# Analysing Tuning Performance Of Evolutionary Algorithm For Search Based Software Testing

**Sudha Rajesh[1], R.B.Sheetal[2], C.Swathi chander[3]**

[1]*Assistant Professor ,Jeppiaar Srr Engineering College, Padur*
[2,3] *Student ,Jeppiaar Srr Engineering College, Padur*

---

**Abstract -** *Software testing is performed to check whether the completed software meets defined requirements. Testing usually carried out to detect fault in the developed software in order to guarantee its quality. Test data generation plays an important role in software testing. Metaheuristic techniques generates the test data dynamically. Using this technique solves the optimization problem faced in software testing. Solving this optimization problem is known as Search Based Software Testing. Genetic Algorithm is a metaheuristic techniques that are used to solve the Search Based Software Testing problem. Genetic Algorithm produces globally optimized solution. Using Genetic Algorithm in testing improves efficiency and effectiveness in generating test data. Further to analysis the quality of the solution or test data obtained from Genetic Algorithm, investigated the performance.*

***Key Words*: Genetic algorithm (GA) ,Software under test (SUT), Search based software testing(SBST), Evolutionary algorithms(EA).**

## 1. INTRODUCTION

The origin of evolutionary algorithm was to imitate the important process taking place in the natural evolution. Charles Darwin was the first person to discuss about this. Evolution is a process that occurs due to chromosomes rather than over organisms. The most common process taking place in evolution is mutation and recombination during the reproduction phase. Using these features an evolutionary algorithm was developed. Evolutionary algorithm is an iterative an stochastic process that operates on a set of individuals (population).Each individual represents a prospective problem to be solved. This solution is obtained by means of encoding or decoding mechanism. Initially the population is generated randomly (perhaps with the help of construction heuristics) and each individual in the population is assigned a fitness value. The fitness value is a type of calculated which has a predefined condition using which the best value is being selected. Using the fitness function the goodness of population is calculated. This value is the qualitative information the algorithm uses to guide the search. The genetic algorithm depends on use of selection, cross over and mutation operation. Replacement is by generating new individuals. Intuitively the genetic algorithm proceeds by creating successive generations of better and better individuals by applying very simple operations. The search is only guided by the fitness value associated to every individual in a population. The value with lesser fitness value be preferred for reinsertion. The problem is the fitness function that for every individual is encharged of assigning the fitness value. Genetic algorithms intensively using such local search mechanism are termed Memetic Algorithms. The EAs exploration compares quite well against the rest of search techniques for a similar search effort. Exploitation is a more difficult goal in evolutionary algorithms but nowadays many solutions exist for evolutionary algorithms to refine solutions. Genetic algorithms are currently the most prominent and widely used computational models of evolution in artificial -life systems. These decentralized models provide a basis for understanding many other systems and phenomena in the world. Researches on genetic algorithm will illustrative examples in which the genetic algorithm is used to study how learning and evolution interact, and to model ecosystems, immune system, cognitive systems and social systems.

## 2. EXISTING SYSTEM

P.Harman and p.mcminn[1] explained the limitation of local search algorithm to generate test data which was gets struck in local optimal solution and genetic algorithm was explained. Then theoretical exploration of the global search technique embodied by GA and behaviour of global search algorithm and local search algorithm are compared.

Korel et al.[2] states the test data generation procedure worked on an instrumented version of the original program without the need for a straight-line version to be produced. The search targeted the satisfaction of each branch predicate along the path in turn. To execute some desired path, the program is initially executed with some arbitrary input. If during execution an undesired branch is taken - one which deviates from the desired path - a local search for program inputs is invoked, using an objective function derived from the predicate of the desired, alternative branch. This objective function describes how close the predicate is to being true. The value obtained is referred to as the branch distance.

D.Goldberg and k.Deb[3] states tournament selection is a noisy but fast rank selection algorithm. The population does not need to be sorted into fitness order. Two individuals are chosen at random from the population. A random number is then chosen. If r is less than p (where p is the probability of the better individual being selected), the fitter of the two individuals 'wins' and is chosen to be a parent, otherwise the less fit individual is chosen. The competing individuals are returned to the population for further possible selection. This is repeated N times until

the required number of parents have been selected. In all probability, every individual is sampled twice, with the best individual selected for reproduction twice, the median individual once, with the worst individual remaining unselected. The resulting selective bias is dependent on p. If p = 1, then in all probability a ranking with a bias of 2.0 towards the best individual is produced.

