Test Suite Reduction based on knowledge Reuse: An Adaptive Elitism Based Intellect approach (AEBI) using Clustering Technique

Asha N, Prasanna Mani

Abstract: ***Testing is a Process, Not just a Phase. Software testing is the most challenging part of Software Development. Setting up the test environment is an important part of STLC. In the evolutionary environment the test suite size grows constantly as the new updated version of software evolves. Test suite reduction techniques plays intelligent role in deriving the subset of the original test suite that covers the user requirements without affecting the accuracy of the test result. Owing to time and test resource constraints a candidate set can be formed from the original test suite in order to reduce the data processed by the ACO. In this paper, we attempt to introduce a novel approach an Adaptive Elitism Based Intellect approach (AEBI) using clustering technique to reduce the test suite size. This AEBI approach concentrates on single/multiple test objective. The experimental results are presented by applying the AEBI approach on real time application system. We found that the AEBI approach supports effective usage of test resource and performs much better in close proximity with optimal solutions.***

Index Terms: Test Suite Reduction, Test Case Optimization, Knowledge Reuse, Elitism, Clustering.

I. INTRODUCTION

Metaheuristic is a heuristic designed for combinatorial optimization. It is an iterative approach that intelligently combines various elite concepts for discovering and exploiting search space [R. Balamurugan., et al, 2015]. The researcher Fred Glover has coined this word metaheuristic and was popularly used for sensational problems [Glover F., et al, 1986]. This high-level procedure finds the optimal solution based on the phenomenon that occurs in nature. It is magnificently applied in most design problems such as in engineering [Tomoiaga B., et al, 2013], job selection, combinatorial, predictions in data mining and Search Based Software Engineering (SBSE) [M Harman., et al, 2001]. Population-based approach is coined as a popular metaheuristic approach which is chiefly classified into Evolutionary Computation (EC) and Swarm Intelligence (SI) [E. G. Talbi., et al, 2009]. The metaheuristic search techniques are widely applied in Search Based Software Engineering. Many such search algorithms were successfully used all over the life-cycle of software engineering including EC algorithms, Differential Evolution and SI algorithms. The EC or SI algorithms are very straightforward that frames the immaterial problems into tangible form and derives the optimized results with appropriate fitness function. This kind of solution extraction results in an unavoidable high computational ingestions and of more time consuming to maintain the quality of software system. Consequently an enhanced version of solution has to be framed for improving the effectiveness and efficiency of search based algorithms.

In evolutionary biology, the gene segment has a significant impact on the evolutionary process and many theories were proposed by considering them as internal factors. Such factors can be studied by acquiring the knowledge of the application domain. The most fascinating biology concept termed Elitism is a general process of building a new population (subset) by letting the best organism(s) from the current generation to move over to the next. This elitism based practical variant of selection/construction of population is said to be of very effective. In the evolutionary environment, executing the huge number of test cases is of highly time consuming and highly exhaustive. So the elitism concept and the clustering concept can be incorporated to reduce the test cases effectively. The clustering based segmented test cases followed by it the elitism based filtered test cases has a significant impact on the optimization results. Our main intension in this paper is to apply elitism concept for test suite size problem, where the tester's domain knowledge is preserved to be used by the next testing generations. The number of test cases to be fed as input to ACO is reduced by our newly framed AEBI approach, thereby the data processed by ACO is reduced and the ant search is made smart.

II. RELATED WORK

Software testing is a vibrant and lengthy process that demands huge portion of testing resources, tester's effort and cost to be spent on testing the software system. Especially the effort put for performing regression testing on the modified software system is very high. With constant modification in the software system the test suite size increases the size of the test suite which has direct impact on the total test cost. Therefore, the test suite reduction techniques can be used to get a minimal subset of test-suite which is sufficient enough to achieve the given test requirements. There are many EC-based and SI-based algorithms presented for TSR recently including Ant Colony Optimization [Y.

Revised Manuscript Received on April 07, 2019.

