

A Review on Existing Techniques for Generating Automatic Test Case for Object Oriented Software

Dr.Rajender Singh Chhillar , Anju Bala

Department Of Computer Science & Applications
Maharshi Dayanand University, Rohtak

Abstract :

An inevitable part of software testing entails the generation of test cases. A good test case should have the quality to cover every aspect of test objective. An effective and efficient test case generation is the most challenging and time consuming task in software testing. A good test case characteristics to cover more given set of path coverage with reducing time and cost of software development. Researcher have proposed different techniques to generate test case automatically. However , those techniques also have some drawbacks. To overcome these drawbacks, we introduce a technique (i.e. GA) to generate small numbers of efficient test cases with expectations to cover more given set of target. In this paper we introduce that Genetic Algorithm is quite useful search method or technique to generate large volume of test cases very effectively and efficiently with multiple domain.

Keywords: Automatic test case generation, classification with pros and cons, Genetic Algorithm, MATLAB.

INTRODUCTION

Software testing is an inevitable part of software development life cycle. It is the most tiring and critical phase in SDLC. Software testing is the process of executing a program with the intent of finding an error. Testing is a very essential, though very labour-intensive and expensive process in software development and handling as well as maintenance of software. It accounts that it takes 50 percent of total cost of entire development [1]. If the testing process could be automated, the cost of developing software could be reduced significantly. A challenging part of this phase entails the generation of test cases. This generation is very crucial to the success of the testing because it is impossible to obtain a fully tested program given that the number of test cases will be able to find maximum numbers of fault in minimum number of iteration. Software program is infinite, and a suitable design of test cases will be able to detect a great number of faults. For the requirement of good and consistent product, the techniques for automatic generation of test cases make an efforts efficiently to systematically uncover the different type of errors with a minimum amount of time and efforts. A test case is a set of tests performed in a sequence and related to a test objective, which will produce a number of tests input values, observed output, expected output, and any other information needed for the test to run, such as environment prerequisites [2].

A good test case should have the quality to cover every aspect of test objective and high rate of fault detection. The method used for automatic generation test case usually based on efficiently searching or finding small set of test cases with high probability of detecting as-yet undiscovered error [1]. There have been few efforts on representing a brief classification, which covers all existing automatic test case generation approaches. In this paper we introduce a general classification for automatic test case generation approaches with comparison between these approaches to show that Genetic Algorithm is the most acceptable approach to generate automatic test case.

1.OBJECTIVE

Our main objective is to generate good test cases with the intent of finding an error as far as possible. There has been a significant amount of work in generation of test cases that makes an effort to increase the amount of observed behaviour. Instead of these wide researches, there have been few efforts on representing a brief classification, which covers all existing automatic test case generation approaches. A general classification for these techniques is mentioned in [3, 4, 5]. In [6] a classification framework for automatic test case generation methods is mentioned which is based on software development phase in which testing is applied. In [7] a classification of search-based automatic test case generation approaches is represented. In this study, we introduce classification and evaluation of automatic test case generation techniques, which tries to cover all existing approaches with showing the requirement of Genetic Algorithm.

2.NEED OF AUTOMATIC TEST CASE GENERATION

The most critical part of testing is the generation of test cases. Effectively speaking there are numerous reasons to automate test case generation task in software testing. Some of the most logical reasons are as follows. Reducing the testing cost: most important reason is to minimize the testing cost. During testing phase the cost can hike more than the expected value due to improper test cases. These improper test cases cause wastage of organizational resources as well as time. So minimizing the cost for getting the adequate product becomes needy[8]. Reducing human malfunctioning: second important reason is to minimizing the human malfunctioning. There is no benchmark to test the test case is valid or not. It basically depends on the testers understanding of the requirement .So there becomes more chance of inclusion of bugs by tester in system after testing. To overcome this problem, automatic test case generation phase becomes very necessary [9]. Increasing software products efficiency: It is very crucial part to overcome the problem is occurred by manual testing to increase the software product efficiency. Therefore, automatic test case design has become very crucial part to assure the quality of today,^s large software products [10]. Reducing number of test cases: Generate efficient test cases are the essential part for convenience the test work and improving the test efficiency. The test work is ineffective because of the great number of the manual test cases, so some Automation algorithms are needed to optimize the test cases [11].

To cover all system requirements: Generation of automating test case process provides a means to ensure that test cases have been obtained in a adequate and objective manner and that all system requirements have been covered [12]. The application of automatic test case generation are as wide as software application. In short, automatic test case increases the integrity of software to make a sound product.

