetic algorithm also used in combinatorial problem.

XI. REFERENCES

[1]. Genetic Algorithms in Engineering and Computer Science , edited by G.Winter, J.Periaux & M.Galan, published by JOHN WILEY & SON Ltd. in 1995.

[2]. [Louis 1993] Genetic Algorithms as a Computational Tool for Design, by Sushil J. Louis, in August 1993

[3]. Foundatiions of Genetic Algorithms Volume 3, by L.Darrell Whitley & Michael D.Vose, in 1995 published by Morgan Kaufmann Publishers, Inc.

[4]. Algorithms and Complexity, by Herbert S.Wilf, in 1986 published by Prentice-Hall, Inc.

[5]. R. P. Pargas, M. J. Harrold, and R. R. Peck, "Test Data Generation Using Genetic Algorithms"

Journal of Software Testing, Verifications and Reliability, vol. 9, pp. 263-282, 1999.

[6]. U. Buy, A. Orso, and Pezzè, "Automated Testing of classes," In Proceedings of the International Symposium on Software Testing and Analysis (ISSTA 2000), August 2000.

[7]. V. Martena, A. Orso, and Pezzè, "Interclass Testing of Object Oriented Software," In Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems (ICECCS2002), 2002.

[8]. P. Tonella, "Evolutionary testing of classes," In Proceedings of the International Symposium on Software Testing and Analysis (ISSTA 2000), pp. 119-128, 2004.[22] S.

[9]. Wappler and F. Lammermann, "Using evolutionary algorithms for the unit testing of object-oriented software," In Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO 2005), pp. 1053-1060, 2005.

[10]. Y. Cheon, M. Y. Kim, and A. Perumandla, "A Complete Automation of Unit Testing for Java Programs," Proceedings of the 2005 International Conference on Software Engineering Research and Practice (SERP '05), pp. 290-295, 2005.

[11]. Y. Cheon and M. Kim, "A Fitness Function for Modular Evolutionary Testing of Object-Oriented Programs" In Genetic and Evolutionary Computation Conference, pp. 1952-1954, 2006.

[12]. M. Y. Kim and Y. Cheon, "A Fitness Function to Find Feasible Sequences of Method Calls for Evolutionary Testing of Object-Oriented Programs," To appear in International Conference on Software Testing, Verification, and Validation, Norway, April 9-11, 2008.

[13]. J. Holland, Adaptation in Natural and Artificial Systems, ISBN 0 472 08460 7. University of Michigan Press, Ann Arbor, MI, 1975.