Asha N , School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India
Prasanna Mani, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

Singh., et al, 2010] and Particle Swarm Optimization [A. Kaur., et al, 2011].Q. Shen., et al proposed a new Test Suite Reduction model based on Subtraction Operation (TSRSO) in which the redundant requirements and the test cases are reduced by column transformation and row transformation operation of the matrix. The researcher has presented the experimental result that the proposed method generates a minimal test suite than that done by existing methods [Q. Shen., et al, 2017]. Zhang, Ya-nan., et al has presented a Test Suite Reduction method based on Modified Quantum Ant Colony Algorithm to reduce the size of the data processed by Ant Colony Algorithm. The experimental results shows that this new algorithm can obtain a subset of test cases with minimum cost and has good stability too [Zhang., Ya-nan, et al, 2017]. Saifan, Ahmad A., et al used clustering algorithm to reduce the number of test cases, thereby he stated that his test case selection method identify the subset that is capable of satisfying the requirements as well as exposing most of the existing faults. The experimental result shows that the approach was able to obtain 93 unique test cases out of a total of 504 [Saifan, Ahmad A., et al, 2016]. Saifan, Ahmad A., et al in his another paper he has used two data mining approaches to classify the test cases to check for redundancy. The author stated with experimental result that the Naïve based and J48 data mining approach has rendered good support in improving the accuracy while classifying the instances and attributes to identify the redundancy of test cases [Saifan, Ahmad A., et al, 2016]. Huang, Chin-Yu has incorporated Fuzzy Expert System into traditional test suite reduction techniques, his experimental results shows that the Fuzzy-HGS, Fuzzy-GRE and Fuzzy-Greedy algorithms yields the same outcome as traditional algorithms for test suite reduction but in terms of Fault Detection Effectiveness loss (FDE loss) it got decreased by 21% and 5% for Fuzzy-HGS and Fuzzy-GRE [Huang, Chin-Yu., et al, 2016]. Ahmed, Bestoun S., et al has used combinatorial optimization to optimize the test suite and the resultant test suite is filtered based on an adaptive mechanism using mutation testing technique. The author has conducted the empirical study on a software system and proved the effectiveness of the technique [Ahmed, Bestoun S., et al, 2016].From the review made by analysing the related work presented by various authors, it is clear that applying expertise testers' knowledge along with heuristic methods for testing gives better result. In my previous paper [Asha N., et al, 2018] I presented a work by applying knowledge based heuristic method for deriving a smart test segment, in which I used ACO algorithm to find best test case execution path by reusing the testers' intellect knowledge for better result. Also I have framed the Knowledge based approach in my earlier work to show the influence of Knowledge Management System (KMS) in yielding better testing results [Asha N., et al, April 2017], and how Knowledge based concept plays important role in agile approach for better testing process [Asha N., et al, Dec 2018]. In this current paper I have framed a novel Adaptive Elitism Based Intellect (AEBI) approach for optimizing the test case execution; in which am applying the clustering techniques to segment the test cases based on the SO/MO test objectives and filtering the test cases based on the Chromosome Value (CV) to reduce the number of test cases. The resultant test case subset is termed as candidate set which can meet all the test requirements. The reduced representative set is fed as input to the ACO which massively reduces the data processing time of ACO thereby the best test execution path can be found within short time.

III. TEST SUITE REDUCTION APPROACH BASED ON KNOWLEDGE REUSE

The evolution in the software industry drives the tester to test and retest the software product frequently. Growing new updated version of software adds new test cases to test newly added functionalities which increase the test cases in the test suite. The probability of test case redundancy increases due to the more number of test cases satisfying the same user requirements or test objective. This test suite size problem can be rectified by removing the unwanted test case which is termed as test suite reduction. The reduced test case derived after filtering the test cases selectively in a certain subset or candidate set is designated as candidate set policy [1]. The AEBI approach introduced in this paper focus on candidate set selection that helps in reducing the orientation of the ants at initial stage and to improve the efficiency of the ant search. The clustering technique applied in the approach segments the test cases based on the User Requirements and Test Objectives. Given Test Suite TS of n test cases, TS = $\{TC_1, TC_2,\ldots, TC_n\}$, each subset $\{TS_1, TS_2,\ldots, TS_n\}$ of test suite TS is grouped. The approach focus on five main mechanisms: (1) Test Suite Management of a Domain, (2) Segmentation of test cases based on Test objective, (3) Selection of Test segment based on User Single/Multiple requirements, (4) Filtering the test cases to obtain final candidate set based on Elitism concept, (5) deriving best test execution path using ACO. The overview of main components of the AEBI approach is epitomized in Figure. 1.

