Review of Automatic Test Case Generation from UML Diagram using Evolutionary Algorithm

Kirandeep Kaur, Vinay Chopra

Abstract- Software testing plays a vital role in software development life cycle. An approach of testing which takes place at design phase can remove errors in the system and improvise the developed project. Automatic test case generation can be used for testing software or real time applications. Many evolutionary algorithms are used for generating test case automatically. This paper represent review of approach of automatic test case generation by analyzing the dynamic behaviour of UML diagram which takes place at design phase of SWDLC by using evolutionary algorithm multi objective genetic algorithm. Single objective genetic algorithm has been already used for automatic testing.

Keywords: UML diagram, Test cases, MOGA, DFS, Tree structure.

I. INTRODUCTION

1.1 Software Testing

Software testing is an important task in the software development life cycle. While developing the software, near about half of budget is spend on the testing related task by the organizations. Software testing can be stated as an investigation conducted to provide stakeholders with information the quality of the product or service under the test. Software testing can also defined as process of validation and verification of the software to satisfy the needs of stakeholders and works as expected [4].

1.2 Automated Testing

We can do testing either manually or automatically. In manual testing, testing is perform by human by testing the code manually and then compares the result with expected behavior and records the observations [6]. Manual testing does not give the better result like effectiveness and efficiency. The best way is to test the software automatically to increase the effectiveness and efficiency and also coverage of software testing. Automated testing use automated tool to test the software which require no investment of human. It only requires the investment of money and resources. As compare to manually testing, automation testing gives accurate and faster result. The most common way to test the software is to generate the test cases and then test automatically by some automated tool. After that result is compare with best cases and then find optimality.

1.3 Unified Modelling Language(UML)

The UML language is used to specify, visualize, modify, constructs and document artifacts of an object oriented software system under the development.

Manuscript Received on October 2014.

Kirandeep Kaur, Department of Computer Science and Technology, Daviet, Jalandhar, India.

Dr. Vinay Chopra, Asst. Prof., Department of Computer Science and Technology, Daviet, Jalandhar, India.

The techniques which are available in UML are data, business, object and component modeling.
These two sub categories are

- Structural Diagram:-

It represents the structure of the software.

- Behaviour Diagram:-

It shows what must happen in system being modelled.

1.4 Multi Objective Genetic Algorithm

Multi-objective optimization refers to the solution of problems with two or more objectives to be satisfied simultaneously. Often, such objectives are in conflict with each other, and are expressed in different units[12].Because of their nature, multi-objective optimization problems normally have not one but a set of solutions, which are called Pareto-optimal solutions or non-dominated solutions. When such solutions are represented in the objective function space, the graph produced is called the Pareto-front or the Pareto-optimal set. A general formulation of a multi-objective optimization problem consists of a number of objectives with a number of inequality, and equality constraints.

A. Single Objective Function Optimization

This optimization is based on optimizing only single objective at a time hence it is known as single objective function optimization [10].

B. Multi-Objective Function Optimization

This optimization is based on optimizing two or more objectives at a time hence it is known as multi-objective function optimization. [10].In case of multi-objective function optimization, each function yields its own decision space. This means that the number of decision spaces will be same as the number of functions that need to be optimized [18].

II. LITERATURE SURVEY

M.Prasanna et.al (2009) represents the new model based approach for automated generation of test cases in the object oriented systems. But there research is limited to the object diagrams taken from the UML model of the system. Genetic algorithms tree crossover technique has been introduced to generate all possible test cases of a given object diagram. The experimental result show that it has the capability to reveal 80% faults in unit level and 88% fault in the integration level. The effectiveness of test case can be checked using a fault injection technique called mutation analysis [17]. Aysh Alhoobet et.al (2010) introduces a new approach which extent the integrated classification tree methodology (ICTM).

