

Modified ACO model for Regression Testing Using Automated slicing

¹Riza Dhiman, ²Dr Vinay Chopra

¹Mtech (CSE), ² Assistant Professor(CSE)
DAV Institute of Engg. And Technology
Jalandhar, India

Abstract : *The Regression testing is the testing which is applied to test the software when some changes are done in the already developed project. The test case prioritization is the technique of regression testing which prioritizes the test cases according to the changes which are done in the developed project. This work is based on automated and manual test case prioritization techniques. In the existing technique the manual test case prioritization is been implemented to detect faults from the project. In the manual test case prioritization two parameters are considered which are, number of times function encountered and number of functions associated with the particular function. On the basis of these two parameters the importance of each function is calculated which are prioritized by calculating FTV value. The FTV value is calculated according to the changes which are defined in the developed project. To increase the fault detection rate of the test case prioritization, automated test case prioritization is being implemented in this work. In the first step of the algorithm, the population values are taken as input which is the number of times function encountered and number of functions associated with a particular function. In the second step, the algorithm will start traversing the population values and error is calculated after every iteration. The iteration at which the error is maximum at that point the mutation value is calculated as the best mutation value of the function. The function mutation value will be the function importance from where the test cases are prioritized according to the defined changes. In the last step of the algorithm the function importance values are accessed according to the defined changes and best fitness value is calculated which will be the final percentage of faults detected from the project after the particular change.*

IndexTerms - Regression Testing, ACO, Test case prioritization, Automated slicing ,FTV(function traversal value).

I. INTRODUCTION

1.1 Software Testing

Software testing is a procedure of testing or comparing the actual outcome with the expected outcome. Testing of the software is done in order to check the correct functionality of the system or project. If the testing will not be performed then system may lead to catastrophic or improper results in the field. So it's better to check or test the system earlier, so that the excellent results can be produced.

1.2 Types of Software Testing

- **Black-Box Testing**

Black-box testing is also known as functional testing. This testing ignores the internal mechanism of the system. It is a testing which is based upon the output and having no knowledge of internal code. It is attesting in which its working is not understood by its user. It has no knowledge of processing of code but only concentrate upon the output.

- **Integration Testing**

This testing is done when two or more modules are combined together into a larger module. It verifies the functionality of the module after integration. It is done at the interfaces of the both the structure and component module. This type of testing is done in distributed or client/server modules.

- **System Testing**

It is also known as end-to-end testing. It tests the complete application of environments. It is based upon the specification and requirement of the system. It checks the entire systems. It is for high level design and comes under the black-box testing .It also checks non-functional requirements also.

- **Load Testing**

Load Testing is done to determine whether the system is able to handle given no. of users or not. It is done only to check the performance of the system is it working well or not. It checks the behavior under heavy loads and inputs. In web application testing it is used to check where the performance of the system degrade and at which point it fails.

- **Grey-Box Testing**

It is a combination of both white box and black box testing. It contains the knowledge of internal structure of code more than black box testing and less than white box testing. It comes under the correctness testing. It also includes reverse engineering concepts also.

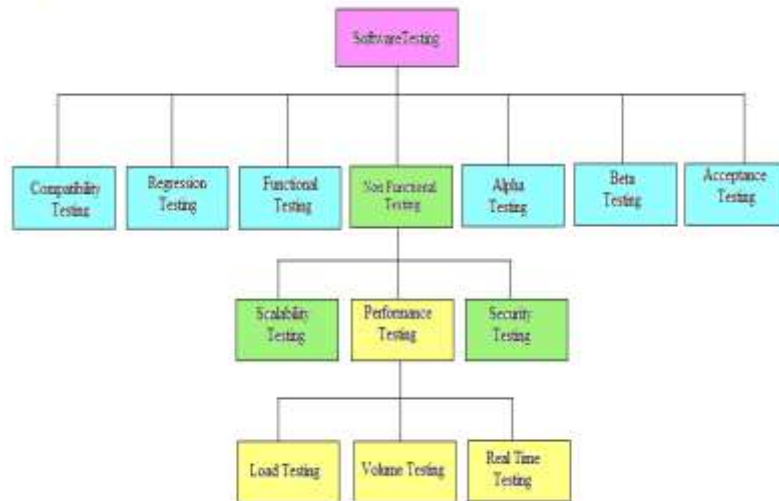


Fig:1 Types of Testing[6].

