Test Suite Optimization using Fuzzy Logic

Aftab Ali Haider, Shahzad Rafiq, Aamir Nadeem
Department of Computer Science
Muhammad Ali Jinnah University (MAJU)
Islamabad, Pakistan
aftab775@yahoo.com, shahzad@jinnah.edu.pk, anadeem@jinnah.edu.pk

***Abstract*—Regression Test suite optimization is an effective technique to reduce time and cost of testing. Many researchers had used computational intelligence techniques to enhance the effectiveness of test suit. These approaches optimize test suite for a single objective. Introduction of fuzzy logic with genetic algorithm and swarm optimization may optimize test suite for multi-objective selection criteria. Secondly, human intervention or expert judgment is required to opt for level of testing, technique used and quality aspect to be tested. Fuzzy logic has proved its worth in many other domains like communication, bio informatics, embedded applications, industrial and engineering control and network optimization. We proposed an expert system that finds a trade off among the quality aspects, technique used and level of testing based on objective function defined by the tester, quite similar to human judgment using fuzzy logic based classification. Main focus of our approach is to find a test suite that is optimal for multi-objective regression testing. For proof of our concept, initially we focused on three quality aspects i.e. performance, throughput and code/path coverage.**

***Index Terms*—Regression Testing, computational intelligence, fuzzy logic, test suite optimization.**

INTRODUCTION

Automated Regression Test (ART) is an effective technique to reduce time and cost of testing. Despite its advantages, ART is time consuming and requires exhaustive execution of large test suite. Test suites mostly contain redundant or duplicate test cases. Such test cases are testing-overhead and fail to uncover new errors. More cost and time is spent due to this inefficiency of test suite. Researchers have proposed test suite selection, optimization or minimization techniques to counter this problem. Test suites are minimized, selected or optimized against some criteria. It has been proved that resulting test suite is capable to find almost entire range of bugs for the criteria selected [12][13]. Testing being a major activity of development life cycle in terms of time and cost, optimization and minimization of test suite is given a special consideration to reduce the overall software development time and cost [15].

Test suite selection, optimization or minimization is an active research area since it considerably reduces the time and cost incurred on testing. There are various conventional techniques available that serve the purpose of test suite reduction and optimization and are discussed in [10].

Aside from conventional techniques, computational intelligence like evolutionary algorithms (EA) and artificial neural networks (ANN)) have also been used to reduce the size and optimize the capability of test suite. Many researchers had produced quite impressive results using artificial Neural Networks (ANN) and evolutionary algorithms like Genetic Algorithms (GA), particle swarm optimization (PSO), Ant Colony Optimization (ACO) and Bee Colony Optimization (BCO).

Suri, et al (2011) proposed a quite useful and novel technique that has combined Bee Colony Optimization (BCO) with genetic algorithm [1]. This approach selects test cases with ‘all fault coverage’ in minimum execution time. The focus of the techniques is a single objective of finding “all faults”. Time minimization is the secondary objective. Results of this technique are not repeatable. Optimal test suite may change in second run of the technique. A test suite selected firstly as optimal may not be selected in second attempt as optimal. This factor indicates that approach is not repeatable.

Sehrawat et al (2012) proposed Neuro Genetic Algorithm for optimization of test suite [2]. This approach selects only those test cases for mutation that cover the entire path. This technique is not optimal for time constrained regression testing.

Kaur et al (2011a) proposed a hybrid algorithm of Particle swarm optimization and mutation [3]. Tests with maximum fault coverage and lesser time execution are selected. The main focus of this approach is to find a test suite that finds maximum faults in minimum time.

Kaur et al (2011b) used code coverage as fitness function for test suite optimization [4]. The objective is to optimize test cases that cover all independent paths with minimum number of test cases. This is purely Genetic Algorithm and reproduces test cases through cross over and mutation. This technique is not also suited for time constrained regression testing.

Kaur et al (2011c) proposed an approach that combines Particle Swarm Optimization (PSO) with cross over [5]. In this approach, test cases that cover the entire path or find all faults are selected as global best. The position and velocity of the local best and global best is compared and are replaced accordingly, if local best is not optimal. This approach has two separate objectives that are not considered simultaneously. First is to optimize on coverage and second is to optimize on the basis of faults. This approach is not useful for time constrained regression testing.

