ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Fuzzy Logic Optimization Regression Test Suite using GA

Deepak Dixit, Dr. Dinesh Kumar, Rohit Singla, Hemant Sharma

M.Tech Student, Dept. of Computer Science& Engineering, Shri Ram college of engineering and Management, Palwal, affiliated to M.D.U Rohtak (Haryana), India

H.O.D, Dept. of Computer Science& Engineering, Shri Ram college of engineering and Management, Palwal, affiliated to M.D.U Rohtak (Haryana), India

M.Tech Student, Dept. of Computer Science& Engineering, Shri Ram college of engineering and Management, Palwal, affiliated to M.D.U Rohtak (Haryana), India

M.Tech Student, Dept. of Computer Science& Engineering, Shri Ram college of engineering and Management, Palwal, affiliated to M.D.U Rohtak (Haryana), India

ABSTRACT-Regression testing is a trying method which is utilized to approve the altered programming. The Regression test suite is ordinarily substantial and needs a wise technique to pick those experiments which will distinguish most extreme or all flaws at the soonest. Many existing prioritization systems organize the experiments on the premise of code scope as for more seasoned form of the adjusted programming. In this approach, another Genetic Algorithm to organize the Regression test suite is presented that will organize test cases on the premise of finish code scope. The genetic calculation would likewise computerize the procedure of experiment prioritization. The outcomes speaking to the adequacy of calculations are given the assistance of an the Fuzzy improved Genetic approach is used to get the desired results.In GA, the ideal arrangement is looked on the premise of craved masses which further can be supplanted with the new arrangement of masses. The time and induction of experiments (masses) is done by the issues and test cases. The two wellness paradigm picked (issues and cases) are most extreme blame shrouded in least execution time and aggregate code scope. From now on, this wellness capacity will help in choosing appropriate masses for issue and cases using Regression testing vide GA and Fuzzy logic,moving forward, the genetic operations are performed. Right off the strike, hybrid, which recombines two issues.Furthermore, transformation, which arbitrarily swaps the issues.however, the bugs are expelled. At last, the arrangement is checked for enhancement. In the event of fuzzy logic which willimproved the performance, then, the new test case is replicated and regression testing with GA will be levied.

KEYWORDS: Genetic Algorithm,Fuzzy Logic, Regression Testing, Automation, .Matlab.

I. INTRODUCTION

Software Testing:Software testing is a process used to help identify the correctness, completeness and quality of developed computer software. With that in mind, testing can never completely establish the correctness of computer software. Only the process of formal verification can prove that there are no defects.There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following rote procedure. One definition of testing is "the process of questioning a product in order to evaluate it," where the "questions" are things the tester tries to do with the product, and the product answers with its behavior in reaction to the probing of the tester. Although most of the intellectual processes of testing are nearly identical to that of review or inspection, the word testing is connoted to mean the dynamic analysis of the product--

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

putting the product through its paces. The quality of the application can and normally does vary widely from system to system but some of the common quality attributes include reliability, stability, portability, maintainability and usability. Refer to the ISO standard ISO 9126 for a more complete list of attributes and criteria. In general, software engineers distinguish software faults and software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended. Software testing may be viewed as a sub-field of software quality assurance but typically exists independently (and there may be no SQA areas in some companies). In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change the software engineering process itself to reduce the amount of faults that end up in the code or deliver faster. Regardless of the methods used or level of formality involved the desired result of testing is a level of confidence in the software so that the developers are confident that the software has an acceptable defect rate. What constitutes an acceptable defect rate depends on the nature of the software. An arcade video game designed to simulate flying an airplane would presumably have a much higher tolerance for defects than software used to control an actual airliner. A problem with software testing is that the number of defects in a software product can be very large, and the number of configurations of the product larger still. Bugs that occur infrequently are difficult to find in testing. A rule of thumb is that a system that is expected to function without faults for a certain length of time must have already been tested for at least that length of time. This has severe consequences for projects to write long-lived reliable software. A common practice of software testing is that it is performed by an independent group of testers after finishing the software product and before it is shipped to the customer. This practice often results in the testing phase being used as project buffer to compensate for project delays. Another practice is to start software testing at the same moment the project starts and it is a continuous process until the project finishes. Another common practice is for test suites to be developed during technical support escalation procedures. Such tests are then maintained in regression testing suites to ensure that future updates to the software don't repeat any of the known mistakes.

