Regression Testing Prioritization, Selection and Reduction using Hybrid Criteria

Nitika Sharma
Lovely Professional University
Phagwara
Punjab

Neha Malhotra
Lovely Professional University
Phagwara
Punjab

ABSTRACT
Regression testing is a software testing technique. Testing and validating the part of code are the activity performed within different phases. Tasks of regression testing are: Test Case Prioritization, Test Suite Selection, Test case reduction which give the guarantee that no intended fault is produced while modifying the code. This paper hybrid all the criteria's in different prospective with existing techniques. Selecting and choosing minimum number of test cases according to the result is our major goal. It will give solution to certain unnecessary results found after testing that further seem to be diminished in execution time. In our work we are formulizing the swarm algorithm for hybrid criteria. Hybrid criteria use Rank, Merge and Choice for building the test cases from test suite for minimizing the redundancy. Branch technique is used if one of test cases fails or does not show any result then next option can be used. Swarm algorithms give additional functions for having effective result with less time and effort. Initial seed value for hybrid criteria's is taken randomly. This research will lead to give better efficiency in regression testing using hybrid criteria. Path Coverage deals with the test case selection as it gives all the details of test cases.

General Terms
Event Coverage, Statement Coverage, Execution Time, Fault Detected, Priority, ACO

Keywords
Rank, Merge, Choice, Average Percent Fault Detection

1. INTRODUCTION
Software engineering is an application deals with the development of software i.e. Analysis, Design, Implementation and Maintenance of software in a systematic manner. Development process involves with the need of the client, it first maintain the list and design architecture. Software engineering deal with two types of Engineers: System Engineer: It is the process of designing the entire development. Computer Engineer: Involved software developer with great experience of years.
Regression testing is a maintenance part of the software testing. Maintenance ensures enhancements after detecting, deleting the absolute error. Thus, it guarantee that modified part of the program do not affect the remaining unmodified part. It requires complete effort, attention, consumes time and is cost effective. It effectively tests that new modified code or new features added are compatible with the existing code.

Minimization of time and effort is done by restarting the only selected part of code. Modification checks the compilation. The main goal is to select the minimum number of test cases and also to choose the selected part to test rather than the entire. Researchers worked on various approaches of regression testing i.e. Test case selection, Test suite reduction, Test case prioritization. Criteria's used by these approaches are code coverage modification of code. Code coverage is to check whole program that all the statements, branches, events, faults etc. are covered or not. More the coverage of code lesser is the possibilities to have errors.

Test case prioritization is for ordering test cases, which are done based on the priority to achieve the goal. Test suite reduction minimization is used to reduce the redundancy as the redundancy consists of space coverage. Regression test selection is to select the smaller number of test cases for a large code, in which subpart helps in modifying only selected portion.

Our main focus is to use the hybrid criteria's to increase the efficiency of a particular case. Various algorithms and techniques are used in regression testing in individual approaches but the techniques used in hybrid approaches are hardly implemented in individual approaches. Most of researchers use the hybrid technique with Pareto optimization, HGS algorithms and Pseudo random number generation, swarm and evolutionary algorithms are helping algorithms in this testing.

2. LITERATURE REVIEW
Sreedevi Sampath *et. al.* In his work gave uniform representation of hybrid criteria's. Hybrid criteria's they used is of various approaches Main goal of their research is to give the uniform representation of approaches. They came to know that test cases have lots of coverage. Factors on which they depend are weightage, priority and hybrid. Methodology used by the researcher to show the effectiveness gave the expected results. This paper concludes that it is effective to have "Hybrid criteria" with under a one roof. [14]

Mai Daftedar *et. al.* proposed their paper on regression testing in automatic pseudo random generation. In this test cases are selected according to the recent seed value i.e. regression test selection approach. It tests the current code modification by taking into the seed value. This is the paper which depends upon the GUI systems which are not embedded. Main goal of this paper is to generate accurate test cases and to have best framework to have quality. Algorithm first do initialization by this current state is achieved. In this paper "Rank" formulation is used. Events are created and then triggering process is also done. But the innovating algorithm takes the delay time into count and also does verification. Tools which shows the efficiency is product development kit and hardware simulator tool. Factors on which they proposed the paper is percentage of efficiency, real time response, failure recovery. [18]

