Automated Test Case Generation using Nature Inspired Meta Heuristics- Genetic Algorithm: A Review Paper

Rijwan Khan[1], Mohd Amjad [2] and Dilkeshwar Pandey[3]

[1]Research Scholor Department of Computer Engineering
Jamia Millia Islamia- A Central University, New Delhi

[2]Assistant Professor Department of Computer Engineering
Jamia Millia Islamia- A Central University, New Delhi

[3]Professor & HOD Department of Computer Science & Engineering
ABES IT, Ghaziabad

ABSTRACT

Software testing is an important phase in software development process. Software testing guarantees for reliability and accuracy in delivering of quality software. Test case are mainly generated for evaluation of software. Manual generation of test cases is time consuming task, so automatic test case generation is preferred. In automatic generation of software test case fro a software/program, an optimized technique or algorithm plays a great role. For optimization, go genetic algorithm is a better chance. Here we discuss research works in which concept of genetic algorithm has been applied.

Keywords:- Automatic test case generation, Software testing, Test Cases, Genetic Algorithms

1.INTRODUCTION

1.1 SOFTWARE TESTING AND AUTOMATIC TEST CASES

As we know that software testing is most important for software industry because it delivers and provides the quality of software to the customer. Software testing tells us how much this software is user-friendly, productivity, probability and proficiency etc. Software testing is time consuming task; it spends almost 50% of software development resources. Software testing is not only used in just debugging and detecting bugs it’s also performs the following functions.
(i) Improving and assuring the quality of software
(ii) Verification and validation of software and
(iii) Estimating reliability of software.

Software testing techniques are classified into two categories- static testing & dynamic testing. In static testing, specification documents, design documents and source code software under test (SUT) are used while the source code is examined statement by statement without executing the software. Thus static testing methods are inspections, desk checking, code review etc. In dynamic testing SUT is executed on input test data and the output is observed. The quality & significance of overall testing is directly affected by the set of test cases that are used during testing. So the automatic test cases generation can reduce the cost of software development. For increasing the efficiency of software testing we need optimal test cases.

1.2 GENETIC ALGORITHM

In 1975, Holland written a book named “Adaptation in natural and artificial systems” in this book he developed the idea of genetic algorithm. Holland described how can natural evolutions algorithms are applied to optimization problems and built the first Genetic Algorithms. Holland’s theory has been further developed and now Genetic Algorithms (GAs) stand up as a powerful tool for solving search and optimization problems. Genetic algorithms are based on the principle of genetics and evolution. Holland proposed GA as a heuristic method based on “Survival of the fittest”. A genetic algorithm is an evolutionary algorithm in which we solve optimization problem. We find approximate solutions to optimization problems with GA. It handles a population of possible solutions. Each solution is represented through a chromosome, which is just an abstract representation. The genetic algorithm loops over an iteration process to make the population evolve. In the each iteration we have the following steps:
(I) Initialize population;

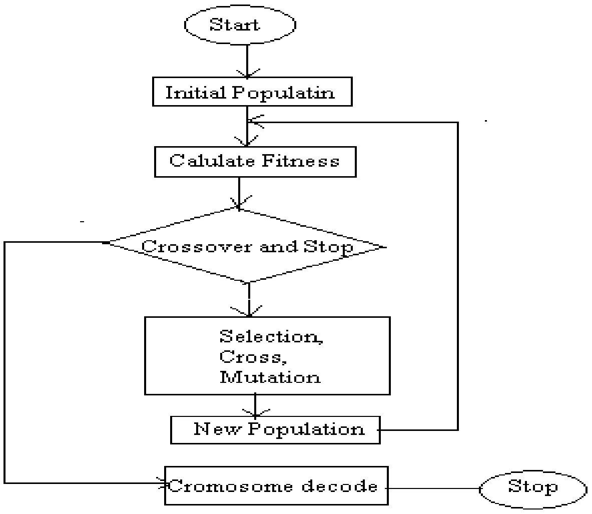
(ii) Evaluate population;