Kaur et al (2011d) proposed a fault based approach for test suite optimization [6]. This is purely genetic algorithm that uses mutation and cross over to reproduce the next generation. This approach will not be successful to find test cases that have maximum coverage.

Krishnamoorthi et al (2009) proposed a pure GA based approach for test suite optimization [8]. This technique traverses the code in iterations and uncovers faults for each iteration. This approach has objective of maximized code coverage.

Nachiyappan et al (2010) proposed an approach using pure GA that finds fitness based on coverage and execution time [7]. The objective is to find a test suite with minimum test size.

Subramanian et al (2010) proposed an approach for test suite optimization through mutant gene algorithm [9]. This approach used mutation score to find fitness. Selection was done through ranking the fitness followed by reproduction through one point cross over. The process was iterated till optimization. The objective was to find test suite with greater fitness based on mutation score.

Kaur et al (2011e) used BCO for test suite optimization [19]. The objective was to cover all paths with the intent of time minimization. In another study, Kaur et al (2011f) implemented BCO to detect all faults in minimum time [20]. The primary objective is to uncover all faults whereas time minimization is second objective.

Mala et al (2010) proposed a hybrid algorithm using simple genetic algorithm and bacteriological algorithm [16]. In this work, the authors have not identified the objective function.

Suri et al (2011) used ACO to optimize the path on basis of minimum time [21]. Test suites covering the paths with minimum time were selected for final testing. This work is constrained to single objective of all path coverage with secondary objective of time minimization.

Vivekanandan et al (2012) used ACO to find more faults in minimum time [22]. Objective was to find the faults earlier or "in minimum time span". This approach optimized the testing strategy for existing ecommerce portal system. ACO was used as first step to induce dependencies in the existing approach.

Maia et al (2011) covered three objectives through use of weighted sum approach [23]. However, the problem with this approach is that it uses these attributes to find the next node. This approach will select optimal test path but will not be successful to find an aggregate weightage of each test.

Suri et al (2012) implemented Swarm optimization and GA hybrid approach and compared this approach with ACO [24]. They found that hybrid approach is much better than ACO algorithm in terms of cost and execution time.

After going through these approaches, we found that Computational Intelligence (CI) has been used to optimize the regression test suite. There exists room to explore CI further in optimization of regression testing.

On the basis of above literature review, we conclude that existing approaches attempt to optimize ART suite using EA, PSO, or ANN with single objective criteria. Kumar et al 2011 [10] recommended that in order to optimize the test suite, there is a need to device a mechanism that is able to select test cases based on multiple criterions. There is an ardent need to optimize ART suite using multiple criteria to get the "optimal test suite with multiple objectives". In this work we have proposed an approach that selects test cases for multiple optimization parameters to optimize the test cases [10].

Our objective function is to optimize test cases based on fault detection, execution time and coverage given in Eq. 1. We have used term throughput to describe the ability of a test case to find the faults. A test case that detects six faults will have throughput equal to 6. Secondly, ability of a test case to find the faults earlier is termed as performance. A test case with shorter execution time will be more performing than the others with longer execution times. Coverage can be described in different contexts. Here coverage is either the path covered or the code covered by a test suite. Ultimate goal of optimization is to reduce the size of test cases or uncover those test cases that have the ability to find the faults earlier.

$$\text{Min}\ (ET)\ \wedge \text{Max}(Cov) \wedge \text{Max}(F\ D) \wedge \text{Min}(S) \qquad (1)$$

Where

ET=Execution Time, Cov=Path / Code Coverage,
FT=Fault Detection, S=Test Suite Size

Secondly, we believe that ART should be expert to balance trade off among quality aspect, technique and level of testing as shown in Fig 1. Existing approaches require human intervention for such decision making. The only effective solution is to solve it in continuous domain using technique like fuzzy logic which had proved its worth in many optimization problems from various domains such as communication, bio-informatics, networking, engineering and industrial control systems, and embedded applications. Therefore, we suggested rules for fuzzy inference to fire the triplet for regression testing. We believe that ART should be performed to balance the trade off triangle of Technique, Quality Aspect and Level of Testing which is not possible in discrete domain. We expect that considering these three factors to balance the following trade off triangle testing time will; be reduced and testing application will be artificially intelligent to make an expert judgment to select the test cases suited for multi-objective criteria.

