An Algorithm For Evaluating Variation Of Cross Over And Mutation Operator In Genetic Algorithm (GAVCOMO)

Seema Sharma, Dr. Shaveta Bhatia

Abstract: A Genetic Algorithm is a heuristic approach which has been inspired by Charles Darwin's theory of biological natural evolution. The algorithm works on the process where the most fitted individuals are selected for reproduction from the population to produce offspring of the next generation. There are a variety of optimization methods which have been applied in many subjects in which Genetic Algorithm is one of the optimization methods which have been used in software testing, Data mining, Neural Network fuzzy System and many more. One of the issues in utilizing Genetic Algorithm is which recombination operator gives the better results. The point of this paper is to show the impact recombination operator on the execution of Genetic Algorithm. This paper presents the comparative study of all types of crossover operators i.e. single point, multipoint and uniform. The mutation operator has also been taken to enhance the search space. In order to accomplish this aim Software testing has been taken as a subject area where the experiments have been done on C program for code testing. The crossover operator that has the best average performance has been taken for the creating the next generation.

Index Terms: Genetic Algorithms, Crossover Operators, Optimization, Mutation Operator, Code testing, Software testing

1 INTRODUCTION

Every software ought to be of high quality and bug free. So as to achieve the errand of the development of such software, we have to have sound and powerful testing procedures. Whatever the approach have been taken into account it should be able to uncover the maximum faults. Software testing is the most basic element of the recipe that makes the software flavorful. Software testing begins from the earliest starting point of an undertaking projects and proceeds till the end. It not just discovers bugs, errors and mistakes yet in addition diminishes the general expense of a task. The general expense of a venture is decreased if the, bugs, mistakes, etc would have found in the early stage. One little bug can be cause of the failure of software. Therefore for delivery a good quality product; we need Software Testing in the Software Development Process. There are two classifications of software testing techniques, static and dynamic. Static technique does not require many resources as they do not confide on the actual execution of a program. On the other hand dynamic testing requires more resources because of the dependency on the actual execution of program. In dynamic testing the result of actual output and the expected output must be compared for finding the status of module under test. Status of module means having bugs or error free modules. Testing can be done in two ways: Manual Testing: It is the process to find the bugs manually. Since the tester should have all the requirements of the customer therefore he ensures that the software will work as per the mentioned requirements. In this process all test cases execute manually and the reports are generated without using any automated tools. Therefore it is time consuming process. Automation Testing: It is the process for executing test cases and generates the results by using automated tools. There are some well-known tools are there in market for testing like QTP/UFT and Selenium. Since all the test cases run through the automated tools therefore it consumes less time. After the extensive literature review, we came to know that in order to generate the best test case it should cover maximum code coverage in comparatively less time and cost. This bi-target benchmark can be accomplished by stepping through to generate the test cases for every individual way [6]. The task of generating test cases can be done through the concept Genetic Algorithm with variation of crossover and mutation operator. (GAVCOMO). The model was proposed for evaluating the variation of crossover and mutation operators that can bring us better solutions. The proposed algorithm was novel and has been implemented [8]. The paper discusses the implementation of the model and the observation therein. It may state the Genetic Algorithm has been used for the purpose of optimization. The results are promising. This works paves the way of the use of most important variable in generic test cases. The paper has been classified out as pursues: Part 1 is the literature review, Part II is the proposed model, Part III is implementation, Part IV is Results, and Part V is Conclusion.

2 LITERATURE REVIEW

A broad survey has been piloted to examine the proposed methodology till now and discover the research gaps in the accessible works. A new strategy has been proposed on the basis of existing work. In 1999, Pargas et al., has developed the technique using Genetic ALgorithm for generating test data for executing every given statement, branch, path, or definition-use pair at least once. The technique was used parallel processing for improving the QOS (quality of services) of the algorithm. The name of the framework developed was TGen [26]. Six programs had been taken for the verification of the proposed work and the outcomes was matched with the Random Test Data Generation approach. The Statement coverage and branch coverage has been used as a parameter for comparing proposed work with random test data generation. Wegner et al., proposed a work for generating...
fully automatic test cases for the structural programming. The testing methodology were based on vertex-oriented, edge-oriented, vertex-edge-oriented and vertex-vertex-oriented. Different fitness function were introduced for different testing methodology. The experiments have been taken in to account for C programming language and the outcomes was compared with Random test data generation approach.