1.3Regression testing

It is a testing that refers to that section of the test cycle in which programs are tested to make sure that changes do not affect features that are not believed to be affected. The process of verifying the customized software in the maintenance phase is known as Regression testing. Time and budget constraints are its major disadvantage due to complex process. Regression testing is the re-execution of a number of subset of test that has previously been conducted. In regression testing as integration testing takings, number of regression tests increases and it is not practical and ineffective to re execute every test for each program function if once change occurs. It is an expensive testing process used to detect regression faults. Research has shown that at least 50% of the total software cost is comprised of testing activities [1].

Techniques of Regression Testing

- **Retest All**

This is one of the methods for regression testing in which all the tests in the existing test bucket or suite should be re-executed. This is very expensive as it requires huge time and resources.

- **Regression Test Selection (RTS)**

Due to expensive nature of “retest all” technique, Regression Test Selection is performed. In this technique instead of rerunning the whole test suite we select a part of test suite to rerun if the cost of selecting a part of test suite is less than the cost of running the tests that RTS allows us to omit.

- **Test case prioritization**

A mechanism is needed for arranging a test case in appropriate order to increase their effectiveness at meeting some performance goal and rate of fault detection such mechanism is known as test case prioritization. Test case prioritization is a method to prioritize and schedule test cases in appropriate order. To run test cases of higher priority before than the lower priority test case in order to minimize time, cost and effort during software testing phase. Various performance goals are like rate of fault detection which a is a measure of how quickly the fault is detected so that during testing faster feedback can provide about system under testing and allow the software tester to correct the software at earlier phase as possible [7].

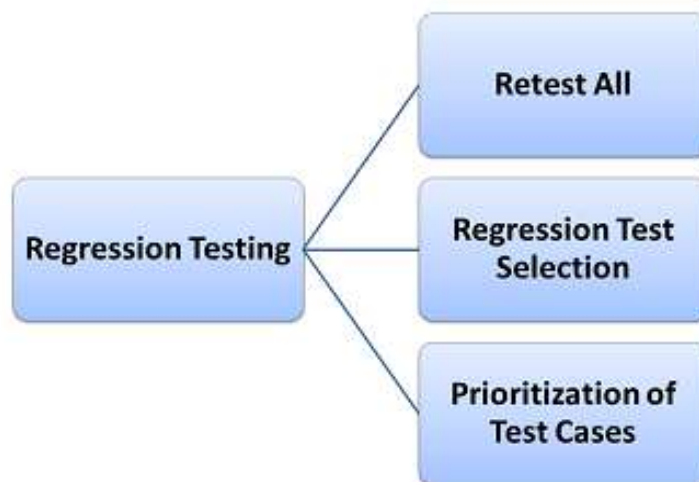


Fig 2 Types of Regression Testing [7].

1.4 Test Case

A test case is set of procedure use to test the software. Test case is a set of condition under which under which a software tester determine whether the application or software system is working correctly or not. To design a test case for particular software the designer must

design positive or negative test case for the software. Positive test cases are design to check software under normal condition and negative test case are design to check software at extreme condition. The order of test case execution affects the time at which goal of testing are fulfil. If the goal is fault detection then a improper execution order might reveal most of fault late which leads to delay in bug fixing activity and the delivery of software.

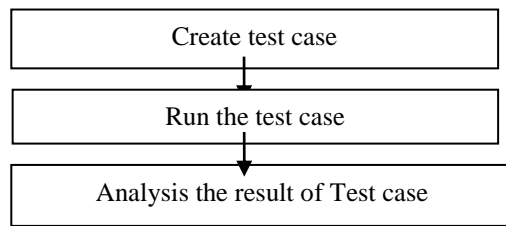


Fig 3 Testing Process [10].

Techniques of Test Case Prioritization

- **Customer-requirement based technique**

In this technique Customer requirement factors are taken into account and provided some weights and based of these values test case weight for requirement is evaluated. Test cases with high weights value are executed first following the ones with lower value. Customers requirement factors are Customer assigned priority on requirement, Requirement complexity and Requirement volatility.