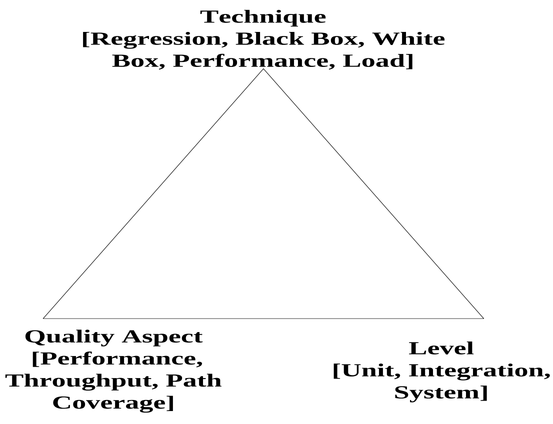

Fig.1 Trade off Triangle for Expert Judgment

Rest of the paper consists of the following sections; In Section II we have described formulation of problem. Section III gives brief overview of solution to the problem. In section IV, we have proposed our algorithm that is an attempt to solve the problem in continuous domain using fuzzy logic. Section V is about the conclusion and future research directions. .

Problem Formulation

Previous work for optimization of ART suite is attempted using single objective e.g. execution time, fault coverage, test suite reduction or code and path coverage. The ultimate goal is to discover a test suite that is smaller in size and is capable to find all error. Generally a test case can be categorized according to its efficacy as best suited, suited or not suited for each objective or selection criteria. Test suite classification for a test suite that covers the entire path is shown in Fig 2. Best suited are those test cases that cover the entire path. When our criteria will be to select test cases that cover entire path, best suited test cases will be selected. Other test cases will be removed and will not become part of next generation. In case we switch over to new criteria of time minimization, test cases selected may or may not comprise those test cases that are previously selected as best suited for path coverage criteria. Same thing will occur for fault detection criteria. The best suited test cases are different for each selection criteria.

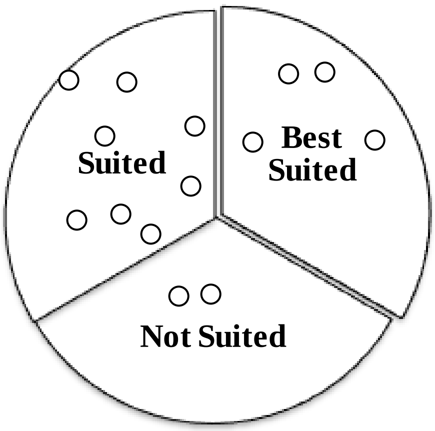

Fig.2 Classification of Test Cases for P/T/C

This shows in-adequacy of the regression test suite optimization techniques as these techniques are optimal for a single criterion and cannot meet the entire objective of regression testing. Basic purpose of ART is to select test cases that are best performing for the entire criterion. Possible cases for three attributes selection criteria are 27.

Case:1 A selected Test Case is best suited for performance, best suited for throughput and best suited for coverage

Case:2 ,A selected test case is best suited for performance, suited for throughput, best suited for coverage

Case:3 ,A selected test case is best suited for performance, suited for throughput, suited for coverage

Case:4 ,A selected test case is best suited for performance, suited for throughput, not suited for coverage

Case:5 ,A selected test case is best suited for performance, best suited for throughput, suited for coverage

Case:6 ,A selected test case is best suited for performance, best suited for through, not suited for coverage

Case:7 ,A selected test case is best suited for performance, not suited for throughput, best suited for coverage

Case:8 ,A selected test case is best suited for performance, not suited for throughput, suited for coverage

Case:9 A selected test case is best suited for performance, not suited for throughput, not suited for coverage

Case:10 A selected test case is suited for performance, best suited for throughput, best suited for coverage

Case:11 A selected test case is suited for performance, best suited for throughput, suited for coverage

Case:12 A selected test case is suited for performance, best suited for throughput, not suited for coverage

Case:13 A selected test case is suited for performance suited for throughput, best suited for coverage

Case:14 A selected test case is suited for performance, suited for throughput, suited for coverage

Case:15 A selected test case is suited for performance, suited for throughput, not suited for coverage

Case:16 A selected test case is suited for performance, not suited for throughput, best suited for coverage

Case:17 A selected test case is suited for performance, not suited for throughput, suited for coverage

Case:18 A selected test case is suited for performance, not suited for throughput, not suited for coverage

Case:19 A selected test case is not suited for performance, best suited for throughput, best suited for coverage

