International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Software Test Case Generation and it's curtail using G-Genetic Algorithm

K Koteswara Rao, P Anil Kumar, Ch. Chandra Mohan

Abstract: ***Project is a collection of similar activities that are going to be executed in certain order. Among the phases of project management testing show business crucial role. The intension of testing is not to prove the correctness; it is the process of verifying and validation. Software Testing is the most challenging job among all the peers of the industry. Exhaustive software Testing is never possible only Optimized software testing is possible. Hence Software Testing can be viewed as optimization problem as it fall under NP complete. Because of the extensive number of experiments that are required to perform adequate testing of the ideal programming application; the different strategies to decrease the test suite is required. One of the normal contemplated strategies is evacuating the repetitive experiments; the reason is insignificant number of experiments and greatest number of mistakes seclusion or revealing. In this exploration work consider is directed to address the usage and viability of G-hereditary calculation so as to decrease the quantity of experiments that don't included unmistakable incentive in the mean of test inclusion or where the experiments can't separate blunders. Hereditary calculation is used in this work to help in limiting the experiments or streamlining the experiments, where the hereditary calculation creates the primer populace arbitrarily, computes the wellness esteem utilizing inclusion measurements, and after that particular the posterity in back to back ages utilizing hereditary tasks choice, traverse and transformation. The hereditary displaying activities are explicit and dependent on the task may fluctuate to ordinary Genetic calculation. This procedure of age is rehashed until there is no adjustment in the wellness esteems for two successive ages, when there is no adjustment in the information age for two emphases so union accomplished or a minimized test case is achieved. The results of study demonstrate that, genetic algorithms can significantly reduce the size of the test cases***

Index Terms: G-Genetic, Testing, Software, curtail.

I. INTRODUCTION

Testing is a movement of evaluating the framework or its segments with the resolved to discover whether it satisfies the measured necessities or not. This action outcome in the unmistakable likely and fluctuation between their outcomes. In humble words it is executing a framework with a specific end goal to find out mistakes or missing prerequisites in opposition to the real longing or necessities.

In some cases it can be characterized as "movement of researching programming thing to recognize the contrasts amongst existing and required conditions and to assess the elements of the product thing".

Importance of Testing

Amid configuration and development programming is tried to reveal mistakes [10]. Testing is a fundamental part of any undertaking or process created to yield the wanted yield. Especially in programming in any upsetting environment, criticality develops with the unpredictability and size of the prerequisites. The IT world has seen numerous fiascos in light of the disappointment of programming items [10]. Presently a day in each industry, guaranteeing the quality and dependability of programming items has turned into a vital issue. Along these lines, to guarantee programming dependability testing is extraordinary and most requesting errands in programming progress. It finds issues and guarantees quality, agreeableness. The objective of testing is to discover issues, not to demonstrate rightness. And importance of the testing was explained in the article optimizing the Software testing efficiency by Using A Genetic Algorithm; A Design Methodology: ACM SIGSOFT Vol 38, Issue3, May 2013[10]

1.1Testing Types:

Hands on Testing: This write contains the testing of the Software physically i.e. without utilizing any modified device or any script. In this write the analyzer assumes control over the part of an end client and tests the Software to determine any un-expected execution or bug. There are diverse stages for industrial testing like unit, Integration, System and User Acceptance testing [9]. Analyzers use test arrangement, experiments or test circumstances to test Software to make sure breadth of testing. Hands on testing likewise incorporate observational testing as analyzers find the product to distinguish blunders in it.

Motorization Testing: It is likewise outstanding as "Test Automation", here the analyzer engraves scripts and practices programming to test. This course includes computerization of a manual procedure, is utilized to re-run the test situations that were done physically, rapidly and more than once. Aside from relapse testing, Mechanization testing is additionally used to test the application from burden, execution and emphasize perspective. It accelerations the test scope; progress precision, spares time and spending plan than others.

When to Automate: It is best suited in the following:

- Hefty and acute projects.

Revised Manuscript Received on 30 July 2019.

***** Correspondence Author

Dr. K Koteswara Rao*, Department of CSE, PVPSIT, Vijayawada, Andhra Pradesh, India.

Mr P Anil Kumar, Department of CSE, PVPSIT, Vijayawada, Andhra Pradesh, India.

Mr Ch. Chandra Mohan, Department of CSE, PVPSIT, Vijayawada, Andhra Pradesh, India.

- Chucks are stable.
- Retrieving the application with numerous users.
- Unchanging Software.
- Convenience of stint.