In this, software allows the tester to construct test cases from functional specification. UML class diagram is used in their research work. Language which is used by them is object constraint language (OCL) to represent the software specification. The results show that ICTM is improved to detect specification automatically. In this restructure algorithm is also purposed to remove duplication of sub trees on different levels [9]. Ashalatha Nayak et.al(2010) discussed the proposed approach of synthesizing test data from information embedded in model elements. Elements are class diagrams, sequence diagrams and OCL constraints. Sequence diagram is enriching with attributes and from class diagram and OCL constraints information is derived and map it onto a structured composite graph called SCG. From SCG, then test specification are generated. By detecting the infeasible scenario, a significant effort which is wasted in trying to satisfy inconsistent constraints can be saved [3]. Santosh Kumar swain et.al (2010) developed an integrated approach to generate test cases from the UML activity and sequence diagram. First of all these diagrams are converted into a graph. After that purposed algorithm is used to generate the test scenario from the constructed graph. Information like as method activity sequence, associated object and constraint conditions for test case generation are exacted from the test scenario. This approach is also work on executable forms of UML as well as code implementation design. But to reduce unduly increasing number of test cases, test cases generating are to be optimized[26]. Santosh Kumar swain et.al (2010) discussed a novel testing techniques for object oriented program. To construct intermediate representation, UML state and activity models are used which named as state activity diagram (SAD). Test cases are generated to achieve the state activity coverage of SADs. Here effectiveness of approach is also evaluated. For UML test path and conditional generation UTCG algorithm is purposed[27]. Hitesh Tahbildar et.al (2011) gives the overview of the automatic test data generation. The different implementation techniques are present with its merits and demerits. Different architecture is taken based on the test data generation implementation techniques. Architecture based on symbolic value execution, architecture based on the concrete/actual value execution, architecture based on both symbolic and actual value execution and architecture based on an object oriented approach are presented. In the research area of automated test data generation, these concepts are mandatory [14]. Pakinam N. Boghdaly et.al (2011) examine the different approaches that had come from the past decade related to generation of test cases and test data from the various different models as an emerging type of models based testing. UML also take part in these models. Mostly researcher pays attention at optimization and reducing the test cases rather than generating the test cases. It also discussed the different model based technique which is automatically generating the test cases from the UML diagrams[20]. Huang Wei et. al.(2011) discussed application of Pareto multi-objective genetic algorithm, which has raised in recent years, in transmission network expansion planning is proposed so as to reconcile the conflicting goals in transmission network expansion planning. With the advantage of this algorithm the "N-1" security criteria is considered as one of the objectives in the form of the penalty function to meet with the high reliability requirements of transmission network[15]. Ranjit Swain et.al (2012) introduces the approach to developing and generating test cases from the UML state machine diagram. In approach firstly the state machine diagram is traversed and then the conditional predicates are selected. To generate test cases automatically by applying function minimization technique, these conditional predicates are transformed . In their research it is also suggested that other diagrams can also used to generate test cases[24]. A.V.K Shanthi et.al (2012) purpose a novel based approach to test the software at the initial state itself so that, for the software testers, it will be easy to test the software at later stage. In their study, test cases are generated from the UML sequence diagram using tabu search algorithm. Tabu search algorithm is an iterative method which has higher level of heuristic procedure design for solving optimization problems. This approach reveal all paths for software to be developed and also obtained test cases valid once, the experimental results shows that this method gives better performance [8]. Ranjita swain vikas et.al (2012) presents the methodology of testing to test object oriented software based on UML state chart diagram. In their methodology, they apply the function minimization technique and then generate the test cases automatically from the UML state chart diagram. Here AGeTeSc algorithm is presented to generate the test cases for soft drink vendor machine. In this number of test cases are reduce and achieve the transition coverage by testing the boundaries determined by simple predicates [23]. A.V.K.Shanthi et.al (2012) introduces a heuristic technique to test the software at the initial state itself so that to test the software in the later stages, it will be easy for software testers to test the software. In their research, they purpose a new approach for generating the test cases from the UML activity diagram using genetic algorithm. It also inspires the developers to improve the design quality, find faults and reduce the time. It may be possible to develop tool for this approach which can be help in reducing cost of software development and improve quality of software [4]. Swati Tahiliani et.al (2012) explores the different algorithm and techniques which are applied on use cases and on other UML diagrams. To derive the test cases, use cases are used in different ways. Sequence diagram is used to generate test cases by different techniques like OCL, functional minimization technique and functional predicts. There are so many approaches used to generate the test cases. Each approach has its own advantages and disadvantages [25]. A.V.H.Shanthi et.al (2012) introduces the approach which is mainly focus on developing effective technique and tool for test case generation. In this approach to generate test cases for object oriented program, data mining technique is used. First of all java programs is tested according to its class diagram mode and then generate test cases using tabu search algorithm. This approach help in generating test cases after the comparison of design phase and error could be detected at early stages in the software development life cycle [5].