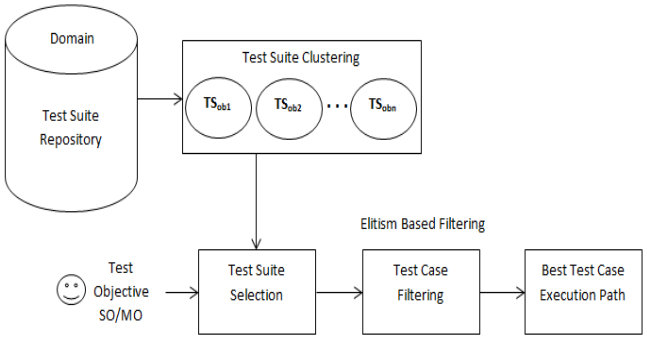

Figure: 1 Block diagram of Adaptive Elitism Based Intellect Approach (AEBI)

A. Test Suite Repository

Software development as become more complex, which implicitly makes the testing process more critical. As per the notification of TechSmith Senior Software Testing Specialist Clint Hoagland: the old testing approach focused on documenting the testing efforts every time and running the tests at the end of each release to ensure error free system.

This testing approach faced major problem like writing and maintaining the test cases every time, which is of time-consuming and of highly expensive too. Maintaining a test repository is a good first step to overcome this problem. By creating a test repository the QA teams can write tests once and store it in the repository to easily retrieve them for future use. Also a common folder structure is used to store overall test cases, only an instance of the test cases is moved into the separate executable folder and the test cases are executed, there by the overwrite of execution result with the latest result is avoided. The test resource in the repository can be leveraged for smooth automation integration that enables the testers to execute the tests as needed without waiting for the manual approval for each interaction. The test repository is revised frequently and actively maintained by eliminating the obsolete tests. By doing so it helps in reducing the overall waste, boost application knowledge, quality and standardizes the testing process.

B. Segmentation of test cases based on Test Objective

In a situation where there are too many test cases to be tested, testing a random subset from them is not defensible, so the tester need to address a systematic solution for this. The better solution is categorizing the likely test cases and selecting a representative test case from each category as shown in figure. 2. By selecting an appropriate representative test case it implicitly guarantees that if the program works properly on executing such test case would also work correctly on all other test cases of that category. An absolute logic called clustering is applied for categorizing the test cases. Doing so helps in reducing the test cases without dropping the confidence in verification. Ample of experience in testing and strong domain knowledge is needed for segmenting the test cases.

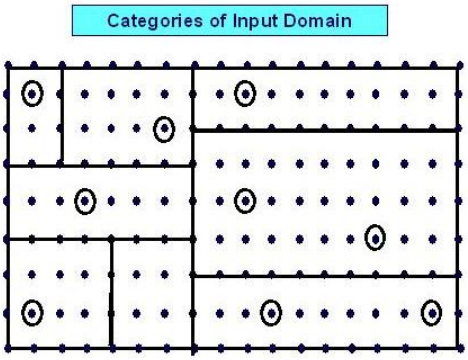

Figure: 2 Categorizing test cases of a domain [source: softwaretestinggenius.com]

The activity of segmenting the population or data points in to number of subgroups in such a way that the data points are more similar within the group when compared to that of the other group. The aim of this part of our approach is to segregate test case groups with similar behaviours to form the clusters. The similar behaviour of the test case is traced based on the test objectives. The unique nature of our approach is; it is feasible to handle both Single Test Objective (STO) and Multiple Test Objectives (MTO). The very standard notion of clusters takes into account groups with less distance among data points, dense areas of the data space, statistical distribution or specified intervals. Here in our approach the clustering technique is used for mining the test cases.

Definition 1: Test Case Segmentation based on Test Objective

Given the Test Objectives Tob = { ob_1,ob_2,…, ob_n}, in general the Test Suite that consist of m number of Test cases under a domain TS = {TC_1, TC_2,…, TC_x}, the test cases are clustered using k nearest classification clustering technique based on the Test Objectives/User Requirements . The Segmented Test Suite (STS) that is clustered in to m different groups, STS = {TS_{ob1}, TS_{ob2},…, TS_{obm}}, each Test Suite of STS comprise of k number of test cases, TS_{ob1} = {TC_1, TC_2,…, TC_k}.

C. Selection of Test segment based on user Single/Multiple requirements

The challenging part of activity in software testing is selecting the appropriate test suite with suitable test cases. In this part of AEBI approach the correct test segment is selected completely based on test objective set by considering the user requirement specifications. The high lighten nature of AEBI is, it works perfectly for both Single Test Objective (STO) and Multiple Test Objective (MTO). If the targeted system is tested without identifying the test segment it affects the overall performance of the STLC. Concurrently it increases testing time and large test resource is needed. In the place of limited test resource some logical technique is need to be applied to reduce the test suite, in this situation AEBI approach supports a lot to derive correct test suite with adequate test cases init.