Step by step instructions to Automate: Mechanization is done by using a concerned computer language, lot of tools open for the purpose of automation cursives. The procedure is:

- Ascertaining regions.
- Appropriate tool assortment.
- Inscription of scripts.
- Test suits development.
- Accomplishment of scripts
- Creating outcome information.
- Recognizing impending bug or recital issue.

Some of the tools used for Mechanization testing

- HP Quick Test Proficient
- SELINIUM
- RR serviceable Tester
- SILK Test
- Test Comprehensive
- Testing Anywhere
- Win Runner
- Load Runner
- Visual Studio Professional
- WATIR

Testing Documentation

It envelops the accreditations of relics which ought to be built up all through the testing of Software, helps in approximating the testing exertion essential, test inclusion, prerequisite following/following and so on. It contain

- Test Proposal
- Scenario
- Test Case
- Traceability Matrix

Test Proposal

It skeletons the approach that will be used to test an application, Stereotypically the Eminence Guarantee Team Lead is accountable for writing a Test Plan which contain

- Introduction of the proposal
- Assumption conditions
- List of Test cases
- The features to be tested
- Type of Approach
- What are the Deliverables
- Resource allocation
- Risks assessment
- Tasks and milestones

Test Scenario

It clarifies what locale will be tried; guarantee that all method floats are tried from end to end. The term test situation and experiments are utilized. At the point when seen from this observation test situations are experiments.[9] [10]

It includes the arrangement of steps, conditions and commitments which can be utilized while execution the testing undertakings. There are various sorts of experiments alike Functional, undesirable, botch, sensible, lustful, UI experiments and so on. Moreover test cases are engraved to keep way of testing scope of Software. More often than not, there is no legitimate layout which is utilized through the experiment composing, the principle segments are:

- **Test case ID**.
- Invention Segment
- Merchandise variety
- Amendment antiquity
- Determination
- Molds
- Pre-Conditions.
- Stages.
- Predictable Outcome.
- Authentic Outcome.
- Pole Circumstances.

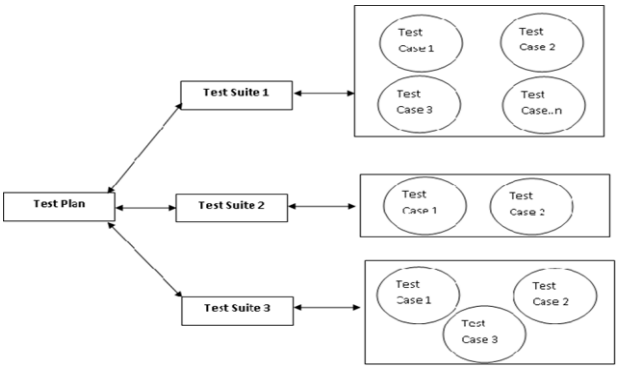
Test case Design Practices

Following are the distinctive design practices

1. Test case generation form the requirements. This includes:
    - Frontier Rate Study
    - Sameness Apportioning
    - Pronouncement Stand Taxing
    - State Changeover Illustrations
    - Use- Case Testing
    - 2. Test case derivation from construction:
    - Declaration Exposure
    - Division Exposure
    - Track Exposure
    - 3. Test case derivation based on experience:
    - Fault Predicting
    - Probing Testing

What is a Test Suite?

It is a holder that has a lot of tests which helps analyzers in executing and announcing the test execution status. It can take any of the three conditions in particular Active, in advancement and finished. A Test case can be added to various test suites and test plans. In the wake of creating a test plan, test suites are formed which thusly can have any number of tests. Test suites are shaped dependent on the succession or dependent on the likelihood. It can contain any kind of tests, viz - utilitarian or Non-Functional[1]



- **Figure 1. Test Case Execution With Test Suite**

II. METHODOLOGY

Stage1: Generation of Test Cases

Programming testing is huge and unquestionable field, repeating the contrasting necessities that component antiquated pieces must fulfill the different exercises required in testing and the various dimensions at which programming can be checked.

Confident testing makes test inputs abstractly from the data space of the thing under test. The fundamental segment of an eccentric test time system is that it produces test duties at emotional from a language structure or some other formal relic clearing up the data space. The most fundamental clarification behind the proposed technique is to gather a compelling framework for diminishing common blemishes in light of ideal test in direct self-self-assured testing. For the time of examinations the proposed procedure is utilizing Object Behavior Dependence Model The resulting information sources are upheld to the GA; the fated technique for executed system is talked about to in Figure The square structure of the recommended philosophy is talked about to in underneath.