Case:20 A selected test case is not suited for performance, best suited for throughput, suited for coverage

Case:21 A selected test case is not suited for performance, best suited for throughput, not suited for coverage

Case:22 A selected test case is not suited for performance, suited for throughput, best suited for coverage

Case:23 A selected test case is not suited for performance, suited for throughput, suited for coverage

Case:24 A selected test case is not suited for performance, suited for throughput, not suited for coverage

Case:25 A selected test case is not suited for performance not suited for throughput, best suited for coverage

Case:26 A selected test case is not suited for performance, not suited for throughput, suited for coverage

Case:27 A selected test case is not suited for performance, not suited for throughput, not suited for coverage

If fuzzy logic is used to represent case 17, Fig. 3 represents the case.

Fuzzy logic has the capability to find the exact solution using membership of test cases for each test case and then aggregating these. The resultant for all the possible cases for a test suite represented with fuzzy logic is shown in Fig 4 where shaded portion is the group of test suites that are best suited for multi-objective criteria of performance, throughput and code/ path coverage.

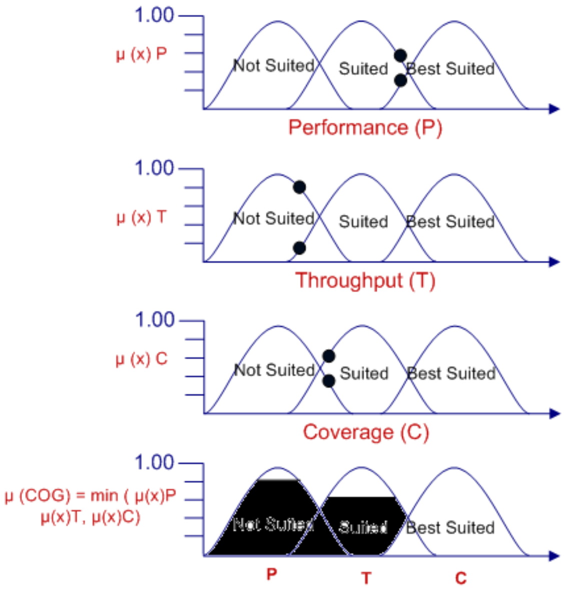

Fig.3 Representation of Case-17 using Centre of Gravity (COG) based fuzzy engine

Traditionally, test cases selected will be those that are best suited against each selection criteria for multi objective function. This will increase the size of test suite. There is a need of mechanism that can optimize the test suite and keep the size of test suite minimum.

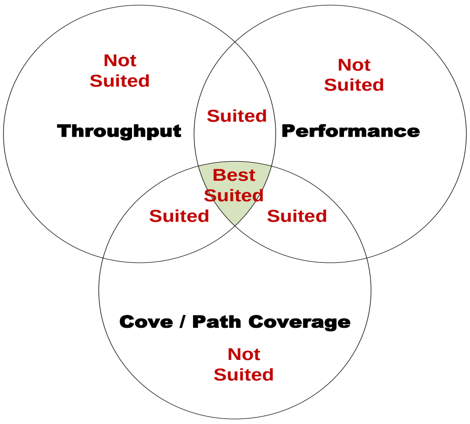

Fig.4 Multi-objective Optimization for P ΛT ΛC

Secondly, existing CI approaches will fail to cater for the entire range of cases mentioned earlier. These approaches can represent Case1 or Case 27. But an optimal test case can be represented with any of these cases. There is a requirement to solve this problem. All the available approaches operate in discrete domain and will not be able to resolve this problem.

PROBLEM SOLUTION

If the same problem is solved in continuous domain with the help of fuzzy logic, the scenario will be quite different. With the help of fuzzy logic, it is possible to compute the membership of each test case in performance, throughput and code/path coverage, as given in Eq. 2. Once the membership of each criterion is computed, then using fuzzy inference engine like COG or Sugeno etc suitability of each test can be determined.

$$\mu_x P + \mu_x T + \mu_x C \quad (2)$$

Where

$P =$ Performance , $T =$ Throughput, $C =$ Coverage

In Fig. 5, we have presented a brief overview of the proposed fuzzy based selection process for regression test suite optimization.