Alpha testing: In software development, testing is usually required before release to the general public. In-house developers often test the software in what is known as 'ALPHA' testing which is often performed under a debugger or with hardware-assisted debugging to catch bugs quickly. It can then be handed over to testing staff for additional inspection in an environment similar to how it was intended to be used. This technique is known as black box testing. This is often known as the second stage of alpha testing.

Beta testing: Following that, limited public tests known as beta-versions are often released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta-versions are made available to the open public to increase the feedback field to a maximal number of future users. Gamma testing is a little-known informal phrase that refers derisively to the release of "buggy" (defect-ridden) products. It is not a term of art among testers, but rather an example of referential humor. Cynics have referred to all software releases as "gamma testing" since defects are found in almost all commercial, commodity and publicly available software eventually. (Some classes of embedded, and highly specialized process control software are tested far more thoroughly and subjected to other forms of rigorous software quality assurance; particularly those that control "life critical" equipment where a failure can result in injury or death).

White-box and black-box testing: In the terminology of testing professionals (software and some hardware) the phrases "white box"and "black box" testing refer to whether the test case developer has access to the source code of the software under test, and whether the testing is done through (simulated) user interfaces or through the application programming interfaces either exposed by (published) or internal to the target. In white box testing the test developer has access to the source code and can write code which links into the libraries which are linked into the target software. This is typical of unit tests, which only test parts of a software system. They ensure that components used in the construction are functional and robust to some degree. In black box testing the test engineer only accesses the software through the same interfaces that the customer or user would, or possibly through remotely controllable, automation

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

interfaces that connect another computer or another process into the target of the test. For example a test harness might push virtual keystrokes and mouse or other pointer operations into a program through any inter-process communications mechanism, with the assurance that these events are routed through the same code paths as real keystrokes and mouse clicks. Where "alpha" and "beta" refer to stages of before release (and also implicitly on the size of the testing community, and the constraints on the testing methods), white box and black box refer to the ways in which the tester accesses the target. Beta testing is generally constrained to black box techniques (though a core of test engineers are likely to continue with white box testingin parallel to the beta tests). Thus the term "beta test" can refer to the stage of the software (closer to release than being "in alpha") or it canrefer to theparticular group and process being done at that stage. So a tester might be continuing to work in white box testing while the software is "in beta" (a stage) but he or she would then not be part of "the beta test" (group/activity) Figure 1 depicts the same.

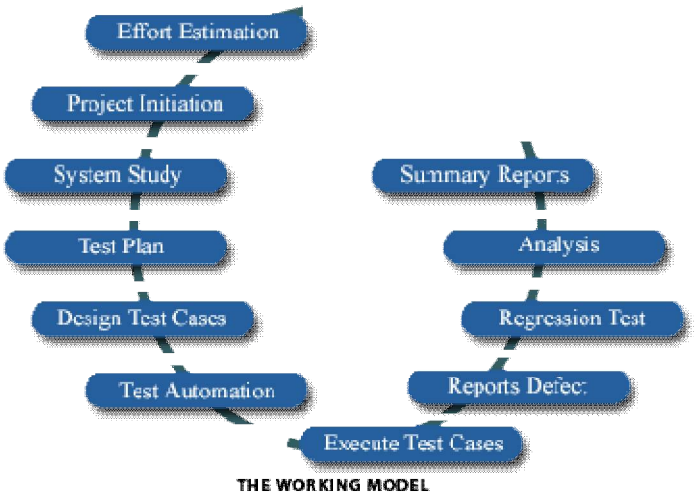

Figure 1. Life Cycle of Testing

Regression testing: Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.Regression testing is an expensive and frequently executed maintenance process used to revalidate modified software. Various problems are associated with regression testing such as regression test selection problem, coverage identification problem, test case execution problem, test casemaintenance problem etc. In test selection problem, appropriate and effective test data is to be selected from the input domain of test data. One more problem may arise, when tester has to select the modified paths from the set of modified path for test case execution i.e. path selection problem. To overcome these problems, this paper presents a combined approach by which the stated problems are resolved in effective manner. By this approach, tester can identify the appropriate paths for test case execution. This approach is used in regression testing to choose an appropriate subset of test cases which suite for a software system, based on theinformation about the modifications made to the system for enhancement.