- **Coverage-based technique**

It is based on code coverage analysis and the measurement of code covered by a test case. Various coverage criterions are considered and the amount of coverage is evaluated and used to prioritize the test cases. Coverage-based technique is a white-box testing technique i.e a method that tests internal structures of a software.

- **Cost effective based technique**

This technique prioritizes the test cases which are based upon costs factors like cost of operation of test cases, cost of analysis, cost of prioritization, cost of execution, validating test cases. Cost is of two types i.e Direct cost include test selection, test execution, result analysis and Indirect cost include overhead cost and tool development cost.

- **Chronographic history-based techniques**

This technique prioritizes the test cases based on test case's earlier executions in order to enlarge or reduce the probability that it will be considered into account in current test execution [5].

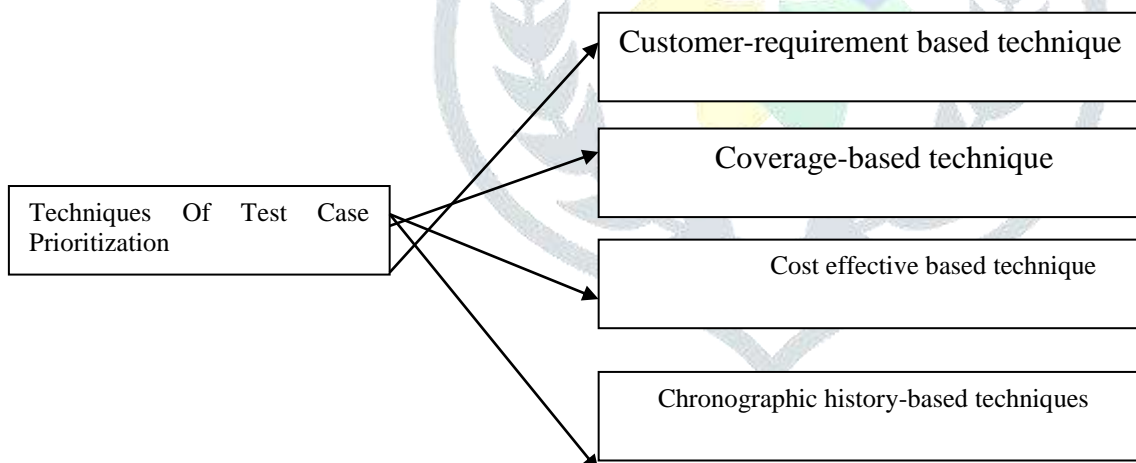


Fig 4 Techniques of Test Case Prioritization [5].

1.6 Need of Regression Testing

- Change in requirements and code is modified according to the requirement.
- New feature is added to the software.
- Defect fixing.
- Performance issues fix.

1.7ALGORITHMS FOR TEST CASE PRIORITIZATION

- **Greedy Algorithm**

It is based upon the principle that the element with the highest weight is taken into account first, followed by the element with second-highest weight and this process continues until a complete solution has been obtained. It is quite a simple algorithm but in some situations where the results are of high quality it is also prove to be attractive one because it is quite inexpensive both in terms of implementation and execution time.

- **Additional Greedy Algorithm**

The Additional Greedy Algorithm is one of the type of Greedy Algorithm, but it follows quite different process. It combines feedback from previous selection and randomly selects the maximum weighted element of the problem from that part that is not being already consumed by the previously selected elements.

- **Genetic Algorithm**

The population is a set of random individuals. In which each individual is represented by the sequence of genes commonly known as the chromosome. In this selection procedure depends upon the fitness value which decides that which individuals are to be selected as the “parents” for producing the next generation. Crossover is a genetic operator which combines two individuals in order to produce a new individual known as offspring. The mutation operator will alter one or more gene values in the individual depending on the probability of mutation.