(iii) while Termination Criteria Not Satisfied

(iv) Then select parents for reproduction;

(v) Perform recombination and mutation;

(vi) Evaluate population;



(Wang Xibo, Su Na, 2011)

1.3 GENETIC ALGORITHM IN SOFTWARE TESTING

The key problem in software testing is to generate test case and its automation. For improvement of efficiency and effectiveness and lowers the cost of software testing we have to find the optimal test cases automatically. Simple random method is not enough to generate adequate amount of test data. Therefore there is need for generating test data using search based technique (Ghiduk and Girgis, 2010). In these search based techniques genetic algorithm is more efficient than other optimization algorithms. Andreas S. Andreou (2011) et. al., proposed a method for specially designed genetic algorithm for automatically generating test cases based on data flow coverage criteria. The performance of their proposed approach is assessed and validated on a number of sample programs of different levels of size and complexity. The associated experimental results indicate successful performance in terms of testing coverage. Dan Liu (2013) et. al. proposed a Modified Genetic Algorithm (MG). The algorithm improved fitness function and adopts real number coding and the principles of logic coverage. They also added genetic-oriented control. The algorithm avoids premature convergence phenomenon and is conducive to population diversity. In their experimental result they proved that MGA has faster convergence speed and higher test data generation efficiency other than traditional genetic algorithm. Ghiduk and Girgis et. Al (2008) proposed a concept of dominance relations between the nodes of Control Flow Graph to reduce the software testing cost. They defined a new fitness function in which they used dominance relationship to evaluate the generated test data. Experiments have been carried out by them to evaluate the effectiveness of the proposed GA technique and to reduce the cost of software testing. The results showed that the proposed GA technique outperformed the Random technique. Testing on some C++ programs was used for showing the results. M.Parthiban (2008) wrote that software industry produces high quality software, in producing high quality software’s to the customer, software testing plays a vital role. In software testing automatic generation can reduce the process time and cost. Automatic generation of object-oriented unit tests in software testing is a challenging task especially in white box testing. In software engineering Software testing becomes a big out daring issue with the development of software tools. Using Genetic Algorithm will provide best results in the software testing toolkit. In their paper, they suggested an upgrading of automated search-based test generation of test cases in order to obtain high branch coverage in white box testing. Peng Lin, Xiaolu Bao (2012) et al. A novel algorithm is proposed to support test case generation of combination design in their paper. First of all, the combination-index table (CIT) is defined to guide the process of test case generation, based on which the adaptive genetic algorithm (AGA) is proposed to generate test cases. Sandra Rapps (1985) et al. wrote a paper. Their paper defines a family of program test data selection criteria derived from data flow analysis techniques similar to those used in compiler optimization. It is argued that currently used path selection criteria, which examine only the control-flow of a program, are inadequate. Their procedure associates with each point in a program at which a variable is defined, those points at which the value is used. Several test data selection criteria, differing in the type and number of these associations, are defined and compared. Wang Xibo, Su Na (2011) et. al discussed that for improving the automation ability of software, the software testing is used to reduce cost and ensuring software quality. Their paper discusses the methods and techniques of genetic algorithm need to solve in realization process: such as coding, the selection of fitness function and the improvement of hereditary operator, etc and also generate test cases. Yang et. al., (2009) proposed an approach of generating test data for a specific single path based on genetic algorithms. They apply GA to search suitable solutions for that a similarity between the target path and execution path with sub path overlapped is taken as the fitness function to evaluate the fitness for individuals. They also conducted several experiments to examine the effectiveness of the designed fitness function, and evaluated the performance. Yuehua Dong (2011) et. al. discussed that software testing is the key of guaranteeing software quality delivering. For automation of software testing the generation of test data is one of the key step therefore the generation of testing data is relating the quality of software production indirectly. They have applied the improved genetic algorithm for automatic test case generation with some experiment analysis and showed in their experiment that the improved genetic algorithm is superior to the basic genetic algorithm on effectiveness and efficiency of automatic test case generation.