Genetic Algorithm (GA):A genetic algorithm is a probabilistic search technique that computationally simulates the process of biological evolution. It mimics evolution in nature by repeatedly altering a population of candidate solutions until an optimal solution is found.The GA evolutionary cycle starts with a randomly selected initial population. The changes to the population occur through the processes of selection based on fitness, and alteration using crossover and mutation. The application of selection and alteration leads to a population with a higher proportion of better solutions. The evolutionary cycle continues until an acceptable solution is found in the current generation of population, or some control parameter such as the number of generations is exceeded.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

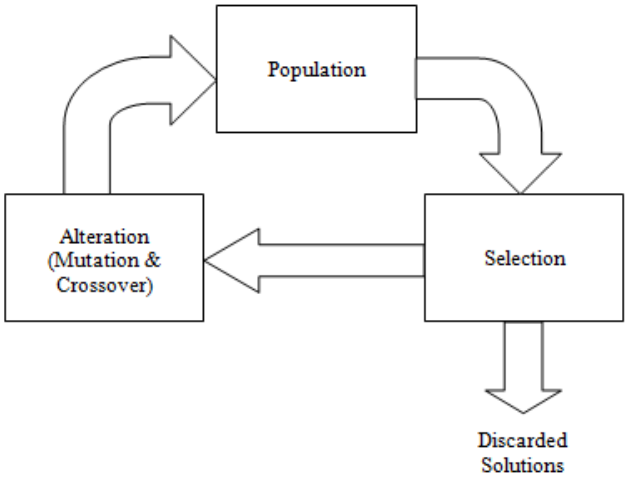

Figure 2 :Genetic algorithm evolutionary cycle.

Fuzzy Logic: Fuzzy Logic is the classical two-valued logic, propositions can take only two truth values from the set $\{0,1\}$. It has been shown using De Morgan's algebras that the propositional logic obtained when the set of truth assignments consists of all values in the unit interval with conjunction, disjunction and negation. De-Morgan algebra is a structure $A = \{A, \vee, \wedge, 0, 1, \neg\}$ such that $\{A, \vee, \wedge, 0, 1, \neg\}$ is a bounded distributive lattice. ¬ is a De Morgan involution such that $\neg(x \wedge y) = \neg x \vee \neg y, \neg\neg x = x$. When the classical set of truth values is extended to include a third value called '*undecided*', the De Morgan algebra generated gives rise to three-valued logic. A structure $([0,1], \wedge, \vee, \neg, 0, 1)$ forms a De Morgan's algebra. Using this as the algebra of truth values, classical fuzzy logic is generated. It has been shown that the equational class of all De-Morgan's algebras can be generated by both three-valued logic and fuzzy logic, hence they are the same. Fuzzy logic extends propositional logic to all values in the interval [0,1]. In such a case, a statement may have a truth value that is neither completely true nor completely false.

Fuzzy logical operations:. The logical operation defined as computation of AND and OR are however not standard. The AND and OR operators in fuzzy logic are generalizations from classical logic and are called t-norms and t-conorms respectively. T-norms and t-conorms are functions from $[0,1]\times[0,1]$ into $[0,1]$ [1]. If $z = T(x, y)$, then x, y, z all belong to the interval $[0,1]$. All t-norms and t-conorms have the properties of commutativity, monotonicity, boundary and associativity. Let *tv* be a truth assignment, then $tv(P\ AND\ Q) = T(tv(P), tv(Q))$ for any t-norm. The basic t-norms are:

1) $T_m(x, y) = \min(x, y)$ (Zahedian intersection)

2) $T_L(x, y) = \max(0, x + y - 1)$ (Bounded difference intersection)

3) $T_p = xy$ (Algebraic product)

As for t-norms, If *C* is a t-conorm then, $tv(P\ OR\ Q) = C(tv(P), tv(Q))$. The basic t-conorms are:

1) $C_m(x, y) = \max(x, y)$ (Standard Union)

2) $C_L(x, y) = \min(1, x + y)$ (Bounded sum)

3) $C_p = x + y - xy$ (Algebraic sum)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

II. LITERATURE REVIEW

Literature review includes various new approaches to generate effective black-box test cases. Literature survey described various new techniques like fuzzy-based age extension, KMPG, permutation technique, HATS algorithm to efficiently generate test cases. Many techniques have their advantages and disadvantages those are concluded briefly through study of various papers.