Jin chen *et. al*. this paper is based on real time research to calculate when GUI regression is failed then who is responsible or whom to blame. Real world testing Scripting testing is best suited. It checks 197 faults come in the industry in GUI. Faults detected were incorrect fault, configuration, bugs etc. main goal is to fix the bugs by using X Tool and In Design. Result shows out of 197 test cases 2 bugs are found in InDesign and 9 test cases are not found anywhere in the code. Researcher assumes that it is resolved in execution environment. Script, Oracle, Test Tools and Configuration are the causes for false positives. [13]

Gurinder singh *et. al.* ensures the modification of the software effective by increasing the efficiency using GA and ACO algorithms. As we know GA is evolutionary algorithm and ACO is a swarm algorithm. Researcher works upon the previous work and then recommend new proposed technique by using test suite, they select the test case randomly first and then put other test cases in their path. Initialization is by half of the ants and other ants then share the information while new test cases are placed in their paths. This research concludes their report by giving the effective algorithm by reducing the time. [10]

3. RELATED WORK

3.1 Existing System

In the existing system, test case prioritization and selection are used individually with single criteria. Regression testing using hybrid criteria's is not yet done with effective results. As in software systems, software tester has to test number of test cases at a time. Test cases can be more than thousand. So it becomes difficult to test all the test cases within short interval of time period. In previous work no such effective results are found in the field of regression testing by using hybrid criteria's, which can help testing with maximum outcome. Lots of time and efforts are taken by the existing system to test multiple criteria's at a time. As we know regression testing is to test number of test cases by using the criteria's individually, which in turn consumes time and efforts. Using collective approaches with the help of Ant colony we are making the result more effective.
Ant colony optimization is not used to have events, statements and branches within one technique in the existing systems. Examining the relationship of multiple criteria to different techniques on application with different characteristics is still not been done.[14]

3.2 Proposed Approach

In our proposed approach, Ant Colony Optimization is used for regression prioritization and selection but using events, statements, branches and execution time in one technique is our required proposed work. In our work main task is to cover maximum number of test cases. How Ant's work in their area to search for a food is given as follows:

- First requirement is to check number of events covered, which can be selected by Ants after choosing the test cases randomly. It tells what and how the documentation is doing.
- After having events we need to check the statements whether all are covered or not. Statement coverage depends on the rank produced by ordering the criteria's. Each criterion has its own advantages so one could not be affected by through another. After ranking, merging technique is used which helps in having all criteria's in one packet so they can work collectively. Ant Colony optimization helps to merge all the statements.
- Branch criteria are to cover all the conditions under the statements and merge them to have coverage of all intended faults.
- Execution time depends on the requirements and criteria's. How modified code affects the existing code, which will calculate how fast faults are covered under each criteria.
- It is important that value of one criterion do not conflict with other. If statement coverage are more and events are more than faults detected will be more.

Number of terms taken within different criteria's is as followed:

a. Total number of test cases to detect the faults "F" in a test suite are:
{T1, T2, T3, T4, T5}
b. Events covered within the test cases are:
{e1, e2, e3}
c. Statement covered within each test cases of criterion are: {s1, s2, s3, s4, s5}
d. Branches covered in the documentation are:
{b1, b2}
e. Fault detection during the regression testing are:
{f1, f2, f3, f4, f5}

3.3 Strategy

Most of researchers use various criteria's and test those criteria's one after another, but using all in one helps in reducing the work of software tester. They use "Branch" and "all-use" criteria i.e. Branch criteria helps if one fails it will pass to another one. Whereas the all-use takes all features into count, it checks in detail what the previous work is. Event coverage helps to checks how many tasks are performed and how many are failed. Statement coverage helps in having sub test cases in one test case. Maximum statements must be covered in each criterion.

Regression testing first collects the data to test and then made the result. Run the test and then compare the tested outcome with the previous result. If results vary then Fault may exist in modified part which may affect the existing part.

3.4 Research Process

Process of having the Ant Colony Algorithm in our research work defines the designing of our work with the algorithm.
Objective: First task is to have the objective of the work. In our research our main objectives are statement coverage, event coverage, less execution time.
Variables: Variables can be length of events, number of test cases, and calculated number of seconds to detect the fault.
Evaluation: Evaluation of the process deals with the failure criteria.
Comparison: Comparison depends on the execution time and fault covered.