D. Filtering the test cases to obtain final candidate set based on Elitism concept

This part in AEBI approach is of highly prominent part as the testers' intellectual knowledge is applied based on the term elitism to extract the exact representative set for the test case execution. In Genetic Algorithm (GA), the notion behind the elitism is to sustain the best solution framed by the proficiency testers' from the generation; it increases the performance of the approach. A practical variant of the general process of constructing a new population is to allow the best organism(s) from the current generation to carry over to the next, unaltered. This strategy is termed as elitist selection and it guarantees that the solution quality obtained by the GA does not decrease from one generation to the next. The elite chromosomes are the best in fitness compared to other members, so the test cases that are good with high fitness value are given first priority to form the representative set and the rest are considered according to the classical way. Elitism rapidly increases the performance of the GA, because it prevents losing the best found solution.

Definition 2: Elitism based filtering (EBF) Given a permutation of test cases in a Segmented Test Suite, the elitism based filtered test case segment is termed as the candidate set which is capable of achieving the optimum of test objective quickly. In this paper we consider a gene segment which is termed as Representative Set derived from the pool of Test Suite Repository for effective test case execution.

E. Deriving best test execution path using ACO

In the beginning, the ants use pseudo-random rules for path selection by considering all the combination of node paths associated with the initial node. Therefore for n number of nodes the node time complexity to be handled is proportional to n2. Its waste of considering all the test cases as it increases the ants' search time. So the Representative Set with limited subset of test case can be used to reduce the time of ants search and to improve the efficiency of the whole iteration. The Representative Set covers all the test requirements. The AEBI approach uses the Clustering and Elitism based filtering techniques to reduce the initial set of test cases which is fed as input to the ACO.Ant colony optimization (ACO) is a Meta heuristic swarm intelligence technique proposed by Marco Dorigo in 1992, it is good in tactically finding optimal solutions for the critical problems.

F. Processing of ant colony optimization

In ACO, the set of artificial ants called as software agents search for best solution to a given optimization problem. The basic functions that are applied to compute best path on a weighted graph are defined below.

G. Node Selection

In ant system the artificial ants concurrently constructs the traversal path. At each construction the ant applies random proportional rule to decide which node to visit next, the probability of ant k at node i chooses to go to node j is

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}}, \quad \text{if } j \in \mathcal{N}_i^k,$$

Where α and β are two parameters that determines the relative influence of pheromone trail and heuristic information. τ_{ij}^{α} denotes the pheromone value of edge i to j and η_{ij}^{β} represents the heuristic information of ant. N_i^k is the feasible neighbourhood of ant k which is not visited yet.

H. Update of pheromone trails based on elitist ant system

The pheromone trails are updated soon after the construction of tour path by each and every ant. First the pheromone value is lowered on all paths applying a constant factor. The pheromone evaporation is implemented by

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \quad \forall (i,j) \in L,$$

where 0 < ρ < 1 is the pheromone evaporation rate, the parameter ρ is used to avoid the unlimited accumulation of the pheromone trails. The Elitist strategy for Ant System (EAS) which was introduced by Dorigo (1992) is an additional concept used to strengthen the arcs belonging to the best path found at the start of the algorithm. This path is denoted as T_{bs} (best-so-far tour) and it is a daemon action of the ACO. After evaporation, the pheromone is deposited on the arcs traversed by ants. The pheromone deposit is implemented by

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} + e\Delta\tau_{ij}^{bs},$$

Where ij^k and ij^{bs} is the amount of pheromone deposited by ant k on the arcs it has visited. The ij^{bs} is defined as

$$\Delta\tau_{ij}^{bs} = \begin{cases} 1/C^{bs}, & \text{if arc } (i,j) \text{ belongs to } T^{bs}; \\ 0, & \text{otherwise.} \end{cases}$$

I. AEBI Approach Algorithm

Algorithm 1denotes the detailed procedure of the novel AEBI approach which consist of the steps that is used for segmenting the test case using Clustering technique and the Elitism concept that is intelligently applied for filtering the test cases based on the Chromosome Value of each test case and the intellectual domain knowledge of expertise testers'.