Last et al. [1] has presented a work on GA based approach for test cases under behavioral testing approach. Author integrated the fuzzy logic with genetic to perform effective problem formulation and to improve the genetic algorithm. Author defined an intelligent approach for test case generation so that the rule based generation of test suit will be obtained. This system is considered as an intelligent system with integrated analysis approach so that the error tracking to the system will be improved to the system. Author defined the complex system analysis so that the error tracking to the system will be performed effectively. The GA based analysis will be performed so that the system to perform the effective testing and performance analysis over the system to generate the effective results.

Tang et al. [27] has defined an effective approach for test case reduction by using regression test selection approach for testing. Author analyzed the set of test cases under requirement analysis and divided the available test suit in two main sets called the current regression test suit and the irrelevant test suit. The constraints are defined by the author to perform the analysis under the effort, failure and protecting code analysis. Author performed the part based analysis over the system and to perform the objective test suit analysis so that the coverage range distribution is performed so that the generation ofeffective test suit will be done. Author defined coverage range analysis so that effective test case generation will be done. Author defined an effective coverage test suit reduction algorithm to solve the problem. Author presented the HATS algorithm to improve the effectiveness of test suit under the fitness rule specification under certain test analysis. Author defined the best case selection approach to get the maximum fitness value. The 14 test suit reduction approach is defined to perform analysis on regression test suit and to obtain the optimum test sequence.

Tao et al. [17] has presented a work on the improvement of test case selection approach for regression testing. Author defined the test case definition in steps so that the improved regression test selection will be performed. Author defined high to low level program analysis. Author defined experimental study on regression test selection approach so that the regression test model will be defined under selection model. Author defined a hierarchical slicing approach for generation of test cases in steps. Author defined the object oriented analysis on test approaches so that the effective test sequence generation will be done.

Chen et al. [10] has presented a prioritization approach to improve the regression testing in case of web services. Author defined an analytical model to obtain the data formation and data flow information analysis. Based on this prior information the weighted graph is constructed and prioritization to test case selection will be done. Author identified the affected element analysis so that the highest weights over the system will be obtained. Author defined the dependency analysis model to improve the priority of the system so that the effective generation of regression test sequence will be done.

Yang and Wu [14] presented an improvement to the web service selection and testing based under the regression test analysis. Author performed the parameter based analysis over the system. The parameters considered by the author are fault analysis, test behavior analysis and the implementation of these factors under the test suit generation and analysis. Author defined the behavior system study to improve the test case selection criteria

Engstrom [5] has defined an improved evaluation approach for test case selection and its improvement under the regression test analysis. Author defined a test selection approach under survey based analysis so that the systematic results will be obtained from the system. Author defined the parametric checks over the system to improve the evaluation procedure and to perform the test cases. 15

Jin and Orso [6] has defined a new approach to perform the behavioral regression testing. Author defined a version system for regression testing to define the dynamic analysis over the system so that the effective test inputs will be drawn from the system. Author defined a systematical model to track the test cases parallel so that the effective system tracking will be done. Author defined the track based analysis over the system so that the improvement gain to the system will be improved.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

III. RESEARCH AND ELABORATION

In this scheme, we propose algorithmic approach is defined at two stages. The first level is defined to identify the test path and second is to identify the optimized path from set of possible paths. The path identification is performed based on the fault and the cost parameters. Once the possible paths are identified, the genetic approach using fuzzy logic is applied to Start and Divide the software program under N number of software modules and
with integrated Test Cases Identify the cost of each test case and assign priority to test cases Identify the possible path over the structure and use it as population set Perform the randomized selection of two test sequence called P1 and P2 Implement the crossover operator to identify next optimized pathImpalement mutation function to modify the generated test case More Path Exist Present the cost effective path as the final result Stopand to identify the effective path. The algorithm opted by the genetic approach is given here under :-

1. Split the software program in N number of software modules
2. Identify the test cases associated with each software modules.
3. Identify the faults over each module and based on it assign the software cost to these software modules
4. Based on the cost analysis prioritize the test cases.
5. Collect the possible test cases over the flow diagram and use it as the population set for genetic process along-with fuzzy logic.
6. For i=1 to MaxIterations

[Repeat steps 7 to 9]

7. Perform the randomized selection under the based effective fitness rule called sequence T1 and T2
8. Perform the fault analysis based crossover using Cyclic approach over these test cases and identify a new Test case
9. Apply the mutation approach to generate a new test sequence
10. Present the obtained test sequence as final result
11. Exit