1.8 Ant colony Optimization Algorithm

Ant colony Optimization algorithm is a mathematical optimization Approach which belongs to the family of local search. Therefore because of this reason it is also known as local search approach. It is an iterative algorithm that starts its search from an arbitrary solution of the problem and then it will find a better solution by simply changing a single element of the solution

The Ant colony Optimization algorithm for test case prioritization is composed of the following steps:

1. Pick a random solution state and make this the current (i.e.initial) state.
2. Evaluate all the neighbors of the current state.
3. Move to the state with the largest increase in fitness from the current state. If no neighbor has a larger fitness than the current state, then no move is made.
4. Repeat the previous two steps until there is no change in the current state.
5. Return the current state as the solution state [3].

Applications of Ant colony Optimization Approach

- Ant colony Optimization can be applied to any problem where the current state allows for an accurate evaluation function. For example, the travelling salesman problem, the eight-queens problem, circuit design and a variety of other real-world problems.
- Ant colony Optimization has been used in inductive learning models
- Ant colony Optimization has also been used in robotics to manage multiple-robot teams which allows scalable and efficient coordination in multi-robot systems.
- Ant colony Optimization allows robots to choose whether to work alone or in team .

II. Research Methodology

The Regression testing is the testing which is applied to test the software when some changes are done in the already developed project. The test case prioritization is the technique of regression testing which prioritizes the test cases according to the changes which are done in the developed project. This work is based on automated and manual test case prioritization techniques. In the existing technique the manual test case prioritization is been implemented to detect faults from the project. In the manual test case prioritization two parameters are considered which are, number of times function encountered and number of functions associated with the particular function. On the basis of these two parameters the importance of each function is calculated which are prioritized by calculating FTV value. The FTV value is calculated according to the changes which are defined in the developed project. To increase the fault detection rate of the test case prioritization, automated test case prioritization is being implemented in this work. In the first step of the algorithm, the population values are taken as input which is the number of times function encountered and number of functions associated with a particular function. In the second step, the algorithm will start traversing the population values and error is calculated after every iteration. The iteration at which the error is maximum at that point the mutation value is calculated as the best mutation value of the function. The function mutation value will be the function importance from where the test cases are prioritized according to the defined changes. In the last step of the algorithm the function importance values are accessed according to the defined changes and best fitness value is calculated which will be the final percentage of faults detected from the project after the particular change.

2.1 Working of existing ACO algorithm

Following are the various steps of the existing ACO algorithm

1. In the existing ACO algorithm, the function importance is calculated on the basis of number of times the function encounter and number of functions associated with the particular function
2. The algorithm takes the input function importance and function which is maximum importance is considered as the best value.
3. The best value is taken into account for the test case prioritization. The function which has maximum importance, test cases of that functions are executed first and so on in the decreasing order

2.2 Steps of the proposed ACO algorithm

Following are the various steps of proposed ACO algorithm

1. In the improved multi-objective algorithm, the function importance is also calculated on the basis of number of functions associated. The function which has maximum association is considered as the most important function
2. To calculate the number of functions associated, the technique of automated slicing is been applied which traverse the DFD and generate final result.
3. The automated slicing will work in the iterative manner and search the best value of the test case as which maximum number of errors get detected from the project