Puneet Patel et.al (2012) introduces approach for generating the test cases from the activity diagram. Here coverage criterion is also considering called activity path coverage criterion. In this first of all from activity diagram activity graph I generated and then from this graph test cases are generated i.e. activity path coverage criteria is followed. The result shows that this approach is help in finding the faults like synchronization faults and loop faults [19]. Rakesh Kumar et. al.(2013) discussed Software testing is most effort consuming phase in software development. Generating test cases automatically will reduce cost and efforts significantly. In this paper, test case data is generated automatically using Genetic Algorithms and results are compared with Random Testing. It is observed that Genetic Algorithms outperforms Random Testing [22]. Vikas Panthi et. al (2013) Software testing plays an important role in software development because it can minimize the development cost. In this paper main features extract from sequence diagram after that they can write java source code for that features according to Model Junit library. Model Junit is an extended library of Junit Library. This paper describes a systematic test case generation technique performed on Model Based Testing (MBT) approaches by using sequence diagram [31]. A. Balieiro et. al. (2013) implemented An adaptive sensing period optimisation scheme for cognitive radio networks based on a multi-objective genetic algorithm formulation is proposed. The proposed scheme aims at maximising the spectrum opportunities as well as minimising the incurred sensing overhead. The simulation results show that the proposed scheme outperforms the non-optimised proposals by up to 90% [1].

III. PROPOSED WORK

In earlier research genetic algorithm has been used to generate the test cases from sequence diagram and the tree is built, crossover technique has been applied. In this papers test case generation takes place at design phase by analyzing the dynamic behavior of the UML diagram. Then multi objective genetic algorithm will be used to generate test cases from UML diagram and corresponding tree structure will be generated. By applying crossover technique new generation of tree will takes place in the form of binary tree. Depth first search will be used for traversing of new generated trees. In the end mutation coverage will cover all test cases sequence to check the validity.

IV. OBJECTIVES

The main objective of this study is to:

I. To design and implement sequence diagrams for real time application using Agro UML and represents the attributes and operations of objects involved in corresponding tree structure.

II. To design and implement Multi-objective Genetic algorithm for automatic test case generation for the above designed tree structures.

III. To compare the effectiveness and performance of designed Multi-objective Genetic algorithm with Genetic

Algorithm on the basis of following parameters:-

1. Fitness function
2. Time and space
3. Fault coverage

V. RESEARCH METHODOLOGY

In designed methodology, sequence diagram is drawn using AgroUML tool. It gives the detailed state of system. Secondly, mapping is done in which diagram is mapped into tree with root node and child nodes. Then thirdly, Multi Objective genetic crossover algorithm is applied which form new generation of trees which then further convert into binary trees. Depth first search technique is applied on these binary trees to generate valid test cases set.

The steps of methodology are given below:-

1. Construct the sequence diagram using AgroUML and save the UML diagram.
2. Then parse this file to capture the object names for the tree formation.
3. Then tree formation is will be constructed using object names and applies Multi Objective genetic crossover technique.
4. Convert the new generation trees into binary trees.
5. Using depth first search technique, traverse new generation of binary trees.
6. Valid test cases set is obtained from step 5.
7. Apply the mutation coverage to check the effectiveness of test cases which gives the efficiency level of approach.

Efficiency level of approach is calculated by the score formula which is calculated by the number of faults injected and faults found.

Flowchart of Research Methodology:

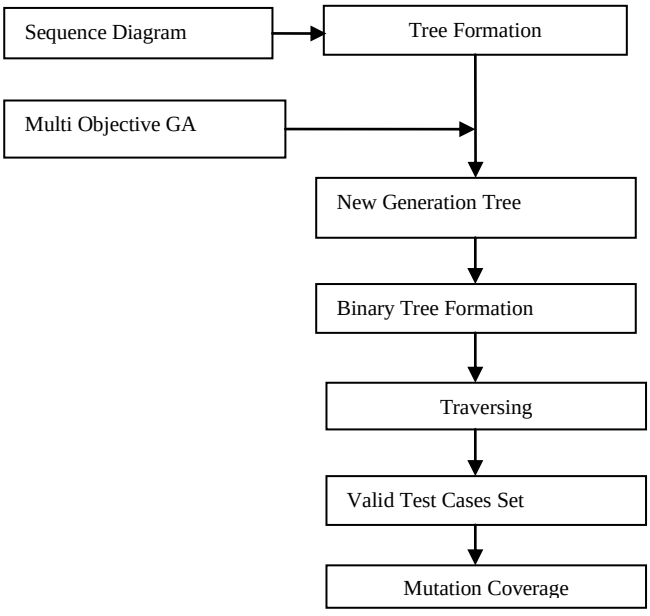

Fig No. 1:- Flowchart of Research Methodology

VI. CONCLUSION AND FUTURE WORK

Evolutionary algorithms have been used for automatic test case generation. Genetic algorithm is most widely used techniques for automatic testing.

Due to limitation of Single objective genetic algorithm that only one objective can be considered for evaluation of test cases, new technique Multi objective Genetic Algorithm will be used for test case generation from UML sequence diagram. In future evolutionary algorithm MOGA can be used for automatic test case generation at design phase from other UML diagrams like class diagram, object diagram, activity diagram etc.