Gong et al. in 2011, proposed the algorithm for generating the test cases based on grouping of the paths. The basic idea of the work was, first make the groups of the target paths based on same traits and each group converted the complex optimization problem to simpler one. Now fitness function was designed according to the group for generating test data. The technique described increase the efficiency for creation of test data [14] In 2008, Sofokleous et.al., developed an algorithm which combines the two techniques for optimization: Batch Optimistic (BO) and Close up (CU) algorithms. The employments uses of two optimization methods were based on the guidelines of edge and condition inclusion criteria. The Authors introduced two weights w1 and w2 that defines the impact of predicate nodes and edges for complete execution of path. These weights were used to calculate the fitness function. Only specified weight would be taken for evaluation of fitness, as it lead to insignificant fitness function if the weights were choose randomly. [28]. Test cases have been generated by Spooner and Miller [36] used the numerical expansion technique rather conception rather than symbolic implementation of given path. The concept is used for numerical programs or the programs that abide floating or doublet data types [21]. In 1976 Clarke[5] had developed an algorithm for automatic generation of test data. The concept was based on the symbolic execution (SE) of the path of given program code that defines a set of conditions of the input variables of source code. The symbolic execution uses depiction of path and useful data of program unit. The set of conditions were solved by an inequality solver and thus generates test data. The Programs of ANSI Fortran were taken as a experiments [4]. For producing automatic Test Data generation, different techniques have been developed till now in which numerous strategies depend on heuristic approach. In 2004 a extensive survey was conducted by McMinn[19],which was test case generation on the basis of search. It was found that various papers have used Genetic Algorithms (GAs) to solve the problem of Automatic creation of test data [36, 27, 23, 35, 15, 28, 20, 29, 32, 11, 12, 31]Xanthakis et al. in 1992 proposed the algorithm which was based on variable length genetic algorithm for optimization. The software path cluster was introduced and executes the based on variable length genetic algorithm for optimization. The experiments have been taken in to account for C programming language and the outcomes was compared with Random test data generation approach.

3 GENETIC ALGORITHM

Since in evolutionary computation, a family of algorithms has been implemented in realistic problems and genetic algorithm (GA) is one of powerful algorithm in the entire family of evolutionary algorithm. It is a feasible, vigorous, heuristic approach and search methodology. The term GA (Goldberg 2006) is a heuristic approach works on the principle of, “nature evolves species of life and genetic selection of fittest individuals”. In this approach, the initial inputs taken as for solving the problem are presented by a population (called chromosomes) which is again goes through the operators of GA i.e cross over and mutation. The initial population (i.e chromosome) may be taken as string of binary digits or real numbers. Every digit in this population is called a gene. The selection of initial population is chosen randomly or can be created manually. The steps of GA are as follows:

**Step 1.** Initialization of population (i.e initial random inputs will be taken called series of chromosomes)
**Step 2.** do
| Step 2. Evaluation (each initial population will be evaluated through appropriate fitness function) |
| Step 3. Selection (selection of population according to fitness function) |
| Step 4. Apply |
| a) Crossover (population) |
| b) Mutate (population) |
| c) Evaluate (population) |
| While (maximum generation found | maximum coverage of path found) |

**Fig-1:** Steps of Genetic Algorithm (Source: Self)

Genetic Algorithm (GA) comprises of three operators Selection, Crossover and mutation which work on primary population. Selection operator: This is the first step is to evaluate the individuals that are chosen for mating process. This process can be accomplished on their fitness value. The individual that are highly fitted are taken for the process so that high quality product is produced. The Fitness means the ability of an offspring to stay alive in nature and reproduce. Therefore fitness of every chromosome is evaluated. The most fitted chromosomes is selected which will go further for the ensuing generation.
1. **Crossover operator:** This operator is employed on the highly fitted chromosomes. In this operation position of gene is selected and then swapping of genes is done at that point. In this two parents are needed to complete the process. This is repetitive task with different parents until the enough individuals are found for the next generation. There are some types of cross over operation.
   a. **Single point cross over:** In this type, one point is selected on which the recombination of genes to generate two offspring.
   b. **Two points cross over:** In this two points are selected and the genes in that segment are swapped each other to generate offspring.
   c. **Uniform crossover:** The genes from two parents are randomly selected to generate new individual.

2. **Mutation:** Mutation operator is applied on subset of arbitrarily selected population. In crossover over we need two parents and in contrast the mutation operator needs only one parent. Mutation flips chromosomes in such a small way for producing new individual. It is used just to bring the variance in the population.

Types of mutation are as follows:
   a. **Bit-flip:** In this mutation, randomly select one or more bits and flip them.
   b. **Swap mutation:** In this mutation type two positions are selected and change the position of genes.
   c. **Scramble mutation:** In this type, a subset of gene is selected and changes the position of chromosome.
   d. **Inversion mutation:** This is same as that of scramble mutation with a little change, a subset is selected and simply invert the position of genes.
   e. **Random mutation:** In many fields where Genetic Algorithm can be applied for finding optimized and robust solution. It is used in engineering design for evaluating fast and economical process for designing simulation and modeling, Traffic and shipment routing for saving time, Robotics, Data mining, software testing and many more. Software testing means to verify the quality of the software. It is the process to check that the software should meet all development requirements, responds efficaciously to all kinds of inputs, and performs capabilities within an appropriate time. Broadly “testing” is classified in to two types white box testing and black box testing. Here the application of genetic algorithm is explained in white box testing. Genetic algorithm optimizes the solution and finds the feasible path for testing analysis. Since the exhaustive testing is very complex task. Even medium size software has not been thoroughly tested, only some portions of the software have been tested and this portion may or may not be error prone. Therefore genetic algorithm finds the most critical path focuses to test them to increase the efficiency of the software. It is a heuristic approach works with a strong parameter fitness value and leads to the next generation with cross over and mutation operators.