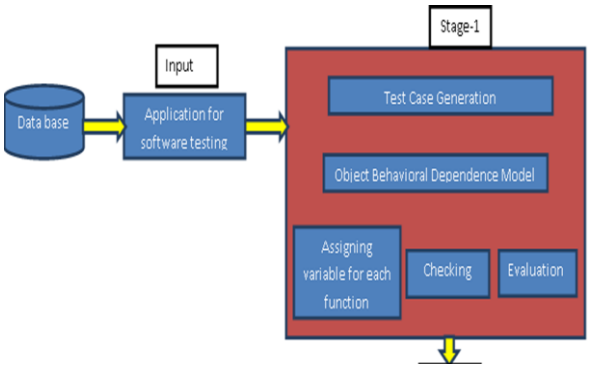


Figure2 Block Diagram of methodology

Stage2: Reduction of Test Cases

In our proposed technique, we make huge preliminaries in which a portion of those examinations are not required for dealing with. In comparable way there might be identical examinations that are being set aside a few minutes break. With a specific genuine goal to pick the concerned preliminaries we require some convincing techniques. The going with time of the recommended framework is false diminishment by system for the GA. In this examination, the ideal information sources will be made in context of Genetic Algorithm (GA) which will diminish the unlawful data sources and relative wellsprings of data.

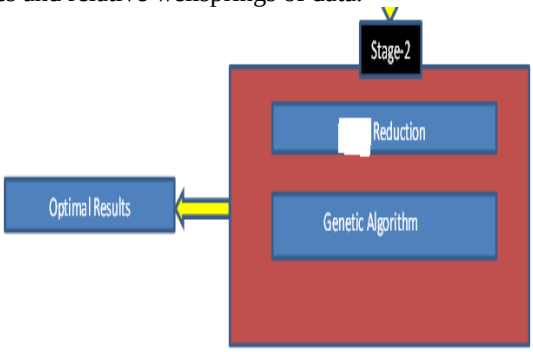


Figure3 Block Diagram of stage 2

The simple form of GA[10]

1. Flinch by methods for indiscriminately incited people
2. Calculate the wellness of every chromosome in the populace
3. Repeat the accompanying strides until n off springs have been made
4. Select a couple of parent chromosomes from current populace
5. With likelihood Pc hybrid the pair at haphazardly picked point to frame two posterity's
6. Mutate the two posterity's at every locus with likelihood Pm
7. Replace the present populace with the new populace
8. Go to Step 2

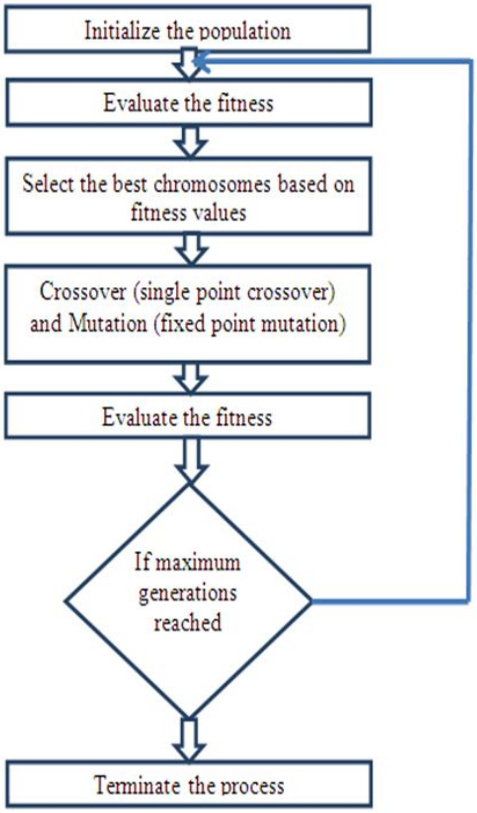


Figure4. Flow of Genetic Algorithm

III. RESULTS AND DISCUSSION

Results and Future work. In this section and Future implementation, we will take the software application as input and for that we will generate the test cases in the phase one. In the phase two the generated test cases can be minimized using Genetic Algorithms by following the specified procedure. And results can be tabulated for the parameters 1. Fitness Value 2.Test case count 3.Time usage 4.Memory and any other parameter. Sample figures are As shown below