REFERENCES

1. A. Balieiro, P. Yoshioka, K. Dias, C. Cordeiro and D. Cavalcanti," Adaptive spectrum sensing for cognitive radio based on multi-objective genetic optimisation", ELECTRONICS LETTERS ,Vol. 49 No. 17, 2013.
2. Alessandra Cavarra, Thierry Jeron, Alan Hartman "Using UML for Automatic Test Generation", ISSTA,2002.
3. Ashalatha Nayak and Debasis Samanta "Automatic Test Data Synthesis Using UML Sequence Diagram", Journal of Object Technology, vol. 09, no. 2, March-April, pp. 75-104, 2010.
4. A.V.K Shanthi and G. Mohan Kumar "A Heuristic Technique for Automated Test Cases Generation for UML Activity Diagram", JCSA, vol. 4, no. 2, pp. 75-86, 2012.
5. A.V.K Shanthi and Dr. G. Mohan Kumar, "A Novel Approach for Generating Test Cases Using Tabu Search Algorithm", UNIASCIT, vol. 2, no. 2, pp. 222-224, 2012.
6. A.V.K.Shanthi and G. Mohan. Kumar, " Automated Test Cases Generation from UML Sequence Diagram", IJSCA, vol. 41, Singapore, pp. 83-89, 2012.
7. A.V.K.Shanthi and Dr. G. Mohan Kumar "Automated Test Cases Generation For Object Oriented Software", Indian Journal of Computer Science and Engineering, vol. 2, no. 4, Aug-Sep, pp. 543-546, 2011.
8. A.V.K.Shanthi and G. Mohan Kumar, "A Heuristic Approach for Automated Test Cases Generation for Sequence Diagram Using Tabu Search Algorithm", European Journal of Scientific Research, vol. 85, no. 54, September, pp. 534-540, 2012.
9. Aysh Alhroob, Keshav Dahal, Alamgir Hossain," Automatic Test Cases Generation from Software Specification", ISE Journal, vol. 4, issue 1, pp. 109-121, 2010.
10. Deb, K. "Multi-Objective Optimization Using Evolutionary Algorithms". Reading, John Wiley & Sons, Ltd, Reprinted April 2002.
11. Fonseca C M, Fleming P J, "Multi-objective optimization and multiple constrains handing with evolutionary algorithms-Part I: a unified formulation", IEEE Transactions on systems, Man and cybernetics, vol. 28, pp. 26-37, 1998.
12. Gu Lei, Yu-geng Xi, "A Parellel Multi-Objective Genetic Local Search Algorithm ", Control Engineering of China, Shenyang, vol. 16, pp. 738- 742, Nov. 2009
13. Heidi A. Taboada, Jose F. Espiritu, and David W. Coit," MOMS-GA: A Multi-Objective Multi-State Genetic Algorithm for System Reliability Optimization Design Problems", IEEE TRANSACTIONS ON RELIABILITY, VOL. 57, NO. 1, MARCH 2008.
14. Hitesh Tahbildar and Bichitra Kalita , "Automated Software Test Data Generation: Direction of Research", IJCSE Survey, vol. 2, no. 1, Feb, PP. 99- 120, 2011.
15. Huang Wei, Fengli, He Zijun, Cui Junzhao, Zhang Li," Transmission Network Planning With N-1 Security Criterion Based On Improved Multi-objective Genetic Algorithm", IEEE(978-1-4577-0365-2) , pp-1250-1254,2011.
16. L.C.Braind, Y.Labiche, She, "Automating Regression Test Selection Based on UML designs", Information and Software Technology, vol. 51, issue 1, January, pp. 16-30, 2009.
17. M. Prassana and K.R. Chandran ,"Automatic Test Cases Generation for UML Object Diagrams Using Genetic Algorithm", Int.J.Advance.Soft Comput. Appl., vol. 1. No. 1, July, pp. 19-32, 2009.
18. Manoj Kumar, Dr. Mohammad, Gyanendra K.Gupta, Amarjeet Singh ,"An Efficient Algorithm for Evaluation of Object Oriented Models", IJCA, vol. 24, no. 8, June, pp. 11-15, 2011.
19. Mr. Kashif Waqas1 Dr. Rauf Baig1and Dr. Shahid Ali2," Feature Subset Selection Using Multi-Objective Genetic Algorithms", IEEE, ISSN- 978-1-4244-4873-9,2009.
20. Pakinam N.Boghdady, Nagwa L.Badr, Mohmed Hashen and Mohnamed F.Tolba,"Test Case Generation and Test Extraction Techniques", IJECS, vol. 11, no. 3, June, pp. 87-94, 2011.
21. Puneet Patel and Nitin N. Patil ,"Test Case Formation Using UML Activity Diagram", World Journal of Science and Technology, vol. 2, no. 3, April, pp. 57-62, 2012.
22. Rakesh Kumar, Surjeet Singh, Girdhar Gopal," Automatic Test Case Generation Using Genetic Algorithm",IJSCR, Volume 4, Issue 6, June, pp-1135-1141, 2013.
23. Ranjit Swain, Vikas Panthi, Prafulla Kumar Behera and Durga Prasad Mohapatra , " Automatic Test Case Generation from UML Sate Chart Diagram", IJCA, vol. 42, no. 7, March, pp. 26-36, 2012.
24. Ranjit Swain, Vikas Panthi, Prafulla Kumar Behera and Durga Prasad Mohapatra,"Test Case Generation Based on State Machine Diagram", IJCIS, vol. 4, no. 2, Feb, pp. 99-110, 2012.
25. Swati, Thilian, Pallavi Pandit, "A Survey of UML Based Approaches to Testing", International Journal of Computational Engineering Research, vol. 2, issue 5, September, pp. 1396-1401, 2012.
26. Santosh Kumar Swain, Durga Prasad Mohapatra and Rajib Mall ,"Test Case Generation Based on State and Activity Diagram", Journal of Object Technology, vol. 9, no. 5, pp. 1-27, 2010.
27. Santosh Kumar Swain and Durga Prasad Mohapatra "Test Case Generation from Behavioral UML Models", International Journal of Computer Applications, vol. 6, no. 8, September, pp. 5-11, 2010.
28. Sangeeta Sabharwal, Ritu Sibal and Chayanika Sharma,"Applying Genetic Algorithm for Prioritization of Test Cases Scenarios Derived from UML Diagrams", International Journal of Computer Science Issue,Vol. 8, issue 3, No. 2, , pp. 433-444, 2011.
29. Srinivas N, Deb K, "Multiobjective optimization using nondominated sorting in genetic algorithms", Evolutionary Computation, vol. 2, pp. 221-248, 1994.
30. V. Chankong and Y. Haimes, "Multi-objective decision making theory and methodology". New York: North-Holland, 1983.
31. Vikas Panthi, Durga Prasad Mahopatra," Automatic Test Case Generation using Sequence Diagram", Proceeding of ICADC, AISC 174, pp.277-284, 2013.
32. V. Mary Sumalatha, G.S.V. P.Raju," Object Oriented Test Case Generation Technique Using Genetic Using Genetic Algorithms", International Journal of Computer Application(0975-8887), Volume61-No.20,January 2013.