3.1 **Proposed model**

The key idea behind the model is to take initial population as input and apply all the crossover operators and mutation operator in order to find the crossover rate and mutation rate. The best crossover rate is able to use in the GABVIE model for finding the optimal path.

\[ n_{cross\_over} = \text{cor} \times \text{population\_shape[0]} \times \text{population\_shape[1]}/100 \]

After selection operation, the chromosomes of the respective parents whose fitness value is selected from the mating pool are combined together (or crossed over) to form new, enhanced, better progeny. In the literature of Genetic Algorithm, there are many types’ crossover methods and some of them have discussed here. Many of the cross over operators used in the literature survey are domain-specific but in this section we will introduce a few generic crossover operators which is not dependent on the problems. It may be noticed that while for hard search issues, a large number of the crossover operators are not versatile and extensible; they are helpful as a first choice. Therefore there is need to adapt scalable crossover operator. Normally in crossover operator two individual are randomly selected and combined together with some probability known as crossover probability.

\[ \text{r < cross\_over\_rate} \]

If \( r \geq \text{cross\_over\_rate} \) then offspring are the replica of their parents.

**Fig 2. One-point, two-point, and uniform crossover methods.**

**Algorithm 1:** For evaluating the variations of crossover and mutation operator.

**Input:** Initial Population as parameters

**Output:** Generation of highest fitted chromosomes

```python
#cross_over_rate is the rate of mutation.
#n_co=number of crossovers, determined by the formula stated
#n is used if multipoint cross over is chosen. Its default value is 1
# The algorithm carries out mutation n_co times. Each time it finds two chromosomes and carries out requisite cross over.

def crossover(type, cross_over_rate, population, n=1):
    n_co = (cross_over_rate * population.shape[0] * population.shape[1]) / 100
```

4744
The proposed algorithm has been implemented in Python. The IDE used for the implementation is Jupyter. The algorithms scan a binary initial population and perform the tasks as suggested in the algorithm. It may be stated that the algorithm has been divided into two parts. The first part is crossover algorithm that has been taken the population as parameter finds the next generation. The second part is mutation algorithm that find the next generation.

4 OBSERVATION AND RESULTS
The algorithm defined in section 2 has been implemented on the python. Let the natal population be of 15*10

|0 1 0 0 1 1 1 0 0 1| [0 0 1 1 0 0 0 0 1 1] [0 1 0 0 1 1 0 0 1 1]
|1 1 0 1 1 0 1 0 0 0| [1 1 0 1 1 0 0 0 1 1] [0 0 1 1 0 1 1 0 0 1 0]
|0 1 0 0 0 0 1 0 1| [1 0 1 0 1 0 1 0 0 0] [0 1 0 1 0 1 1 1 1]
|0 1 1 1 1 0 0 1| [1 0 0 1 0 0 1 1 0 0] [1 0 1 1 1 1 1 0 0] [0 0 1 0 1 0 1 1 0 0]
|0 1 1 1 0 0 0| [0 1 1 0 0 1 1 0 0] [0 0 1 1 0 1 0 1 0 0] [0 0 1 0 1 0 1 0 1 0 0]

The results of single point crossover are
11 0 3 where 11 and 0 row are selected from the initial population and 3 is the point of mating

|0 1 0 1 0 1 0 0 0 1| [Child 1 i.e. 11th row]
|0 1 1 1 1 1 1 0 0 0| [Child 2 i.e 0th row]

The results of uniform crossover
[0, 0, 0, 0, 1, 0, 1, 0, 1]
The results from uniform crossover
[0, 1, 1, 1, 0, 0, 0, 0, 0]
The results from mutation operators

The following empirical analysis has been carried out to find better solution.
- Variation of the types of cross over
- Variation of the types of mutation

Moreover the technique would be applied to larger programs.

It may be stated that the number of experiments carried out for cross over were 11 as the cross over rates 5, 5.5, 6, 6.5 etc. were taken. Likewise, the number of experiments carried out for varying mutation rate was 7. The mutation rate was varied from 2 to 5 in the gap of 0.5. Theses 18 experiments were carried out for 7 programs hence 126 experiments were carried out to find the optimal solution.

- It was noted that variation of mutation rate has more effect on the technique.
- At mutation arte =3.5 best answers were obtained.
- If the cross over rate was 8.5 best answers were obtained.

REFERENCES