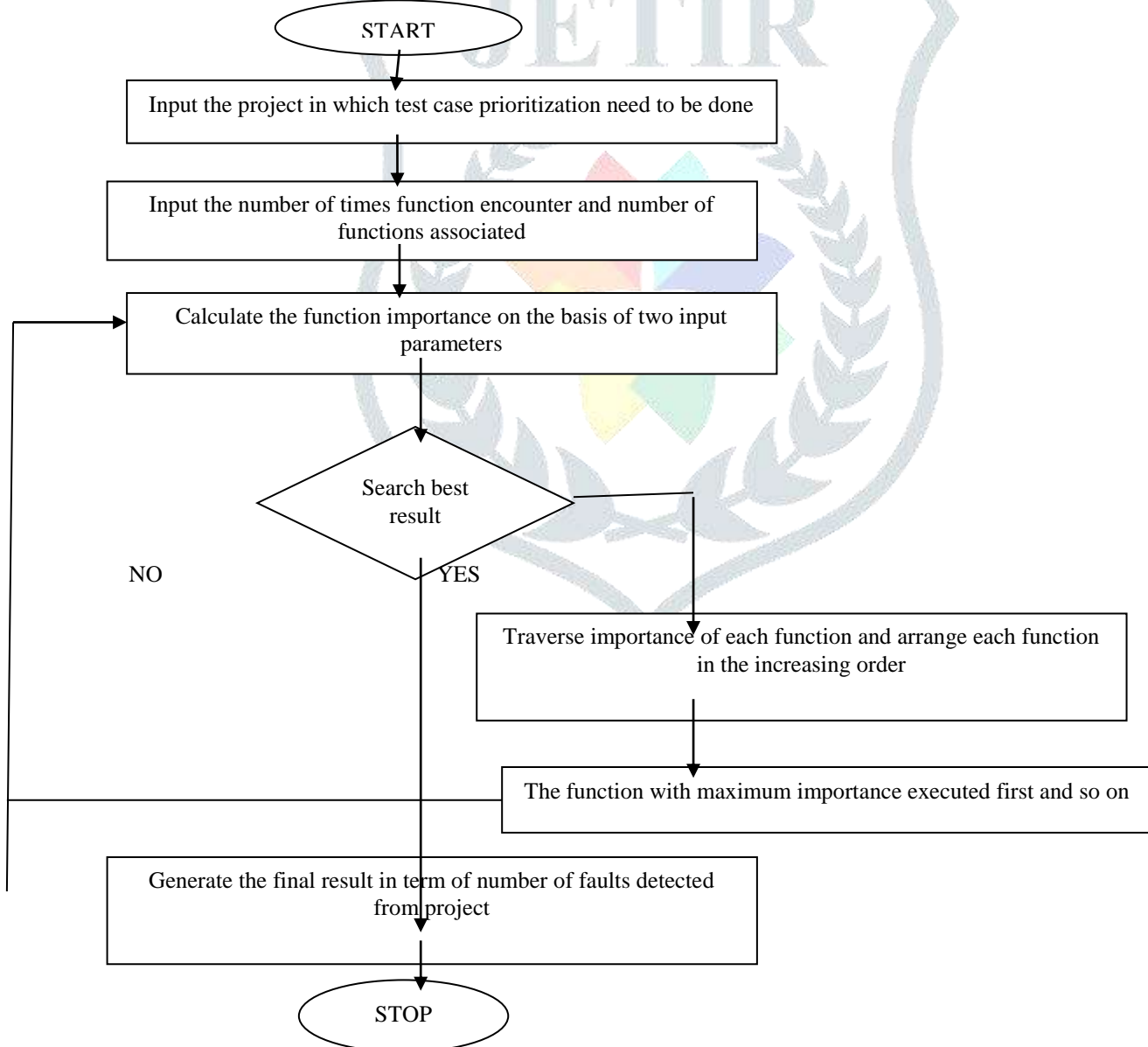
2.3 Proposed Algorithm

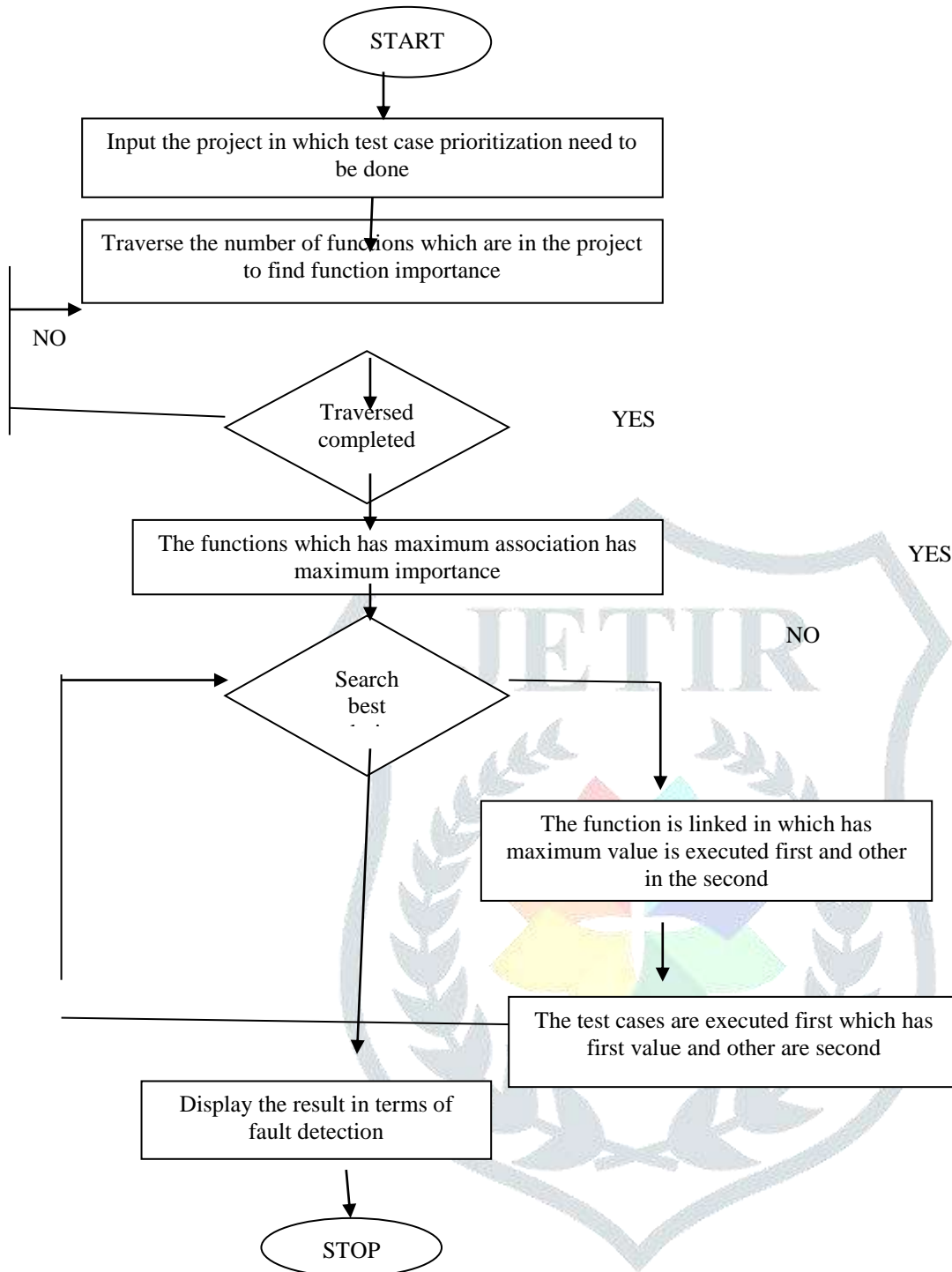
Input :
 Test cases= $P(i)$
 Number clicks on each function = $F(i)$
 Output: prioritized testcases

```

I<-Consider value of F(i) for the each test case
Test case F(i) value <- i
while ( fault value of each test case is calculated )
    a=F(i)
    calculate number of links L(i)=F(i)/F(i)
    if(L(i)>L(i+1)
        b=L(i)
    else
        b=L(i)
    end
Calculate fault value Fault (i+1)=fault(i)/L(i)
if Fault(i) > Fault(i+1)
    best_so_far <-Fault(i)
i <- generate an individual randomly
end
    
```

Flowchart of Existing Algorithm



Flowchart of proposed work**III. Implementation****3.1 Tool Description**

The matlab is the tool which is used to perform mathematical complex computations. In this MATLAB simplified C is used as the programming language. The MATLAB has various inbuilt toolboxes and these toolboxes are mathematical toolbox, drag and drop based GUI, Image processing, Neural networks etc. The MATLAB is generally used to implement algorithms, plotting graphs and design user interfaces. The MATLAB has high graphics due to which it is used to simulate networks. The MATLAB has various versions by current MATLAB version is 2015. The MATLAB process elements in the form of MATRIXs and various other languages like JAVA, PYTHON and FORTAN are used in MATLAB. The MATLAB default interface has following parts