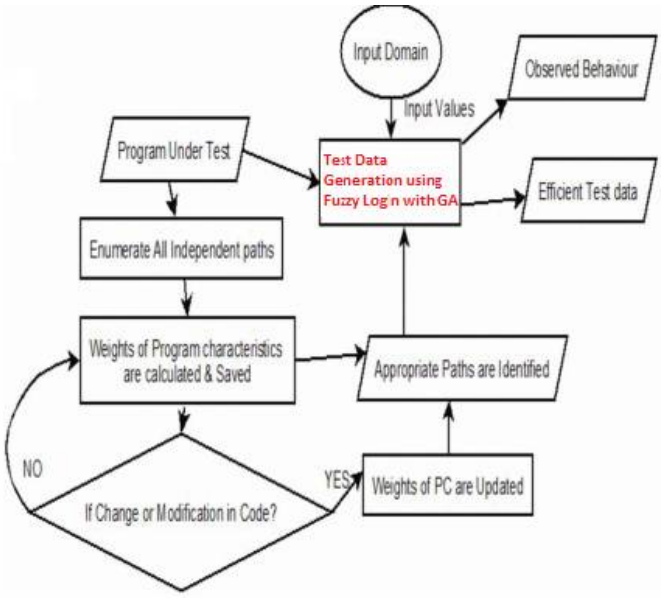

Figure 3: Work Flow Diagram

The first stage of presented regression testing based test path generations approach is to generate the effective sequence path. To generate this path, an effective fuzzy along with GA approach based analysis is performed. The first stage of work is to generate this sequence path under the cost analysis. The prioritization can be here performed under different vectors defined Inthi scheme, an example of test path generations is considered under following vectors defined in table 1:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Parameter	Values
Number of Test Cases	26
Prioritization approach	Fault Base
High Fault Priority	3
Medium Fault	2
Low Fault	1
Number of Levels	9

Table 1.The sequence path generation over the sequence path

Under the scheme generation of test sequence under the test cost analysis. The cost analysis is here defined under the fault analysis and the prioritization approach. The fuzzy effective cost analysis is performed in the form of fitness function. In this section, The sequence generated is analyzed under different vectors. These vectors are defined here under different vectors. The genetic parameters along with regression testing parameters are defined here under:-

Parameters	Values
Total Test Cases	10
Prioritization Approach	Random
Cost Assignment	Prioritization Based
Selection	Random
Optimization Criteria	Fuzzy based Cost Effective
Crossover	Cyclic
Mutation Operator	Random
Number of Iterations	100

Table 2: Test Sequence Generation

As the genetic with fuzzy procedure is applied, the results obtained from the work are shown in table 3.

Result Type	Value
Optimum Test Sequence	3 1 2 4 6 5 8 7 9 10
Initial Cost	5.3
Optimum Cost	4.9
Time Taken	3.21 sec

Table 3 : Results

IV. CONCLUSION

In this present work, an improvement to the existing Fuzzy Logic Optimization Regression Test Suite using GA path estimation approachis defined. The presented work has defined a fuzzy improved genetic approach to identify the effective test sequence under the cost analysis. In this work, the test cost assignment is done based on the prioritization of test sequence. As the regression testing is about the analysis of selective test cases. These selective test cases are identified by high priority test cases. In this work, the analysis is given on different prioritization approaches. Once the priority is assigned to the test cases, the next work is to apply the fuzzy based genetic approach to generate effective regression path. The path generation is here defined under fuzzy effective cost estimation. Fuzzy logic is applied as the constraint to the test cost estimation during the fitness rule. The work is tested under different number of iterations and different methods of prioritization assignment. The results show the random prioritization to the test cases gives more flexibility in test path so that the test sequence can be optimized. Whereas the ascending and descending order based

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

prioritization sequence restrict the test path so that the lesser optimization can be obtained.