To generate automatic test cases using Genetic Algorithm is very powerful and novel approach.

3.GENETIC ALGORITHM

INTRODUCTION

GA based on natural phenomena. The main concept of GAs is “natural selection” and “genetic inheritance”.the father of GA is **John Holland, university of Michigan (1970) in U.S.A.** It is based on the evolutionary principle. GA belongs to the class of probabilistic algorithm. GA provide directed search algorithms based on the mechanics of biological evolution. The best reason behind to use GA than other searching algorithm is that GA performs a multi-directional search by maintaining the optimized solutions not a single point. Main application areas of GA is AI ,business, scientific and engineering circle. But these can be used in many forms like to generate automatic test cases etc. GA class of probabilistic optimization algorithm. Genetic algorithms are categorized as global search heuristic or GA is good heuristic search for combinatorial problems[20]. Ex.*TSP, pen movement of a plotter, real world routing of school, prisoner,^s dilemma.*

4.CONCEPT

The Genetic Algorithm is an example of a heuristic search, which gives the optimized result and works on randomly generated population. How Genetic algorithm works are highlights under the following.

Highlights on each section.

1. How Genetic Algorithm starts

- a. Generate the initial population randomly i.e. either generated by manually or through function.
- b. Calculate the fitness and select the chromosomes with the best fitness value..
- c. Select two parent chromosomes from current population.
 - with the use of mutation and crossover operator, recombine the selected chromosomes.
- c. Replace the current population by inserting the new offspring . check the fitness value.
- e. if a stop criteria is met , then return the chromosomes with the best fitness. Otherwise go to step b.

2. Random generation of population

Genetic Algorithm always start with a randomly generated population for any problem.

2. Generate new generation:

Randomly generated population is called current population. GA helps to select the individual from current population called 'parents'. Check fitness value, if it met then stop. Otherwise apply genetic operator i.e. mutation and crossover. And create new population and check the fitness value till fitness value met. Three important striking feature of GA:

- a. *Elite selection*: Sometimes individual in current population with best fitness value is present already. GA copy these chromosomes as it is in next population.. it is the significant feature of GA.
- b. *Crossover* : Gas have many type of crossover. With help of crossover , we creates a better offspring.
- c. *Mutation* : it is also called simple search operator. Mutation probability set low as compared to crossover. Suppose if crossover is fail to generate best offspring then mutation operator apply. Mutation flipping the 0's and 1's and get the new offspring. Mutation mainly performed to pruning the dead state.

4.Stopping criteria of GA flow:

The Genetic algorithm uses the following five conditions to determine when to stop.

- Generation reach to maxima
 - When specified time becomes elapsed.
 - Fitness value shows same result.
 - No showing the improvement in objective function i.e. stall generation
 - No showing the improvement in the objective function in given interval time i.e. stall time limit.
- These are the basic stopping criteria[19].

5. SELECTION METHOD

A Genetic Algorithm (GA) has several genetic operators that can be modified to improve the performance of particular implementations. These operators include parent selection, crossover and mutation. Selection is one of the important operations in the GA process[18]. There are several ways for selection.

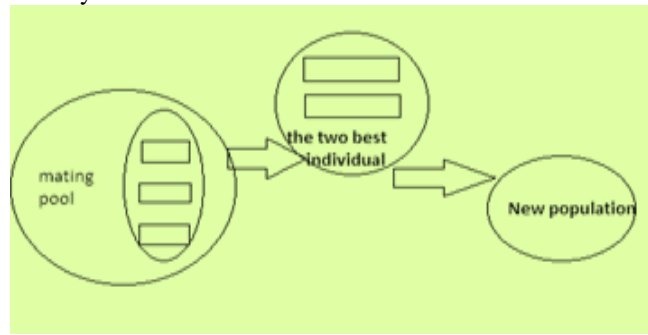


Fig.1 selection

There are different types of selection strategies :

- A) *Tournament Selection*
- B) *Proportional Roulette Wheel Selection*
- C) *Rank-Based Roulette Wheel Selection*
- D) *Boltzmann Selection*
- E) *Elitism*
- F) *Stochastic Universal Sampling*

Out of these selection strategies or methods, we can use any one or combination of these according to our requirement[20].

6. SEARCH BASED ALGORITHM FOR AUTOMATIC TEST CASE GENERATION

Here figure 1 shows a variety of mostly used searching algorithm(methods) to cover the purpose of automatic test case generation.