3.5 Methodology

3.5.1 Ant colony using Hybrid Criteria

- Main goal of Ant colony optimization algorithm is to have end result on the basis of achieving minimum number of test cases. Ants select the Path with the deposition of pheromone.
- Methodology is used to have required objectives which can be achieved by selecting test cases in proposed Ant Colony Algorithm. Ant colony

optimization main task is to do Pheromone deposition after knowing the best suited path for food. Ants have different behavior and different types of Ants choose different path for Pheromone deposition and Trail pheromone evaporation.

- Pheromone deposition is too calculating the fitness i.e. Number of test cases is equal to number pheromone deposited by Ants in the path randomly.
- Trail pheromone is one who deposits the pheromone and after giving path they evaporate.
- High probability path is calculated with the help of pheromone deposition.

Ant colony Optimization is a part of swarm algorithms and consists of following reasons to choose in our work

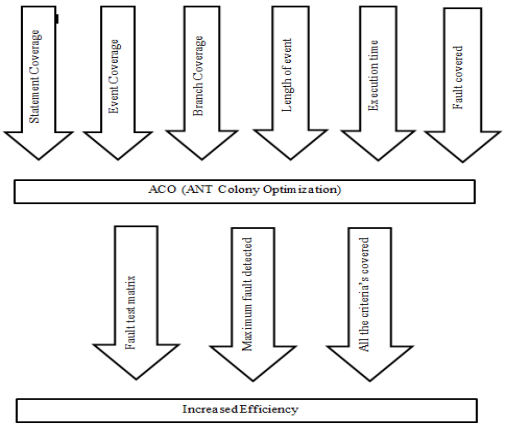

Fig. 1 Working of ACO

Ant colony Optimization is a part of swarm algorithms with consists of following reasons to choose in our work

- Ants have number of control parameters. Maximum control in the algorithm is covered by the ants who deposited higher density pheromone while going from the path randomly.
- Ants in the process cover the path faster and are less time consuming. Whereas in BCO optimization it is hard to calculate.
- ACO Optimization consists of both the terms exploitation and exploration. Pheromone deposition is the one by through ants can communicate whereas in BCO it is hard to understand the communication because of Waggle dance
- In ACO pheromone evaporate after showing the path. But in BCO three kind of Bee's exists who do path coverage employed, onlookers and scout which make the working typical.
- ACO took first priority of test cases randomly then work further according to pheromone deposition.

3.5.2 Metrics used in hybrid criteria

APFD (average percent fault detection) is to calculate the value of priority according to different outputs and software. To measure effectiveness of a test order, we use the average percent of faults detected metric. Although several metrics exist to evaluate prioritized test orders, APFD is the most commonly used metric. For a test suite T with n test cases, if F is a set of m faults detected by T, then let TFm a be test case from the test suite having that particular fault . So APFD metric is given as:

- Values calculated by APFD should be > 0.5 then it will be effective in our work.
- Total number of test cases are 5

$$APFD = 1 - \frac{TF1+TF2\ldots\ldots+TFm}{mn} + \frac{1}{2n}$$

Where, TF1 is the value of that test case which is having first fault.TF2 is the value of that test case which is having second fault and so forth.

m= Modified number of lines, n= Number of test cases.

4. IMPLEMENTATION

Implementation of Ant Colony Optimization (ACO) algorithm process is as follows:

- Input details about test cases in a test suite.
- Run ACO algorithm for all Ants.
- The best path will be chosen from each iteration based on:
    a. Maximum statement coverage
    b. Maximum Events coverage
    c. Maximum Statement coverage
    d. Maximum fault coverage
    e. Minimum Execution Time
- Assigning the control parameters.
  Initial solution:
  Total number of solutions = Total number of Pheromone deposited
- Probability of all the criteria's is calculated. Consider we are on test case i and we have to find probability of test case j.
- Update the Pheromone
- Deposit Pheromone on the best path by using add.
- Reduce Pheromone on the other edges by using evaporate.
- Stop the iteration process if all of the following are covered:
    a. Statement
    b. Events
    c. Branches
    d. Time
    e. Fault
- Check whether all the cases are considered or not.

- Determine the final path based on:
  a. Maximum fault detection
  b. Fault test case matrix
  c. All coverage criteria's

4.1 First Hybrid Criteria

Number of criteria's on which our hybridization depends is Events, Statements, Branches, Execution time and Error covered. Test cases must be selected first to have the total number of inputs within the test suite. Length of the test cases covers the number of events, statements and branches covered with each criterion of the test cases. First we have to initialize the control parameters needed for ant's to prioritize the test cases.