REFERENCES

1. Williams L., "Testing Overview and Black-Box Testing Techniques", page no.35-59, 2006.
2. Pressman, Roger S.,"Software engineering: a practitioner's approach" 5th edition,2001.
3. Last M., Eyal S., and Kandel A.,"Effective Black-Box Testing with Genetic Algorithms",2005
4. Singh K., Kumar R., "Optimization of Functional Testing using Genetic Algorithms" ,International Journal of Innovation, Management and Technology, Vol. 1, No. 1, April 2010.
5. Engström E., "Regression Test Selection and Product Line System Testing", Third International Conference on Software Testing, Verification and Validation, 978-0-7695-3990-4/10 $26.00 © 2010 IEEE
6. Jin W., Orso A., "Automated Behavioral Regression Testing", Third International Conference on Software Testing, Verification and Validation 978-0-7695-3990- 4/10 © 2010 IEEE
7. Do H., Mirarab S., "The Effects of Time Constraints on Test Case Prioritization: A Series of Controlled Experiments", IEEE Transaction on Software engineering, vol. 36, no. 5, September, 0098-5589/10/ 2010 IEEE
8. Foo K., Jiang Z., Adams B., "Mining Performance Regression Testing Repositories for Automated Performance Analysis",10th International Conference on Quality Software, 1550-6002/10 © 2010 IEEE
9. Zhang C., et al. "An Improved Regression Test Selection Technique by Clustering Execution Profiles", 10th International Conference on QualitySoftware 1550-6002/10 © 2010 IEEE
10. Kumar A., Tiwari S., Mishra K., "Generation of Efficient Test Data using Path Selection Strategy with Elitist GA in Regression Testing", 978-1-4244-5540-9/10 ©2010 IEEE
11. Li B., et al., "Automatic Test Case Selection and Generation for Regression Testing of Composite Service Based on Extensible BPEL Flow Graph", and60773105, and partially by National High Technology Research and Development, 26th international conference on software maintenance, 978-1-4244-8628-1/10 ©2010 IEEE
12. Kumar M., et al.," Requirements based Test Case Prioritization using Genetic Algorithm", IJCST Vol. 1, Issue 2, December 2010.
13. Rothermel G., Roland H.," Test Case Prioritization: An Empirical Study", International Conference on Software Maintenance, Oxford, UK, September, 1999, IEEE Copyright.
14. Rus I., et al., "Software Dependability Properties: A Survey of Definitions, Measures and Techniques". Fraunhofer Technical Report 03-110, January 2003.
15. Yang B., Wu J., "A Regression Testing Method for Composite Web Service" 978-1-4244-5316-0/10/$26.00 ©2010 IEEE
16. Gu Q., Chen Q.," Optimal Regression Testing based on Selective Coverage of Test Requirements", International Symposium on Parallel and Distributed Processing with Applications, 978-0-7695-4190-7/10 $26.00 © 2010 IEEE
17. Tao C., et al.," An Approach to Regression Test Selection Based on Hierarchical Slicing Technique", 34th Annual IEEE Computer Software and Applications Conference Workshops, 978-0-7695-4105-1/10 $26.00 © 2010 IEEE.Kaur A., Goyal S.," A genetic algorithm for regression test case prioritize using code coverage", International Journal on Computer Science and Engineering (IJCSE), ISSN: 0975-3397 Vol. 3 No. 5 May 2011.
18. Wong W., Agrawal. H.," A Study of Effective Regression Testing in Practice", 8th IEEE International Symposium on Software Reliability Engineering (ISSRE'97), pp 264-274, Albuquerque, NM, November 1997.
19. Kumar A., Tiwari S., "Generation of Efficient Test Data using Path Selection Strategy with Elitist GA in Regression Testing", 978-1-4244-5540-9/10/$26.00©2010 IEEE
20. Li B., Qiu D., "Automatic Test Case Selection and Generation for Regression Testing of Composite Service Based on Extensible BPEL Flow Graph", 26th IEEE International Conference on Software Maintenance 978-1-4244-8675-1/10$26.00 © 2010 IEEE
21. Reddy S., Kumar S.,"An effective approach to regression test optimization technique", Indian Journal of Computer Science and Engineering (IJCSE), ISSN:0976-5166 Vol. 2 No. 5 Oct-Nov 2011.
22. Albertins L., Sampaio A., "A Permutation Technique for Test Case Prioritization in a Black-box Environment", 2nd Brazilian Workshop on Systematic andAutomated Software Testing, page no. 1-10, 2008
23. Parsa S and Khalilian S., Gaurav Duggal, Mrs. Bharti Suri "Understanding Regression Testing Techniques".
24. Gu1 Q., Tang B.," Optimal Regression Testing based on Selective Coverage of Test Requirements", International Symposium on Parallel and DistributedProcessing with Applications,978-1-7695-4152-7/10 $26.00 © 2010 IEEE
25. Zhang C., Chen Z., Zhao Z.," An Improved Regression Test Selection Technique by Clustering Execution Profiles", 10th International Conference on Quality.