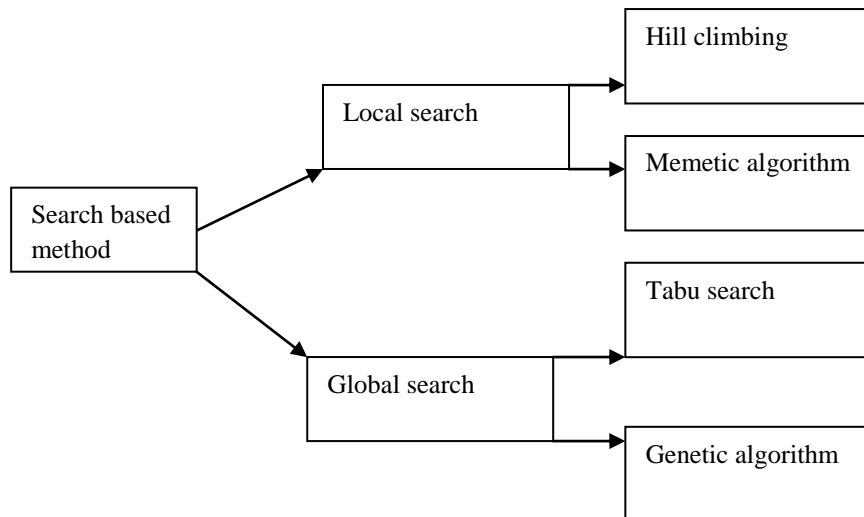


Fig.2. search methods

1. Generation of Test Case By Hill Climbing Technique

In the field of computer science, hill climbing is a mathematical optimization algorithm uses a simple heuristic function. It is the part of local search. It is the expanded algorithm of depth-First-Search. it select the point in neighbourhood , so it is also called neighbourhood search. It follows the iterative approach. The procedure of Hill Climbing technique:

- It starts with a single point in the search space.

- In first iteration, it choose a randomly new single point from neighbourhood N.
- If the selected point gives the best fitness value according to objective function, it becomes the current point, otherwise go to step 2.

If there are no better solutions in N, a local optimum has been reached. In such a case, a restart of the algorithm from a new random solution can be done. Sometimes this method is known as iterated hill climbing. In general HC is a simple algorithm but may lead to a local optimum.

Disadvantages: one of the main problem with hill climbing is getting stuck at local maxima. When you stuck at local minima you have no other choice ,so have to select random node again and again till time limit. When time limit expired ,you have to select least local minima. Basically three main drawbacks of HC is Local maximum, Plateau and Ridge.

2. By Memetic Algorithm

The Memetic Algorithms (MAs) are meta heuristics that use both global and local search (e.g., a GA with the feature of HC). Meta heuristic means that combine the structure and strategy guidelines for developing a specific heuristic method to fit a particular a kind of problem. It is based on by the cultural evolution. A meme means imitation in cultural transmission. From an optimization point of view, a MA also population based like GA, but during generation of offspring local search is applied rather than global search. After time limit it also reach to local maxima. The MA also used [4] for testing of object-oriented software is quite simple. It is built on GA, and the only difference is that at each generation on each individual a Hill Climbing is applied until a local optimum is achieved. So the number of generation is less than Genetic Algorithm.

*Disadvantages:*A major difference between the algorithm is that the time it takes to complete iteration in Memetic Algorithm is much more than the time it takes genetic algorithm to complete the same iteration. This is because of the local search algorithm that is embedded in Memetic Algorithm.

3. By Tabu search

Tabu means taboo i.e restricted. Tabu Search (TS) is also metaheuristic search technique based on the premise that Taboo search is more intelligent technique adaptive memory and responsive exploration must be incorporated with problem solving. Tabu Search algorithm also based on that of the next k neighbours, but in this Tabu list have to be maintained of visiting neighbours. The role of Tabu list(memory) can change according to progress of algorithm. Most striking feature of Tabu Algorithm ,it provide a number of parameter ,to select parameter according to problem to be solved.

Outline of Tabu Search Algorithm:

- Select an individual from next k- neighbour. While maintaining visiting elements, is called candidate list strategy.
- Check the fitness value based on objective function.
- If the objective function me, , then stop . otherwise go to step 2.
- If no optimal solution is found in time limit and reach a local optimum then select the solution from candidate list without exploiting neighbour element.

Another striking feature is memory has to be maintained i.e. short term memory and long term memory to store recent moves and frequency of these moves.