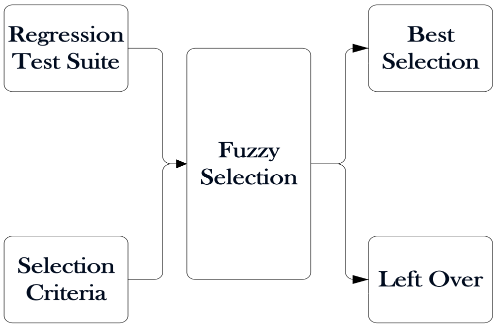

Fig. 5 Fuzzy Selection of Test Cases

This approach will be helpful to select truly optimal test suite. A test case selected may be ‘best suited for throughput’, ‘not suited for performance’ and ‘suited for code/path coverage’. If we have had applied our existing approaches, we might not have been able to select such a test case as optimal. It will be possible only through fuzzy logic to select an optimal test suite with minimum size. This is described in Fig. 3 in problem formulation section, where Case-17 has been represented using fuzzy logic. We can see that Test case is suited for performance, not suited for throughput and suited for coverage. This test case is candidate to be selected as part of optimal test suite. However, existing approaches using CI will not select this test case as optimal test case for multi-objective test criteria.

Our second objective is to develop an expert system that can replace human judgment for methodology and granularity selection for testing. Architecture of the system is given in Fig. 6.

The tester will identify a single input for any of three required inputs. System will identify the rules to be executed for the desired input. A set of rules will be available with the fuzzy inference engine. A typical matrix of fuzzy rules is shown in Table I. At unit level quality aspect to be tested are performance and throughput, as given in Eq. 3, and for system testing criteria selected will be performance, throughput, and coverage, as described in Eq. 4. The objective of the trade off triangle is to select only those test cases for black box testing that meet the objective of performance and throughput in parameters.

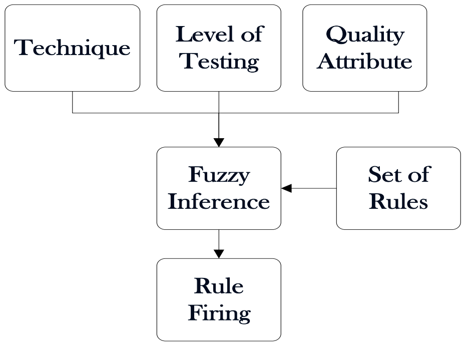

Fig. 6 Expert System to balance Trade off Triangle

This expert system will give an input to the proposed algorithm to select which of the test cases for what kind of quality aspects. Rules are formulated using heuristics like if A and B occur than C will Occur. Suppose If Unit Test Testing occurs than White Box Testing will occur and if Unit Testing and White Box Testing Occur than Performance aspect will be criteria.

TABLE I RULE FIRING FOR EXPERT SYSTEM

Rules	Level of Testing	Testing Technique	Quality Aspect
Rule-1	Unit Testing	White Box Testing	Performance
Rule-2	Unit Testing	White Box Testing	Throughput
Rule-3	Integration Testing	Black Box Testing	Performance
Rule-4	Integration Testing	Black Box Testing	Throughput
Rule-5	System Testing	Regression Testing	Performance
Rule-6	System Testing	Regression Testing	Throughput
Rule-7	System Testing	Regression Testing	Coverage

The rules will be fired in a sequence. Suppose Rule 1 and Rule 2 suggest that criteria for testing should be performance and throughput. These rules can be sued to determine the selection of criteria to select the test cases for the regression test suite.

$$(\text{Rule-1} \wedge \text{Rule-2}) \text{ determine Criteria } (P \wedge T) \quad (3)$$

Similarly

$$(\text{Rule-5} \wedge \text{Rule-6} \wedge \text{Rule-7}) \text{ determine } (P \wedge T \wedge C) \quad (4)$$

PROPOSED ALGORITHM

M. Kumar, A. Sharma, R. Kumar et al has identified multi objective test case optimization categories and parameters. A weighted sum approach has been suggested. The author has foreseen that fuzzy logic can be used in future to implement this approach [10].

Ours work is quite similar to this approach but we have proposed approach with the conception of Computational Intelligence. Secondly, ours approach is purely attributed to regression testing. In this work, we have selected fault oriented objectives as our quality attributes that include coverage, execution time and fault detection. Thirdly ours approach is a mixture of CI techniques and extract of these techniques is ours solution set for regression testing. We have presented the flow chart of our approach in Fig 7. Objective function module is basically the expert system that proposes the quality aspects, granularity of testing and technique to be followed. Based on it, our algorithm selects the test cases using fuzzy logic as discussed earlier.