1. **Command Window:-** The Command Window is the first importance part of MATLAB which is used to show output of already saved code and to execute MATLAB codes temporarily
2. **WorkSpace :-**The workspace is the second part of MATLAB which is used to show allocation and deallocation of MATLAB variables. The workspace is divided into three parts. The first part is MATLAB variable,variable type and third part is variable value
3. **Command History :-** The command history is the third part of MATLAB in which MATLAB commands are shown which are executed previously
4. **Current Folder Path :-** The current Folder path shows that path of the folder in which MATLAB codes are saved
5. **Current Folder Data: -** The Current Folder Data shows that data which is in the folders whose path is given in Current Folder Path

REFERENCES

- [1] Leung H., White L. "Insight into regression testing," in Proc. 27th IEEE Int.Conf.Softw. Eng., Vol. 20,pp. 60-69,1989.
- [2] Rothermel G., Chu C., "Prioritize the test case for Regression testing," *IEEE Transactions on Software Engineering*,Vol.5,2001.
- [3] Elbaum S., Rothermel G., "Selecting a Cost-Effective Test Case Prioritization Technique," in Proc. IFIP International Conference Testing Software System, Vol.45, pp. 90–95, 2004.
- [4] Khan S., Awais A. "TestFilter:A Statement-Coverage based test case reduction technique," in Proc. 12th IEEE International Conference Engineering, Vol.11, pp. 5–12, 2006.
- [5] Harman Z., Hierons R. "Search algorithms for regression test case prioritization," in Proc. 12th IEEE International Journal Software Engineering, Vol.33, pp. 225–237,2007.
- [6] Harman Z., Hierons R. "Test case prioritization: An empirical study," in Proc. IFIP International Conference Testing Software System, Vol.35, pp. 85–90, 2009.
- [7] Tonella P., Avesani P., "Using the Case-Based Ranking Methodology for Test Case Prioritization," in 22nd IEEE International Conference on Software Maintenance,2009.
- [8] Daengdej J. "Test case prioritization techniques," in Proc. IEEE International Journal Software Engineering Knowledge Engineering, Vol. 22, no. 2, pp. 161–183, 2010.
- [9] Mirarab S., "The effect of time constraint on test case prioritization," *IEEE Transaction on Software Engineering* ,Vol.36, No.7,2010.
- [10] Chen L., Wang Z., Xu L., Lu H., Xu.B., "Test Case Prioritization for Web Service Regression Testing," in Proc. Fifth IEEE International Symposium on Service Oriented System Engineering ,Vol.46, No.8,2010.
- [11] Malhotra R., Kaur A., Singh Y., "A Regression Test Selection and Prioritization Technique," *Journal of Information Processing Systems*, Vol.6, No.2, 2010.
- [12] Srikanthi H., Williamsi L.,Osborne J., "System test case prioritization of new regression test case," *IEEE Transaction on Software Engineering* ,VOL.36, No.2,2011.
- [13] Zhang X., Uma G.,"Factors Oriented Test Case Prioritization Technique in Regression Testing using Genetic Algorithm", *European Journal of Scientific Research*, Vol. 74, No.3,2012.
- [14] Haidry S., Miller T., "Using Dependency Structures for Prioritisation of Functional Test Suites", *IEEE Transaction on Software Engineering* ,VOL.38, No.8,2012.
- [15] Jatain A., Sharma G., "A systematic review of techniques for Test case prioritization," *International Journal of Computer Applications*, Vol. 68, No. 2,2013.
- [16] Joshi S., Adiga D., Tiple B., "Effective use of Prim's Algorithm for Model Based Test Case Prioritization", *International Journal of Computer Science and Information Technologies*,Vol.34, No.16, 2014
- [17] Muthusamy T., Seetharaman K., "Effectiveness of Test Case Prioritization Techniques Based on Regression Testing, Vol. 54, No. 15,2014.
- [18] Konsaard P., Ramingwong L., "Total Coverage Based Regression Test Case Prioritization using Genetic Algorithm," in Proc. IEEE International Journal Software Engineering Knowledge Engineering, Vol. 24, no. 5, pp. 24–31, 2015.
- [19] Wang H., Xing J., Yang Q., "Modification Impact Analysis based Test Case Prioritization for Regression Testing of Service-Oriented Workflow Applications," in Proc. 39th IEEE International Journal Software Engineering Knowledge Engineering, Vol. 30, No. 8,2015.