Iterations.	Fitness. Standards using GA
25.	625
50	750
75	699
100	682

Figure5: Fitness comparison

Iterations	Test case count	
	Without GA	With GA
25	756	446
50	756	430
75	756	478
100	756	443

Figure6: Test case count comparison

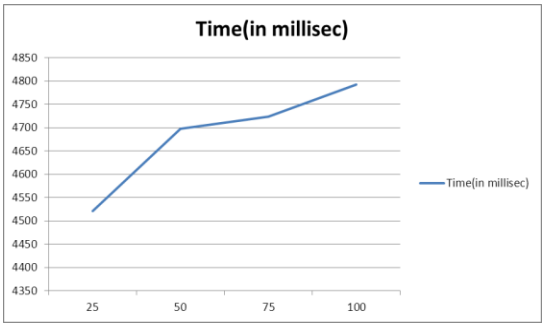


Figure7: Time utilization comparison

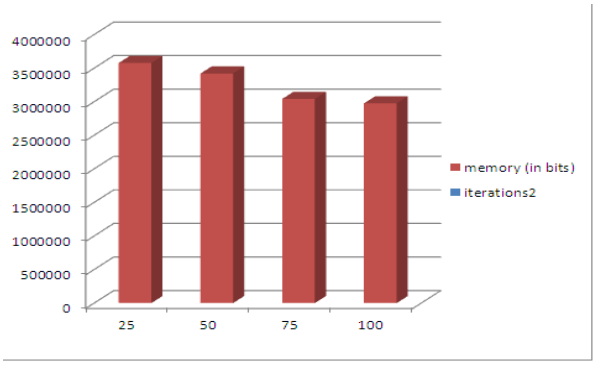


Figure8: Memory utilization comparison

REFERENCES

1. Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to genetic algorithms*. Springer Science & Business Media.
2. Sivanandam, S. N., & Deepa, S. N. (2007). *PRINCIPLES OF SOFT COMPUTING (With CD)*. John Wiley & Sons.
3. Dr.K.V.K.K Prasad (2008), "Software Testing Tools", Dreamtech, ISBN 81-7722-532-4. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
4. Waman S Jawadekar, "Software Engineering principles and practice " The McGraw-Hill Companies, ISBN 13: 9780070583719
5. Somerville. "Software Engineering". Addison-Wesley, 1995, ISBN-0201427656.
6. S.N. Sivanandam, S. N. Deepa (2012)."Principles of Soft Computing"2nd Edn, 978-81-265-2741-0, Wiley- India
7. S. Rajasekaran, G. A. Vijayslakshmi Pai(2007) 'Neural Networks, Fuzzy Logic and Genetic Algorithms "PHI 2003 ISBN 971-81-203-2186-1.
8. KK Rao, GSVP Raju "Theoretical Investigations to Random Testing Variants and its Implications." International Journal of Software Engineering and Its Applications Vol.9, Not. 5(2015). pp. 165-172.
9. Rao, K. K., Raju, G. S. V. P., & Nagaraj, S. (2013). Optimizing the software testing efficiency by using a genetic algorithm: a design methodology. *ACM SIGSOFT Software Engineering Notes*,*38*(3), 1-5..

AUTHORS PROFILE

Dr K Koteswara Rao completed B.Tech in CSE from ANU, M.Tech In CSE from JNTUH, PhD in CSE from JNTUK. Research Adviser to Nan Yang Academy of Scinces Singapore, Published 40 research articles, among them 15 are Scopus indexing and 4 are ISI web of Science, Clarivate analytics indexing. Reviewer for various peer reviewed Journals, Research interest includes Soft Computing and Software Engineering. Member of IACSIT, IAENG, Received best teacher and best researcher award.

Mr P Anil Kumar completed B.Tech in CSE from JNTUH and M.Tech in CSE from JNTUK Kakinada. Published 8 research articles in reputed International Journals and Conferences.. Interested in Research areas like Internet of Things, Computer Networks, Cloud Computing and Data Analytics. He published and applied 1 Patents and waiting for results. Mr P Anil Kumar is Life member Computer Society of India, ISTE, and Professional member of ACM,

Mr Ch. Chandra Mohan completed B.Tech in CSE from JNTUH and M.Tech in CSE from JNTUK. Published 8 research articles in reputed International Journals and Conferences.. Interested in Research areas like Internet of Things, Computer Networks, Cloud Computing and Data Analytics. He published and applied 1 Patents and waiting for results. Mr P Anil Kumar is Life member Computer Society of India, ISTE, and Professional member of ACM,