The main goal behind Tabu search is to generate of test case for structural testing and covering maximum branch of coverage for PUT.t.

*Disadvantage :*In Tabu search algorithm, memory have to be maintained. Pure Tabu search that uses only one solution can easily miss some promising areas of the search space in this algorithm larger set of parallel solutions does not ex-change information (Zdansky AND Pozivil, 2002) .

4. By Genetic Algorithm

GA based on natural phenomena. The main concept of GAs is “natural selection” and “genetic inheritance”.the father of GA is **John Holland, university of Michigan (1970) in U.S.A.** It is based on the “evolutionary principle, survival of fittest”. GA belongs to the class of probabilistic algorithm[4]. GA provide directed search algorithms based on the mechanics of biological evolution. The best reason behind to use GA than other searching algorithm is that GA performs a multi-directional search by maintaining the optimized solutions not a single point.

GA can be used for [13] generating test cases for structural testing as well as behavioural testing. With the one run in GA ,cover multiple paths. It establish relation between old and new test caeses.

Brief explanation on what elements are required to generate test cases using GA:

- Genetic representation for a given problem i.e. generation of test case[14].
- Adopt method to create initial population.
- Evaluate fitness function to objective function.
- Genetic operator to recombine the individuals.
- Assign value of various parameter used by genetic algorithm.

It takes less time than other search algorithm.

Overall the best & best, genetic algorithm is the most preferred algorithm used in test case generation task till now, as a result of leading to acceptable test suits[11].

GA is very helpful to sort the optimization problem. It gives the clear view in dilemma. GA also helpful in many useful areas: TSP, Prisoner's dilemma and quadratic equation. In this type of problems, even in a "single point crossover" the mutation is expected to happen 200-2000 generation. It is easy to change the mutation as it is implemented as a separate function.

From the above discussion, it can be seen that the GA differs from traditional technique and Genetic Algorithms. The main noticeable differences are:

- GA works in parallel in population rather than a single point.
- GA run on coded version of the problem parameter rather than parameter or variables.
- GA use probability than deterministic rule
- GA does not use derivatives functions, it uses fitness functions.

The important characteristics of GA is, it gives the no. of solutions and generation, user can select solution based on his objective. GA also works in combinatorial problem. In cases where a particular problem does not have one individual solution, for example TSP. Then the GA is potentially useful for identifying these alternative solutions simultaneously.

7. MATLAB

MATLAB stands for (**Matrix Laboratory**), a product of Mathworks, it is a scientific software package developed to provide an integrated environment for numeric computation and graphics visualization in high-level programming language. Originally it was written by Dr Cleve Moler, Chief scientist at MathWorks, Inc., to provide easy access to matrix software developed in the LINPACK and EISPACK projects [19].

MATLAB 's basic data elements are matrix.

MATLAB has a wide collection of functions useful to the genetic algorithm practitioner and those wishing to experiment with the genetic algorithm for the first time.

In MATLAB's high-level language, problems can be coded in **m-files** in a fraction of the time that it would take to create C or FORTRAN programs for the same purpose. It also provide advanced data analysis, visualization tools and special purpose application domain toolboxes. Now a day's MATLAB is often employed in research and industry and it is an example of a high-level "**scripting**" or "**4th generation**" language[18].

The main data structures in the Genetic Algorithm toolbox are:

- Fitness function
- No. of variables
- population size
- selection
- mutation
- crossover

8.FUTURE WORK

We will generate automatic test cases using genetic algorithm based on MatLab gatool or without MatLab. MatLab provide a Genetic Algorithm tool that is a graphical user interface that enables us to use the genetic algorithm without working at the command line.

Our proposed algorithm follow the following steps(without MatLab):

- Generate the state flow diagram for given program.
- analyze and collect of possible paths between initial and goal state.
- repeat steps 4 & 5 till most distinguish path not occurred.
- Select two possible path $p_i \{p_{i1}, p_{i2}, p_{i3}, \dots, p_{in}\}$ and $p_j \{p_{j1}, p_{j2}, p_{j3}, \dots, p_{jn}\}$
- Perform crossover on p_i & p_j such that

-create a new sequence $p_k = \{p_{k1}, p_{k2}, p_{k3}, \dots, p_{kn}\}$

-the sequence will be selected on the basis of the frequency of a state at index position i .

- Perform mutation on pk to pruning the dead state.
- Exit

With MATLAB gatool:

You have to must enter the following detail according to problem:

- Fitness function
- No. of variable
- Population
- Fitness scaling
- Selection
- Mutation
- Crossover
- Stopping criteria.