P. McMinn[4] states that metaheuristic search technique can be applied in search based software testing problem.

Xue.Y et al.[5] has investigated the use of test-suite reduction technique for regression testing using GA. Regression testing is carried out efficiently by means of selecting minimum test cases to check the modified code. GA uses the operators: reproduction, crossover and mutation to check on the modified code with boundary coverage analysis and generates test suite. GA already holds the original test suite of unmodified code so by using its operations on the modified code it generates reduced test suites.

## 3. Proposed System

The architecture diagram for the proposed system is represented in the figure. This architecture is made up of number of components were each and every component has an important function to perform. Initially it starts with SUT which consist of the code to be tested and the test data adequency criteria contains the target followed by the instrumented code .The instrumented code is equivalent to the SUT but it consist of predicate node.---- then the genetic algorithm components will perform its functions and then the performance is analyzed. The instrument code's is given as input to the genetic algorithm part where iterative process takes place and the test data is generated. Next step is a genetic algorithm operations.6

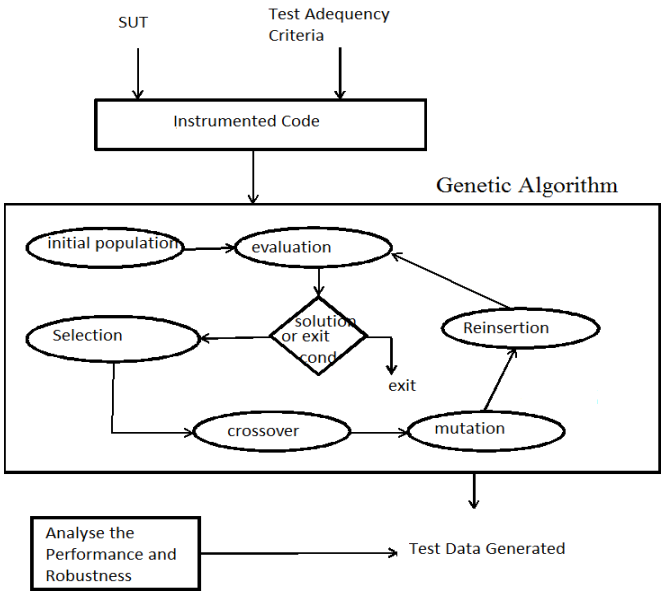


**Fig -1**: Architecture diagram

### 3.1 FITNESS FUNCTION

The fitness function combines a measure known as the approach level with the branch distance. The approach level is a count of predicate node where not encountered in the path executed to reach target node. The following equation 1 and 2 is used to calculate the fitness function.
To calculate fitness value

Fitness=approach level+ normalize (branch distance).....(1)
Normalize (branch distance)=1-pow(1.001,branch distance).....(2)

Branch distance are calculated using koreal branch prediction table. Each predicate node is numbered in decreasing order from starting node to target node that is from n to 0. So one value is subtracted from the approach level when it reaches the next node in the path to reach the target node. So on reaching target node without any deviation from starting node approach level node is 0. If the test data deviated from target ,the fitness value will be higher. Lower numerical fitness value represent fitter input data.

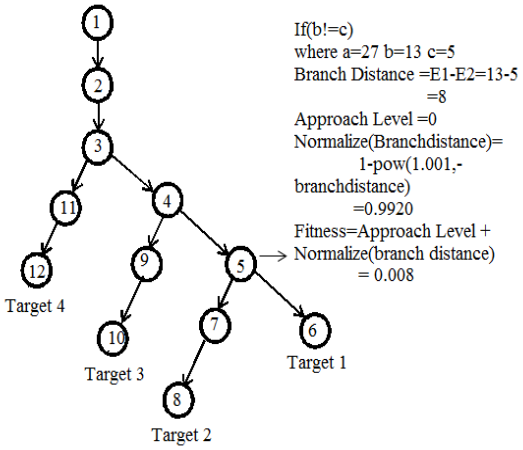


**Fig -2**:Fitness function calculation

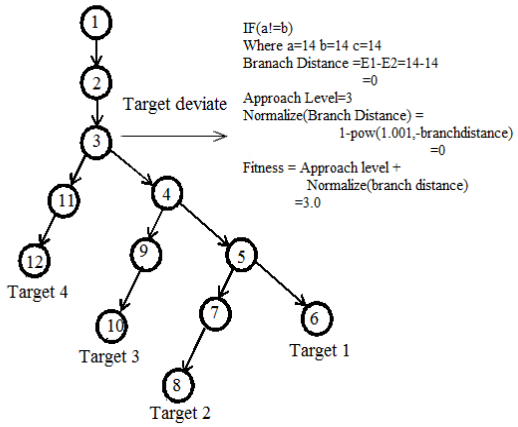


**Fig -3**:Fitness function calculation

### 3.2. GENETIC ALGORITHM

i. Selection:-selection is used to select a Individual from a population used to breed new set of population.

ii. crossover:- it is used to combine two chromosome by interchanging or flipping bits to generate new population.

iii. mutation:- it randomly introduce a bit in chromosome.