Algorithm 1 An Adaptive Elitism Based Intellect Appraoch (AEBI) Algorithm

// Algorithm for Test Case Segmentation

Let Tob be Total objectives Tob = { ob_1,ob_2,…, ob_n} // each Test Case satisfy any of the given

objectives

Let STS be set of Segmented Test Suite STS = {TS_{ob1}, TS_{ob2},…, TS_{obm}}

Let TS be Test Suite that consist of Test Cases TS = {TC_1, TC_2,…, TC_x}

Let CV be the Chromosome Value of each Test Cases CV = {TC_{cv1}, TC_{cv2},…, TC_{cvy}}

Let UR be User Requirements/ Objectives UR = {ob_1, ob_2,…, ob_z}

TSClustering (UR)

Output STS = {TS_{ob1}, TS_{ob2},…, TS_{obm}}

1 Set i = 0
2 For each $TS_{ob(i)}$ in UR, Repeat steps 3 to 9
3 Initialize $TS_{UR(i)}$ = { }
4 set j = 0
5 For each TC_j in TS, Repeat steps 6 to 8
6 If TC_j contains UR_i then
7 Add TC_j to $TS_{UR(i)}$
End if
8 Next j
9 Next i
10 Return STS

// Algorithm to filter Test Case with Highest Chromosome Value

GenerateHCV (TS_{obx}, K)

Output HCV = {TC_1, TC_2,…, TC_k}

Build Binary Search tree for TS_{obx}

Retrieve last k nodes i.e., nodes with maximum CV value and add to HCV_{obx}

return HCV_{obx}

// Algorithm to generate Nodal Graph NG

// Let THCV be Total HCV set for all given objectives THCV = {HCV_{ob1},…, HCV_{obn} }

GenerateNG (THCV)

1 Let G = { }
2 Add source-node to G
3 Set i = 0
4 For each HCV_{obi} in THCV, Repeat steps 5 to 11
5 Set j = 0
6 For each TC_j in HCV_{obi} , Repeat steps 7 to 10
7 If G contains only source-node then
8 Add TC_j to G
Else
9 Add TC_j to G and create edges from all the existing nodes to TC_j except source-node
End if
10 Next j
11 Next i
12 Return G

// AEBI Approach

1 Read UR
2 STS = TSClustering (UR)
3 Set i = 0
4 For each TS_{obi} in STS, Repeat steps 5 to 6
5 HCV_{obi} = GenerateHCV (TS_{obi})
6 Next i
7 G = GenerateNG (HCV)
8 BTEP = Ant (G)
9 Stop

// Algorithm for Ant Search for Best Test Execution Path

Ant (G)

1 Start at source-node
2 BTEP = { }
3 Repeat steps 3 to 5 till end node is reached
4 check the CV for each node TC_x in the next level
5 Add the node TC_x to BTEP with highest CV
6 Return BTEP

IV. Experimental Result and Discussion

This section presents the experimental results of the proposed AEBI approach and it focus on improving the data processing of the ACO algorithm. The experimental study was made with the set of Test Cases of Wearable Embedded software (WES) application system. The sample test cases of the WES were considered to show the working prototype of AEBI approach. Table. 1 presents the set of test cases accounted. The experiment was conducted by extracting five Test Objectives/ User Requirements: ob1: Effective Execution Time, ob2: Cost, ob3: Statement Coverage, ob4: Fault Detection Rate, ob5: Efficiency. For the subjected WES project the Test Suite of size 100 was extracted from the Test Suite Repository.

Test Scenario	Test case	Description
Wearability	TC1	The miniaturization of integrated circuits and batteries will help designers to improve sensor wearability and the user's level of comfort.
Timing Constraints	TC2	The time-frame between the sampling of sensor inputs and triggering an alarming situation should have strict timing constraints.
Energy Constraints	TC3	During the test time to estimate the energy consumption of software and to check whether the software satisfies certain energy constraints.
Reliability	TC4	Performing feature extraction on the sensor, and transfer only information about an event instead of transferring raw data from the sensor. This reduces communication requirements and increases reliability.
Security	TC5	Wireless sensors must meet privacy requirements mandated by the law for all devices and must guarantee data integrity. Though key establishment, authentication, and data integrity are achieved.
Interoperability	TC6	The interoperability of wireless sensors will promote vendor competition and eventually result in more affordable systems.
Physical environment	TC7	The testing of software under all operating conditions of the physical environment. The testing process should ensure that the respective software acts appropriately in different environmental conditions.
Simulation	TC8	Response time for processing the events that occur simultaneously.
Compatibility/conversion testing	TC9	Products that are designed to replace an existing obsolete product. This testing aims to find compatibility defects between the tested product and the obsolete product.
Logic Analyser	TC10	Testing for occurrence of signals in a proper sequence.