-

4.1.1 Table showing the entire Criteria's Results.

- Each criterion shows its own coverage area individually depends upon coverage faults detected.
- Execution time shows the duration of detecting the fault in time.
- Value of priority is changing with each run.
- Best solution chosen out of each run will depend on the APFD value.

Table 1: Results of Fault Coverage

RUN	Statement	Event	Branches	No. of fault covered	Execution time in sec	Priority	APFD
1	4	2	0	2	5		
	2	2	1	2	1		
	3	2	2	2	8		
	2	1	1	1	1		
	1	2	1	0	9	2,4,1,3,5	0.62
2	4	2	0	2	5		
	2	2	1	2	1		
	3	2	2	2	8		
	2	1	1	1	1		
	1	2	1	0	9	3,2,1,4,5	0.74
3	4	2	0	2	5		
	2	2	1	2	1		
	3	2	2	2	8		
	2	1	1	1	1		
	1	2	1	0	9	1,3,2,5,4	0.72

4.1.2 Calculated Value of APFD Comparison

- Best case chosen is 0.74 i.e. Run 2. Graph shows the relationship of APFD values and our result values to show the efficiency.

-

- Rank arranges which path to follow. Merge initializes values and merges the all the values and detect the fault. Choice selects only one from a set of equally important criteria using a user-supplied selection function. Choice gave the accurate value which we need to select.

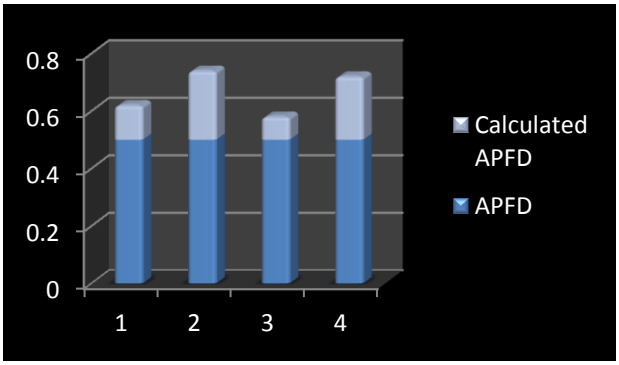

Fig. 2 Calculated Value of APFD

4.2 Second Hybrid Criteria

Different cases have different results; in this case we took different program to test the priority. All the criteria's varies according to the code coverage. According to priority swapping technique is used to set a graph for Ants. Ants select the path on the basis of previous iteration density of finding the fault.

4.2.1 Calculating Input Variables

After having the input files we will calculate the variables as follows.

- Total Ants in the loop: It is to tell total number of Ants who are going to find the solution to a particular problem.
- Total number of test cases used: Total number of test cases is chosen from the test suite.
- Iteration Used: Out of total number of test cases how much iteration is chosen to be used are given.
- Pheromone deposit factor: It tells about the pheromone deposition by Ants.
- Pheromone decrement factor: It gives the value of how much density of pheromone is evaporated during iteration.
- Constant alpha: Alpha value is chosen to be 2 in our work.
- Constant beta: Beta value is chosen to be 1 in our work.

Ant colony optimization show effective result for checking the test case priority. One task is to have the code and took the test cases which can be possible in code. Test case selection and prioritization is essential for maintaining software. Every developer/tester faces this challenge in each organization. In the absence of any effective technique, the random selection of test cases may prevail and the outcome regarding the correction of the program may be illusive and sometimes becomes incorrect. The impact analysis of the changes to the program may further become difficult and time consuming. Hence, an effective technique not only reduces maintenance effort but can also perform the desired impact analysis properly. Moreover, such a technique is now a focus of maintenance activities and helping to preserve the quality of the software. The proposed regression test selection and prioritization technique is efficient in regression testing and thereby reduce process of selecting the priority. Adequate regression testing will also ensure the quality and reliability of the modified software.