Test data is generated after exit of genetic operation. Then test data is investigated based on the performance of the solution obtained.

### 3.3. ANALYSE THE PERFORMANCE

TRIANGLE CLASSIFICATION PROGRAM- This program classifies a triangle on the basis of its input sides as non-triangle or a triangle, i.e., isosceles, equilateral or scalene .It takes three real inputs all of which represent the sides of the triangle. Its CFG has 12 Nodes with 8 predicate nodes. The maximum nesting level is 5.Scalene triangle is chosen as a target. For triangle classification program, GA starts discovering the best fitness value in each generation. At each generation the best fitness value is discovered and it is plot as a graph. The graph shows that gradual decrease in fitness in each generation and at certain point the fitness value is become constant. The graph clearly shows that GA start with higher fitness value and after each generation it starts discovering the best fitness value. At certain point the fitness value remains unchanged, so the GA takes fitness value as the best fitness value.

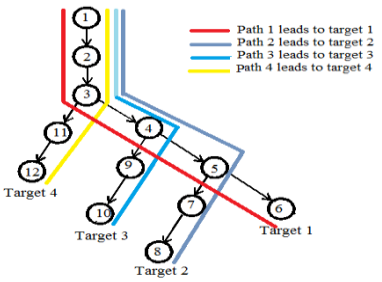


**Fig -4**: Control dependency graph

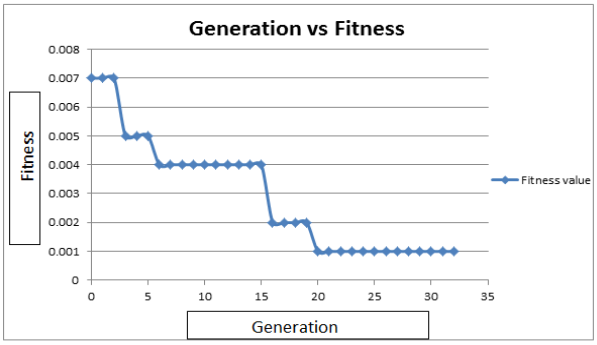


**Fig -5**: Generation vs Fitness graph

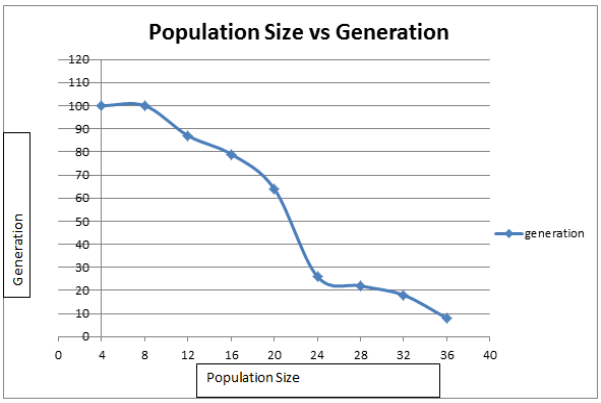


**Fig -6**: Generation vs population Size

## 4. CONCLUSIONS

To conclude, Genetic algorithm was applied on the software under test using randomly generated initial population. For each individual population the fitness value was calculated based on the fitness function. Set of lowest fitness value is selected by applying the tournament selection process and this resultant is eligible for the next operation, while the remaining population were eliminated. After applying selection process, crossover operation was performed and then sequentially mutation operation was carried out. By repeating this process, at certain stage fitness value remains constant and this concludes that optimal test data was reached. Further, this procedure was applied to three software under test and their performance were investigated.

## REFERENCES

[1] M.Harman, P.McMinn," A theoretical and empirical study of search based testing: local global and hybrid search", IEEE Transactions on Software Engineering, 36(2), 2010, pp. 226-247.

[2] B.Korel, " Automated Software Test Data Generation", IEEE Transaction on Software Engineering, 16(8), 1990, pp. 870-879.

[3] k.Deb, D.Goldberg, " A comparative analysis of selection schemes used in genetic algorithms", Foundations of Genetic Algorithms, 1991, pp. 69-93.

[4] P.McMinn, " Search-Based Software Test Data Generation: A Survey",Journal of Software Testing, Verification and Reliability, 14(2), 2004,pp. 105-156.

[5] M.A.Xeng-ying, H.E.Zhen-feng, B.Sheng,YE.quing, " A genetic Algorithm for Test-suite Reduction", IEEE International Conference,1, 2005,pp. 253-262.