Table: 1 Test Cases of WES application system

Step1: The extracted test cases and the corresponding user requirement met by each test case, the preserved chromosome value of each test node to be used by next new population is shown in the table. 2.

Test Cases (TC)	Ob_1	Ob_2	Ob_3	Ob_4	Ob_5
TC_1	*		*		
TC_2		*			*
TC_3				*	
TC_4	*	*			
TC_5			*	*	*
TC_6		*	*		
TC_7				*	*
TC_8	*				
TC_9		*		*	
TC_{10}	*		*		*

(a)

Test Cases (TC)	Chromosome Value (CV)
TC_1	0.7216
TC_2	2.342
TC_3	0.598
TC_4	0.928
TC_5	3.467
TC_6	1.767
TC_7	3.918
TC_8	2.128
TC_9	1.928
TC_{10}	0.825

(b)

Step2: The extracted test cases are segmented based on the finite set of test objectives using the k-nearest classification clustering technique and from the Segmented Test Suite (STS) the Test Suite are selected according to the User Requirements, Table. 3. Based on this selection the test cases are selectively reduced to large extent.

Segmented Test Cases based on Test Objectives	
TS_{ob1}	$TC_1, TC_4, TC_7, TC_8, TC_{10}$
TS_{ob2}	TC_2, TC_4, TC_6, TC_9
TS_{ob3}	$TC_1, TC_3, TC_6, TC_{10}$
TS_{ob4}	TC_2, TC_3, TC_7, TC_9
TS_{ob5}	$TC_1, TC_3, TC_7, TC_{10}$

(a)

User Requirements: $ob_1, ob_2, 0b_5$	
TS_{ob1}	$TC_1, TC_4, TC_7, TC_8, TC_{10}$
TS_{ob2}	TC_2, TC_4, TC_6, TC_9
TS_{ob5}	$TC_1, TC_3, TC_7, TC_{10}$

(b)

Table: 3 (a) Segmented Test Cases; (b) Selected Test Suites

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Step3: Each test case node holds its Chromosome Value (CV) by following the Elitism concept. The selected test cases are held in the Binary Search Tree structure which makes easy to search for the nodes with Highest Chromosome Value (HCV). From the Search tree the top k nodes are filtered to form the candidate set based on candidate set policy Figure. 4. The filtered/reduced representative set test cases are fed to the ACO with which the data processing is improved intelligently and the searching iteration of ant is reduced.

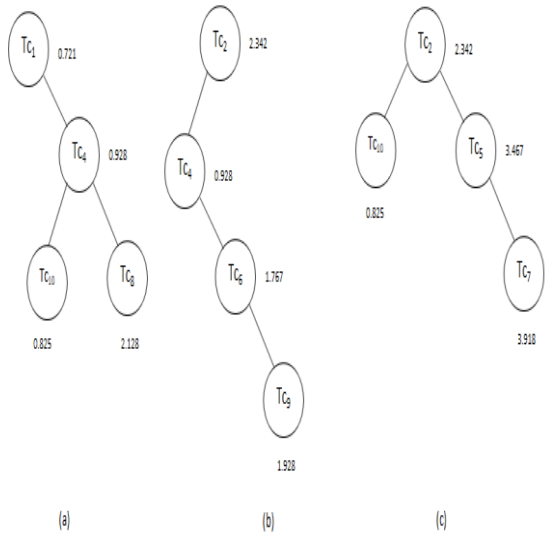

Figure: 2 Filtered test cases sorted using Binary Search Tree, (a) objective ob_1 is met, (b) objective ob_2 is met, (c) objective ob_5 is met

The experimental result of AEBI approach is presented in Figure.3, the intermediate result of segmented test suite is extracted and displayed for further treatment. The experiment was implemented and tested in Windows 10 platform with a system configuration of Intel 64 bit, core i5 processor, 500 GB memory and 8 GB RAM.