A brief overview of our proposed algorithm is given as under;

Step:1 Random generation of initial population for regression testing of system under test

Step:2 Tester intervention to identify objective function to balance the trade off triangle

Step:3 Inclusion of respective technique, Level of testing, along with quality attributes.

Step:4 Calculation of execution time, fault detected and code / path coverage for each test case.

Step:5 Implementation of Fuzzy Inference Engine to select population for reproduction

Step:6 Use GA and Swarm Optimization on selected population and keep separately as colonies.

Step:7 Select common test cases between the populations and among the populations.

Step:8 Use Fuzzy Selection e.g. min (max) or union to breed next generation.

Step:9 Compare generation with initial population in terms objective function.

Step:10 If the generation is optimal to be selected otherwise iterate step 6 onward.

Initial population contains randomly generated test cases. Test cases are executed and execution time, faults detected and coverage is calculated for each test case. Membership for each test case is calculated using fuzzy logic and fuzzy selected set is further mutated, crossover and optimized through swarming. These three colonies are further intersected to find the set of common test cases. The test cases that are common in all the three colonies are 'Alpha tests' and those which are common between any of two colonies are termed as 'Omega Tests'. Alpha tests are the martial race of test cases and are potential test cases to optimize the test suite with reduced test size. This concept has been taken from the wolf packs where alpha couple is the superior couple and has the right to reproduce. The next to alpha are Omega couples who can become Alpha after acquiring strength to beat the Alpha couple or after the death of male companion of Alpha couple.

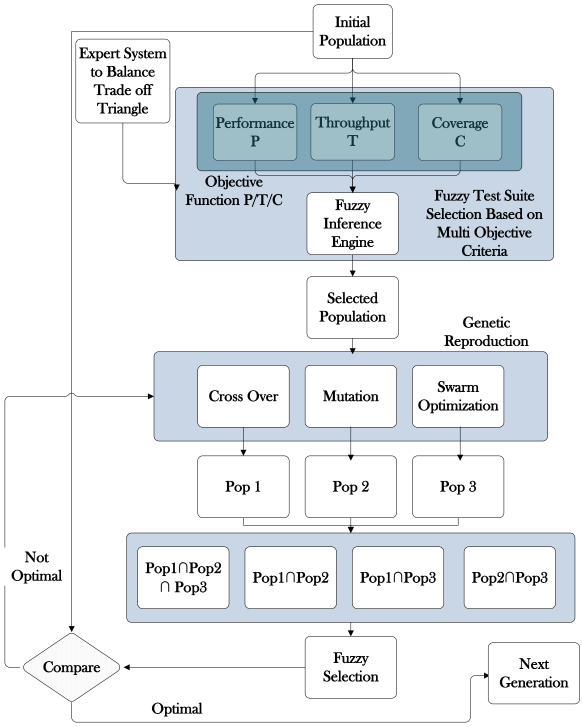

Fig. 7 Flow Chart of proposed algorithm

We translated this concept in our theorem by declaring those test cases as Alpha that are present in all the reproductions and those that are common between any two reproductions are termed as Omega test cases. Our choice is to choose Alpha test cases as optimal test cases and be selected for next generation, but in case Alpha test cases are not present then Omega will get the chance to replace and become Alpha test cases with the permission for next generation.

CONCLUSION AND FUTURE WORK

Fuzzy logic has the prospects to optimize complex problems. We are optimistic that this approach will be successful to produce better results in comparison to other CI techniques. In future we will implement our proposed algorithm and compare it with existing approaches. However, expert system to select the test criteria is a workable proposal and can be useful to generate purpose oriented test cases for system under test (SUT).

REFERENCES

[1] B. Suri, I. Mangal, V. Srivastava "Regression Test Suite Reduction using an Hybrid Technique Based on BCO And Genetic Algorithm" Special Issue of International Journal of Computer Science & Informatics, Vol.- II, Issue-1, 2, ISSN: 2231–5292.

[2] R. Sehrawat, V. Srivastava, I. Mangal "A Neuro-Genetic Algorithm for Selection of Test Suites in Regression testing" International Journal of Computer Application Issues2, Vol-1 (Feb-12), ISSN: 2250-1797

[3] A. Kaur, D. Bhatt "Hybrid Particle Swarm Optimization for Regression Testing", International Journal on Computer Science and Engineering, Vol. 3 No. 5 (May-11), pp1815~1824, ISSN: 0975-3397

[4] A. Kaur, S. Goyal"A Genetic Algorithm for Regression Test Case Coverage Prioritization Using Code Coverage", International Journal on Computer Science and Engineering, Vol. 3 No. 5, (May-11), pp1839~1847, ISSN: 0975-3397.