Table 2: Results of Fault Coverage APFD value

RUN	Statement	Event	Branches	No. of fault covered	Execution time in sec	Priority	APFD
1	4	2	0	4	5		
	2	3	1	4	1		
	3	2	2	5	8		
	2	2	2	3	1		
	1	2	1	4	9	3,2,1,4,5	0.9
2	4	2	0	4	5		
	2	3	1	4	1		
	3	2	2	5	8		
	2	2	2	3	1		
	1	2	1	4	9	5,2,1,3,4	0.86

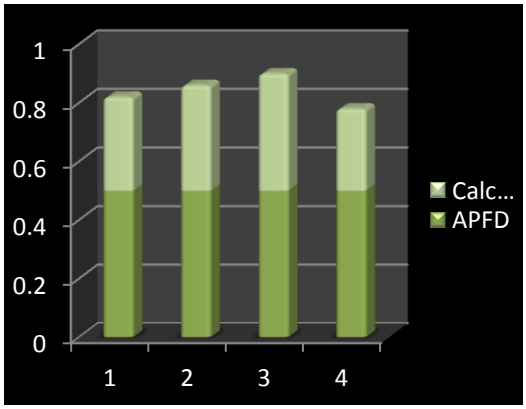

Fig. 3 Graph Showing Calculated Solution

- Best case chosen in this case 0.9 i.e. Run 1.
- Above given table and graph shows APFD value with the same input variables.
- Out of all iteration's, best case chosen by the Ants is a shortest path covered.
- In number of Runs, number of Ants i.e. 4 has different iteration.
- Priority selected in the best solution is 3, 2,1,4,5.

4.2.2 Swapped output of Faults after prioritization is as follows:

Swap technique is used to give the priority in the given matrix. Test case T1, T2, T3, T4, T5 is swapped according to priority calculated. Swapped test cases are given as

T3=T1, T2=T2, T1=T3, T4=T4, T5=T5

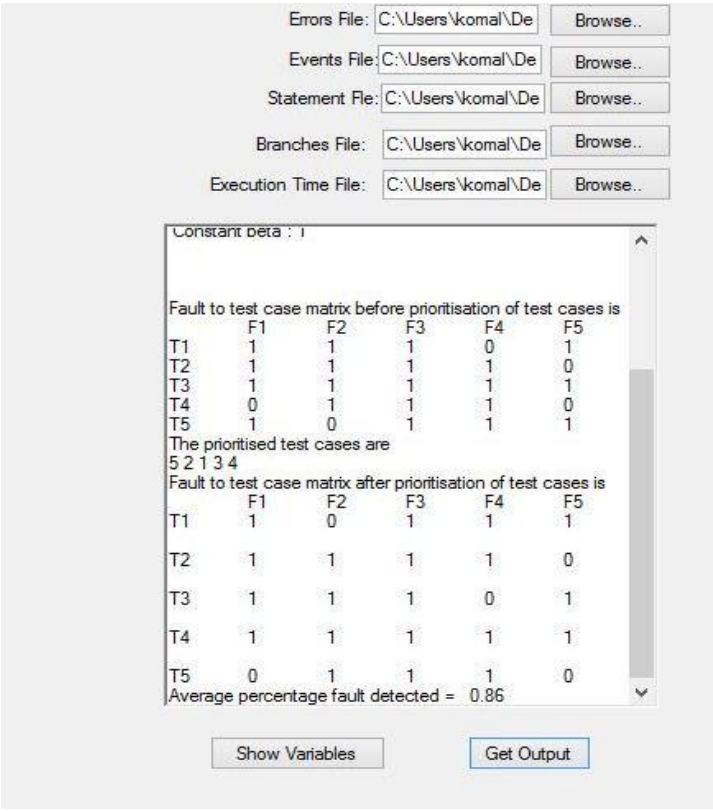

Fig. 4 Swapping of Fault Coverage

The best solution came while resesach was 0.9 i.e. upto 90% of faults are covered within the coverage. In this maximum executive lines are covered and others faults are put in the defer which can be resolved later in the process. Faults are put are defer which are not important to be resolved i.e. not creating any problem in having the final output. It will resolve the fault with low execution time.