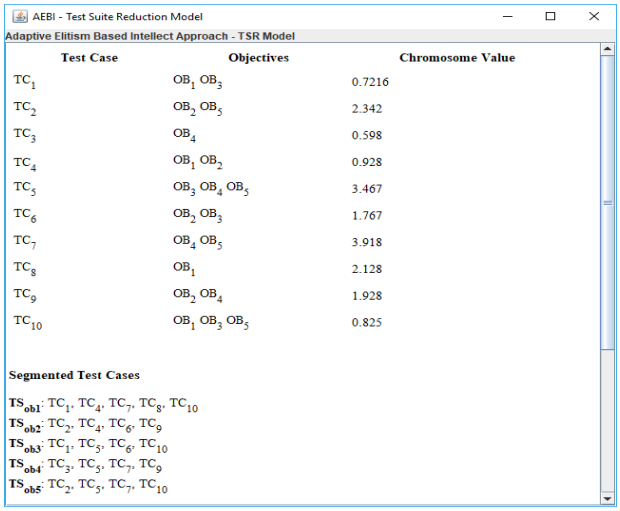

Figure: 3 The intermediate result of Segmented Test Suite (STS)

The test case Nodal graph is created with the representative set of test cases and the ant search starts from the source-node. Based on the Elitism concept, with the previous intellect knowledge of each node Chromosome Value the ant sets the best path for test case execution. The graph is constructed with the obtained intermediate result of filtered test cases with HCV. The ant selects the next node based on the top most HCV and the path is set for the best test case execution sequence. According to the graph depicted in Figure. 4, the path sequence Tc8 - Tc7 - Tc9 is selected as best path within one single iteration.

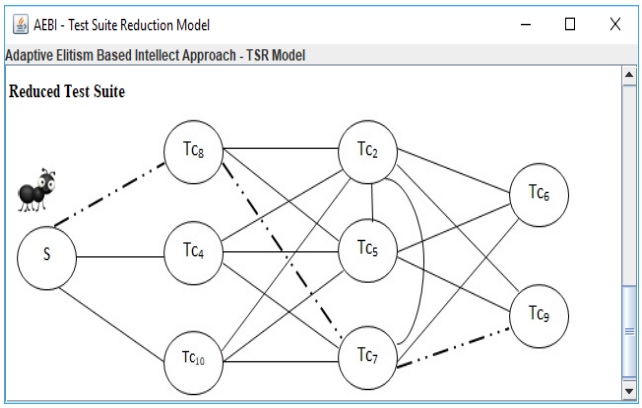

Figure: 4 The Best Test Execution Path (BTEP)

The comparative result of AEBI approach is presented in Figure. 5. The graph is constructed with respect to the number of Test Cases executed and the User Requirements. The proposed approach has produced the optimal solution for Test Suite Reduction. In the graph the very first comparison bar is made by considering the Test Objectives $ob_1+ob_2+ob_5$, where 70% of efficiency is achieved when compared with Total Test Suite, in the second bar comparison 80% of the efficiency is achieved when considering ob_1+ob_4 and in the third bar comparison 90% of the efficiency is achieved when considering ob_2+ob_5.

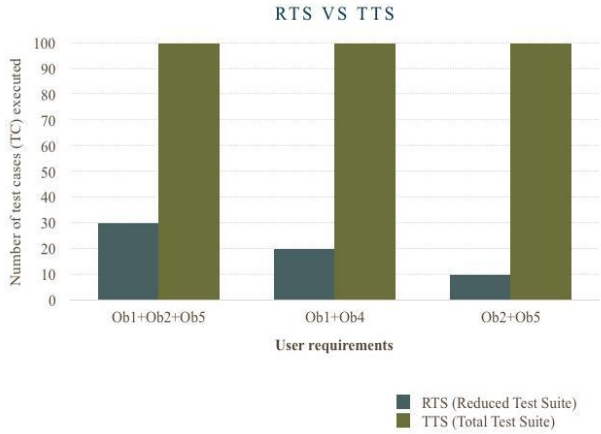

Figure: 5 The comparative graph of experimental results

V. CONCLUSION

Testing process is a highly challenging practice to ensure the quality of a System Under Test . The size of the test suite increases constantly as the evolution takes place in software industry. The gradual increase in the Test Suite Size badly needs high testing resources, budget and time. Test Suite Reduction is represented as a proficient approach to solve the Test Suite Size problem. The metaheuristic algorithm can be applied to this problem to bring better solution. The most emerging Elitism concept is incorporated; where Elitism is related with memory: "remembering the best solution found". With this intellectual idea the candidate set is formed from the original test suite to reduce the data processed by the ACO thereby 100% possibility is achieved to conduct test within the limited test resources. In this paper we have framed an AEBI approach that handle Test Suite Reduction problem considering both Single and Multiple Test Objective. The Elitism concept applied in this approach supports in deriving the Test Case Segment (called as gene segment) from the test repository, which is termed as Representative Set/Candidate Set. The consistent candidate set reduces the data processing time of ACO and the Elitism concept helps in finding the Best Test Execution Path (BTEP) in a single iteration of ant search. The empirical study is made on an application system and the intermediate results are presented clearly in section 4. These results provide the evidence that the proposed AEBI approach has skilfully solved the Test Suite Reduction problem and the candidate set derived has assured 100% improvement in efficiency and the effectiveness of the testing approach. The data processing of ACO is reduced to a greater extent. The search of best path is achieved in a single iteration. Furthermore, in future the paper concentrates are comparing the proposed technique with some significant and state-of-the-art inspired technique and the comparative study result can be presented for performance evaluation.