9.CONCLUSION

The main objective in this paper is to illustrate that how the new technology of MATLAB can be used in order to implement a genetic algorithm in optimization problems. It uses the power of genetic algorithms to generate fast and efficient solutions in real time. MATLAB is very new approach to provide very interactive or GUI based toolboxes out of these toolboxes, one most important toolbox is gatool. We will use this toolbox to generate automatic test cases. we also gone through a very brief idea on generating automatic test cases techniques with including Genetic Algorithm advantages which is a very new approach for problems related to optimization.

REFERENCES

- [1] R. Blanco, J.Tuya and B. Adenso-Díaz, “Automated test data generation using scatter-search approach”, *Information and Software technology*, vol. 51, Issue 4, (2009), pp. 708-720.
- [2] B. N. Biswal, S. S. Barpanda and D. P. Mohapatra, *International Journal of Computer Applications*, vol. 1, Issue 14, (2010).
- [3] R. Jeevarathinam and A. S. Thanamani, “Towards Test Cases Generation from Software Specifications”, *International Journal of Engineering Science and Technology*, vol. 2, Issue 11, (2010), pp. 6578-6584.
- [4] A. Arcuri and X. Yao, “Search based software testing of object-oriented containers”, *Information Sciences*, vol. 178, no. 15, (2008) August, pp. 3075-3095. [5] E. Alba and F. Chicano, “Observations in using Parallel and Sequential Evolutionary Algorithms for Automatic Software Testing”, *Computers & Operations Research*, vol. 35, no. 10, (2008) October, pp. 3161–3183.
- [6] M. Prasanna, S. N. Sivanandam, R. Venkatesan, R. Sundarajan, “A Survey on Automatic Test Case generation”, *Academic Open Internet Journal*, vol. 15, (2005). [7] P. McMinn, “Search-based software test data generation: A survey”, *Software Testing, Verification & Reliability*, vol. 14, no. 2, (2004) June, pp. 105–156.
- [8] A. Sharma, A. Jadhav, P. R. Srivastava and R. Goyal, “Test cost optimization using tabu search”, *J. Soft. Eng. Appl.*, vol. 3, no. 5, (2010), pp. 477–486.
- [9] V. Rajappa, A. Biradar, S. Panda, “Efficient software test case generation using genetic algorithm based graph theory”, *Proceedings of the First International Conference on Emerging Trends in Engineering and Technology*, (2008), pp. 298-303. *International Journal of Software Engineering and Its Applications* Vol. 6, No. 4, October, 2012
- [10] S. K. Swain, D. P. Mohapatra and R. Mall, “Test case generation based on state and activity models”, *Journal of Object Technology*, vol. 9, no. 5, (2010), pp. 1 – 27.
- [11] Q. Li and J. Li, *Proceedings of the International Symposium on Intelligent Information Systems and Applications*, (2009).
- [12] S. J. Cuning and J. W. Rozenblit, “Test scenario generation from a structured requirements specification”, *journal of Intelligent and Robotic Systems*, vol. 41, no. 2-3, (2005), pp. 87-112.
- [14] G. Dunwei, Z. Wanqiu and Z. Yan, *Chinese Journal of Electronics*, vol. 19, no. 2, (2011).
- [13] B. N. Biswal, S. S. Barpanda and D. P. Mohapatra, *International Journal of Computer Applications*, vol. 1, Issue 14, (2010).

- [14] S. Wappler and J. Wegener, “Evolutionary testing of object-oriented software using a hybrid evolutionary algorithm”, IEEE Congress on Evolutionary Computation, (2006).
- [15] E. Díaz, J. Tuya, R. Blanco and J. J. Dolado, “A Tabu Search Algorithm for Structural Software Testing”, Journal Computers and Operations Research, ACM, vol. 35, no. 10, (2008), pp. 3052–3072.
- [16] ‘Tutorial on Genetic Algorithm’- Dr. Adel Abdenmour(Electrical Engineering Department).
- [17] ‘Teaching Genetic Algorithm Using Matlab’- Y.Z.CAO and Q.H.WU
- [18] “principal of soft computing” second edition- Dr. S.N. Sivanandam (department of computer science and engineering)
- [19] International journal of latest trends in engineering and technology(IJLTET) vol.3 issue 3,2014.
- [20] T. Blicke, L. Thiele, A Comparison of Selection Schemes used in Genetic Algorithms. TIK-Report, Zurich, 1995