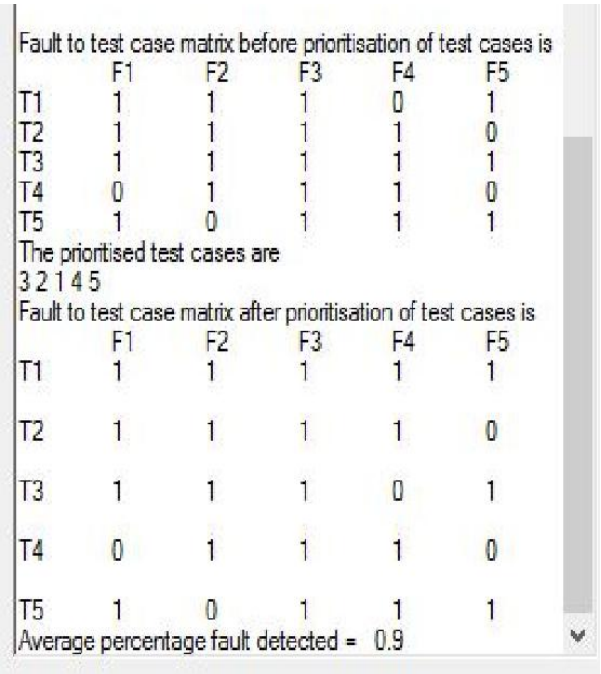

Fig 5. Best solution

4.3 Final Results

Fig. 5 and Fig. 6 shows how ANTS are moving from one destination to another in the search of food with the help of pheromone deposition.

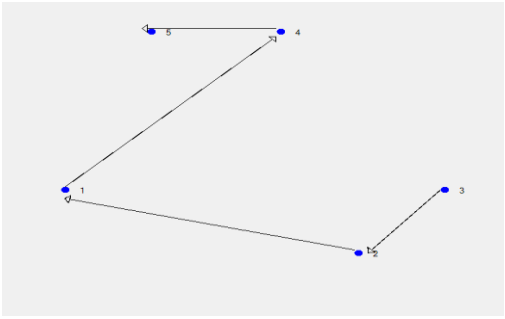

Fig. 6 First Best Priority Graph

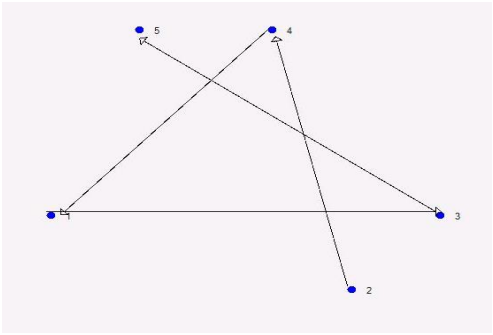

Fig. 7 Second Best Priority Graph

5. CONCLUSION AND FUTURE WORK

The goal of our work was to use the hybrid criteria by using rank, merge and choice techniques to perform the various regression testing activities. We focused on regression test case selection, prioritization and then reduction process. So

our result shows the effective priority of finding the faults by covering all the criteria's together. Our result helps software testers in their practice to test thousands of test cases in an effective manner. Our result is efficient because of following reasons:

- Proposed technique increases the efficiency of hybrid criteria.
- It increased the guarantee that modified part is corrected and do not have intended fault.
- Test cases selected covered all the statements, events, branches within the documents.

Future work is to analyze the process of regression testing using uniform representation of hybrid criteria with the large program to make it more effective.

6. ACKNOWLEDGMENTS

Foremost I would like to thank my advisor mentor Neha Malhotra for continuous support, motivation and patience. Without her guidelines this work cannot be done.

7. REFERENCES

[1] A. K. and M. G. 2012. Multi Objective Test Suite. Minimization Using Quantum Inspired Multi Objective Differential Evolution Algorith . *IEEE, Volume 71*.

[2] Aftab Ali Hader, Shahzad Rafiq and Aamer Nadeem. 2012. Test Suite Optimizaion Using Fuzzy Logic. *IEEE*.

[3] Bharti suri, and S.S. 1924-1932. Implementing Ant Colony Optimization for Test Case Selection and Prioritization. *International Journal On Computer Science and Engineering*.

[4] Bing JIANG, Yangminn MU and Zhihua Zhang. 2010. Research of Optimization Algorithm for Regression Testing Suite. (O. C. Laboratory, & Computer School, Eds.) *IEEE, Volume 2*, pp. 303-306.

[5] Chengying Mao, X. Y. 2012. Generating Test Data for Structural Testing Based on Ant Colony Optimization. *12th International Conference on Quality Software* pp. 98-101. Xi'an, Shaanxi: IEEE.

[6] Daniel Di Nardo, N. A. 2013. Coverage-Based Test Case Prioritisation: An Industrial Case Study. *IEEE*, pp. 302-311. Luembourg: Sixth International Conference on Software Testing, Verification and Validation.