[5] A. Kaur, D. Bhatt "Particle Swarm Optimization with Cross-Over Operator for Prioritization in Regression Testing", International Journal of Computer Applications (0975 – 8887), Volume 27– No.10, August 2011

[6] A. Kaur, S. Goyal "A Genetic Algorithm for Fault based Regression Test Case Prioritization", International Journal of Computer Applications (0975 – 8887) Volume 32– No.8, October 2011

[7] S.Nachiyappan, A.Vimaladevi, C.B.SelvaLakshmi, "An Evolutionary Algorithm for Regression Test Suite Reduction" Proceedings of the International Conference on Communication and Computational Intelligence- 2010, Kongu Engineering College, Perundurai, Erode, T.N.,India.27 – 29 December,2010.pp.503-508.

[8] R.Krishnamoorthi, S.A.Sahaaya Arul Mary "Regression Test Suite Prioritization using Genetic Algorithms" International Journal of Hybrid Information Technology, Vol.2, No.3, July, 2009.

[9] S. Subramanian and R. Natarajan, "A Tool for Generation and Minimization of Test Suite by Mutant Gene Algorithm" Journal of Computer Science 7 (10): 1581-1589, 2011 ISSN 1549-3636

[10] M. Kumar, A. Sharma, R. Kumar "Towards Multi-Faceted Test Cases Optimization" Journal of Software Engineering and Applications, 2011, 4, 550-557.

[11] Z. Xu K. Gao T. M. Khoshgoftaar "Application of Fuzzy Expert System In Test Case Selection For System Regression Test"

[12] S. Raju, G.V. Uma "Factors Oriented Test Case Prioritization Technique in Regression Testing using Genetic Algorithm" in European Journal of Scientific Research , ISSN 1450-216X Vol.74 No.3 (2012), pp. 389-402

[13] S. Prasad, M. Jain, S. Singh, C.Patvardhan "Regression Optimizer: A Multi Coverage Criteria Test Suite Minimization Technique" in International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-08, Volume 1– No.8, April 2012.

[14] C. Sun "A Constraint-based Test Suite Reduction Method for Conservative Regression Testing" in Journal of Software Vol 6, No 2, Feb 11.

[15] S.Kadry "A New Proposed Technique to Improve Software Regression Testing Cost" in International Journal of Security and Its Applications Vol. 5 No. 3, July, 2011.

[16] D. J. Mala, E. Ruby, V. Mohan "A Hybrid Test Optimization Framework – Coupling Genetic Algorithm with Local Search Technique" In Computing and Informatics, Vol. 29, 2010, 133–164.

[17] S.Yoo, M. Harman "Pareto Efficient Multi-Objective Test Case Selection" in proceedings of ISSTA '07, July 9-12, London, U.K.

[18] S. Yoo, M. Harman "Regression Testing Minimisation, Selection and Prioritisation : A Survey" In proceedings of Softw. Test. Verif. Reliab. 2007; 00:1–7

[19] A. Kaur, S. Goyal "A Bee Colony Optimization Algorithm for Fault Coverage Based Regression Test Suite Prioritization" International Journal of Advanced Science and Technology, Vol. 29, April, 2011

[20] A.Kaur, S. Goyal "A Bee Colony Optimization Algorithm for code Coverage Test Suite Prioritization" International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 4 April 2011

[21] B.Suri, S. Singhal "Implementing Ant Colony Optimization for Test Case Selection and Prioritization" International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 5 May 2011

[22] K.Vivekanandan, G.Keerthi Lakshmi "Improving Regression Testing through Modified Ant olony Algorithm on a Dependency Injected Test Pattern" International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012

[23] C.L.B. Maia, T.N. Ferreira, F. G. Freitas, J. T. Souza "An Ant Colony Based Algorithm for Test Case Prioritization with Precedence"

[24] B.Suri, I. Mangal "Analyzing Test Case Selection using Proposed Hybrid Technique based on BCO and Genetic Algorithm and a Comparison with ACO" International Journal of Advanced Research

in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012