2. CONCLUSION

The optimization techniques have been applied by many researchers for automatic test case generation but no one could achieve the best performance for every piece of code. So the automatic test case generation has become an NP problem hence scope remains open to apply some more techniques to achieve better results. We think that genetic algorithm itself can't provide better result. Now we will design an optimization technique for automatic test case generation by combining genetic algorithm and cuckoo search.

REFERENCES

[1] Adaptation in Natural and Artificial Systems, John H. Holland, The University of Michigan Press, Ann Arbor, MI, 1975.

[2] Andreas S. Andreou, Kypros A. Economides, Anastasis A. Sofokleous, An automatic software test-data generation scheme based on data flow criteria and genetic algorithms, Seventh International Conference on Computer and Information Technology, 0-7695-2983-6/07 $25.00 © 2007 IEEE.

[3] Binitha S, S Siva Sathya, A Survey of Bio inspired Optimization Algorithms, international journal of soft computing and engineering(IJSCE) ISSN: 2231-2307, Volume-2, issue-2, May 2012.

[4] Chen Yong, Zhong Yong, Tingting Shi1, Liu Jingyong, (2009), "Comparison of Two Fitness Functions for GA-based Path-Oriented Test Data Generation" , Fifth International Conference on Natural Computation, IEEE 2009.

[5] Dan Liu, Xuejun Wang, Jianmin Wang,Automatic Test Case Generation Based On Genetic Algorithm, Journal of Theoretical and Applied Information Technology 10th February 2013. Vol. 48 No.1

[6] Ghiduk, Ahmed S and Girgis, Moheb R.(2010), Using Genetic Algorithms and Dominance Concepts for Generating Reduced Test Data, Informatica (Slovenia), Volume 34, Number 3, pp.377-385.

[7] M.Parthiban, M.R.Sumalatha, GASE -An Input Domain Reduction and Branch Coverage System Based on Genetic Algorithm and Symbolic Execution 2008.

[8] Praveen Ranjan Srivastava, Priyanka Gupta, Yogita Arrawatia, Suman Yadav, Use of Genetic Algorithm in Generation of Feasible Test Data, SIGSOFT Software Engineering Notes, March 2009, Vol 34 No.2.

[9] Peng Lin, Xiaolu Bao, Zhiyong Shu, Xiaojuan Wang, Jingmin Liu, Test Case Generation Based on Adaptive Genetic Algorithm, 978-1-4673-0788-8/12/$31.00 ©2012 IEEE

[10] Sandra Rapps, Elaine J. Weyuker, Selecting Software Test Data Using Data Flow Information, IEE Transaction on software engineering, Vol. SE-II, No. 4, April 1985.

[11] Wang Xibo, Su Na, Automatic Test Data Generation for Path Testing Using Genetic Algorithms, 2011 Third International Conference on Measuring Technology and Mechatronics Automation, 978-0-7695-4296-6/11 $26.00 © 2011 IEEE.

[12] Yang Cao, Chunhua Hu and Luming Li(2009), An Approach to Generate Software Test Data for a Specific Path automatically with Genetic Algorithm, International Conference on Reliability, Maintainability and Safety, pp.888-892.

[13] Yuehua Dong ,Jidong Peng, Automatic Generation of Software Test Cases Based on Improved Genetic Algorithm, IEEE 2011.

[14] S. N. Sivanandam, S. N. Deepa, Introduction to Genetic Algorithm, Springer, book.

AUTHOR

Rijwan Khan received the B.Tech degree in Computer Science and Engineering from BIT in 2006 and M.Tech degree in Computer Science and Engineering from IETE Delhi in 2010. Now he is pursuing his Ph.D degree in the department of Computer Engineering from Jamia Millia Islamia- A Central University, New Delhi.