[7] Elie Shaccoour, fadi Zaraket and Wes masri. 2013. Coverage Specification forTtest Case Intent Preservation in Regression Testing. *IEEE*, pp. 392-395.

[8] Geniana Ioana Laiu, O. A. 2012. Automatic Test Data Generation for Software Path Testing using Evolutionary Algorithms. *Third International Conference on Emerging Intelligent Data and Web Technologies*, pp. 1-8. Bucharest.

[9] Ghinwa Baradhi, Nashat Mansour. 1997. A comparative Study of Five Regression Testing Algorithms. *IEEE*, pp. 174-182.

[10] Gurinder singh, Dinesh gupta. 2013. An Intergarted to Test suite Selection Using ACO and Genetic Algorithm. *IJARCSSE*(Issue 6), pp. 1770-1778.

[11] Harrold, M. J. 2009. Reduce,Reuse,Recycle,Recover: Techniques for Improved Regression Testing. *IEEE*.

[12] Hualing zhao, Xiaoxia wu and Hanfeng chen 2010. A New Algorithm in Detecting Changepoint in Linear Regression Model. (C. o. Statistics, Ed.) *IEEE*, pp. 2261-2264.

[13] Jin chen, Mengxiang lin, Kai yu and Bing shao 2012. When The GUI Regression Test Failed, What Should Be Blamed. (B. U. School of Software, Ed.) *IEEE fifth international conference on software testing,verification and validation*, pp. 467-470.

[14] K. Karnavel, J. 2013. Automated Software Testing for Application Maintenance by Using Bee Colony Optimization Algorithms (BCO). *Information Communication and Embedded Systems* pp. 327-330. Chennai: IEEE.

[15] Kropp, M. *Introduction to Software Construction*. University of Applied Sciences Northwestern Switzerland.

[16] Luciano S. de Souza, P. B. 2011. A Multi-Objective Particle Swarm Optimization for Test Case Selection Based on Functional Requirements Coverage and Execution Effort. *23rd IEEE International Conference on Tools with Artificial Intelligence*, pp. 245-252. Boca Raton.

[17] Lulia STEFAN , Liviu MICLEA 2012. The Usage of Contextual Information to Develop the Data Test Vector. *Automated Department*.

[18] Mai Daftedar, Mohamed Shalan 2012. Automated Pseudo-Random Regression Testing for GUI-Centeric Embedded Software. *IEEE*, pp. 293-298.

[19] Md. Junaid arafeen, Hyunsook 2011. Adaptive Regression Testing Strategies: An Empirical Study. *IEEE*, pp. 130-139.

[20] Nirmal Kumar Gupta, M. K. 2013. Improving GA Based Automated Test Data Generation Technique for Object Oriented Software. *3rd IEEE International Advance Computing Conference*, pp. 249-253. Ghaziabad.

[21] Osman Gokalp, A. U. 2012. Improving Performance of ACO Algorithms Using Crossover Mechanism Based on Mean of Pheromone Tables. *International Symposium on Innovations in Intelligent Systems and Applications*, pp. 1-4. Trabzon.

[22] Roykrong Sukkerd, Ivan Beschastnikh, Jochen wuttke, Sai Zhang and Yurily Brun 2013. Understanding the Regression Failures Through Test-Passing Code Change. *IEEE* .

[23] Rui Ding, X. F. 2012. Automatic Generation of Software Test Data Based on Hybrid Particle Swarm Genetic Algorithm. *IEEE* pp. 670-673. Kuala Lumpur: Symposium on Electrical & Electronics Engineering.

[24] Sreedevi Sampath, Renee bryce and Atif M.Memon 2013. A Uniform Representation of Hybriid Criteria for Regression Testing. *IEEE, Volume 39*, pp. 1326-1343.

[25] Toshihiko koju, Shingo Tokada and Norihisa doi 2003. Regression Test Selection Based on Intermediate Code for Virtual Machines. *International Conference on Software Maintenance* pp. 1-10. Department of Information and Computer Science.

[26] Wang Jun, Z. Y. 2011. Test Case Prioritization Technique Based on Genetic Algorithm. *International Conference on Internet Computing and Information Services*, pp. 173 - 175. Hong Kong.

[27] Yi, M. 2012. The Research of Path-Oriented Test Data Generation Based on a Mixed Ant Colony System Algorithm and Genetic Algorithm. *8th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-4. Shanghai.