REFERENCES

1. R. Balamurugan; A.M. Natarajan; K. Premalatha (2015). "Stellar-Mass Black Hole Optimization for Biclustering Microarray Gene Expression Data". Applied Artificial Intelligence an International Journal. 29 (4): 353–381.
2. Glover F., (1986). Future paths for integer programming and links to artificial intelligence, Computers and Operations Research, 13, 533–549 (1986).
3. Tomoiagă B, Chindriş M, Sumper A, Sudria-Andreu A, Villafafila-Robles R. Pareto Optimal Reconfiguration of Power Distribution Systems Using a Genetic Algorithm Based on NSGA-II. Energies. 2013; 6(3):1439–1455.
4. M. Harman and B. F. Jones, "Search-based software engineering," Inf. Softw. Technol., vol. 43, no. 14, pp. 833–839, 2001.
5. E.G. Talbi, Metaheuristics: From Design to Implementation. Hoboken, NJ, USA: Wiley, 2009, vol. 74.
6. Y. Singh, A. Kaur, and B. Suri, "Test case prioritization using ant colony optimization," ACM SIGSOFT Softw. Eng. Notes, vol. 35, no. 4, pp. 1–7, 2010.
7. A. Kaur and D. Bhatt, "Hybrid particle swarm optimization for regression testing," Int. J. Comput. Sci. Eng., vol. 3, no. 5, pp. 1815–1824, 2011.
8. Shen, Qing, Yunliang Jiang, and Jungang Lou. "A new test suite reduction method for wearable embedded software." Computers & Electrical Engineering 61 (2017): 116-125.
9. Zhang, Ya-nan, et al. "A Test Suite Reduction Method Based on Novel Quantum Ant Colony Algorithm." 2017 4th International Conference on Information Science and Control Engineering (ICISCE). IEEE, 2017.
10. Saifan, Ahmad A., et al. "Test case reduction using data mining technique." *International Journal of Software Innovation (IJSI)* 4.4 (2016): 56-70.
11. Saifan, Ahmad A. "Test Case Reduction Using Data Mining Classifier Techniques." JSW 11.7 (2016): 656-663.
12. Huang, Chin-Yu, Chung-Sheng Chen, and Chia-En Lai. "Evaluation and analysis of incorporating Fuzzy Expert System approach into test suite reduction." *Information and Software Technology* 79 (2016): 79-105.
13. Ahmed, Bestoun S. "Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing." Engineering Science and Technology, an International Journal 19.2 (2016): 737-753.
14. N. Asha, Prasanna Mani, " Applying Knowledge Based Heuristic Method for Efficient Test Automation", Journal of Advanced Research in Dynamical & Control Systems, Vol. 10, 12-Special Issue, 2018.
15. N. Asha, Prasanna Mani, "Knowledge-based Acceptance Test driven agile Approach for Quality Software Development", International Journal of Resent Technology and Engineering, Volume-7 Issue-4S2, December 2018.
16. N. Asha, Prasanna Mani, "Customized Services to Generate Test Suits for Testing Custom Software Application System based on Knowledge Reuse", Journal of Advanced Research in Dynamical & Control Systems, 01-Special Issue, April 2017.

AUTHORS PROFILE

ASHA. N working as Assistant Professor (Senior) in the School of Information Technology and Engineering, VIT University, Vellore District, Tamil Nadu, India. She has completed her Master Degree M.E in Computer Science and Engineering. She is pursuing Ph.D in the area of software testing. She is a life member of Computer Society of India (CSI). Her current research includes Software Engineering, Software Testing, Information Security, IOT and Big Data.

Dr. Prasanna Mani is an Associate Professor in the School of Information Technology and Engineering, VIT University at Vellore, Tamil Nadu, India. He has completed his Ph.D in the area of software testing. His research area includes Software Testing, IOT, Big Data, Artificial Intelligence, Business Informatics and Information System. He is associated with many professional activities and has published many technical papers in various international journals and conferences.