

A Parallel Early Binding Recursive Ant Colony Optimization (PEB-RAC) Approach for Generating Optimized Auto Test Cases from programming Inputs

Yogita Dubey
M.Tech VI Semester
Department of CS
BUIIT, BU, Bhopal, India

Divakar Singh
Head of Department
Department of CS
BUIIT, BU, Bhopal, India

Anju Singh
Assistant Professor
Department of CS and IT
UTD, BU, Bhopal, India

ABSTRACT

Software testing is one of the most important parts of the software development. It also takes too much time to complete, because there are test cases are used for the testing of the software. So data mining techniques are used to improve the performance of the testing by reducing the size of the test cases. In this paper a Parallel Early Binding Recursive Ant Colony (PEB-RAC) System technique is presented with automated testing to provide an efficient way of the software testing. A result analysis is shown in the result section, this analysis shows that proposed technique provide better result as compare to the other technique.

Keywords

Optimization techniques, testing approach, auto case generation, SDLC

1. INTRODUCTION

Software engineering is the field in computer science in which programs are made and compile to design software. Provide reliable software for the user is a basic requirement in software engineering. Thus software testing is one of the key steps in software development life cycle. In software testing, testing to check correctness, completeness and reliability of the software is performed. There are many ways by which defects can be generated in the software like error in the programming is introduced by the programmer, or user requirement cannot be fetched correctly during the requirement analysis phase etc. These issues are generates bugs in the software and degrades the performance of the software.

This paper focuses on automated testing for the software. In this some tools, programs or applications are used to test the quality of the software. In olden days testing is conducted by manually which does not provides performance that is required. So testing by automatic tools can enhance the performance of the testing. In automated testing test cases are used to test the quality of the software. Test cases are the programs which are used to verify functionality and feature of the software. But sometimes redundancy is generated in the test cases that cause degradation in the performance of the automated testing.

Thus data mining techniques are used to provide reliable and small number of test cases for testing that can speed up testing process and also enhance the performance of the automated testing. Data mining is technique in which suitable no. of patterns are generated from the data. And technique can be used for application like classification, regression, clustering etc. thus it provides a suitable no of patterns of test cases which can be provide desired performance as require by the system.

For proposed work a PEB Recursive Ant Colony system is used to provide optimize test cases for the automated testing.

2. RELATED WORK

In this section a study over techniques which used for generating test cases for automated testing is presented. There are different techniques which are used for test cases generation for automated testing. Like in [1] a technique is presented in which UML diagram with genetic algorithm is used to generate test cases. In this technique genetic crossover's technique is used to generate test cases from sequential diagram. And a mutation testing is implemented to test the effectiveness of the methodology. In [2] a test case reduction technique using clustering is presented. In clustering data objects can be group into cluster in which each object having similar properties with respect to other objects in the cluster. In a combination of software engineering and data mining technique is used to provide reduction in the test cases which improves the performance of the testing. WEKA a machine learning tool which is written in java is used to implement the whole process. In [3] automated testing technique with data mining for software testing is presented. In this technique firstly software requirements are converted into UML diagrams and then test cases from that diagram is generated. Then data mining techniques are used to reduce these test cases in that way an efficient way to reduce test cases for automated testing is presented. In [4] technique to categorize functional and nonfunctional requirements of the software is presented. In this a data mining technique over SRS state charts to mine test cases for automated testing is presented and in that way it generated a reliable and small number of test cases for automated testing.

In [24] an ant colony based optimization based test case generation is presented. In test case generation, test cases are generated that satisfies testing criteria. ACO optimization technique which uses the behavior of the ants to provides solution for the many optimization problems. In case of test case generation by ACO, first a transformation graph is generated and then an optimal path is formed to and a condition to stop generation and a process to frame these result to find an optimal path. By using these step test cases is generated from UML diagram. In [25] an intelligent technique based software testing method is presented. In that technique intelligent water drop (IWD) is used with white box testing to generate basis bath testing and ACO ant colony optimization is used for test case generation. That way, this technique provides a method which uses ACO ability to generate test cases and bath testing a technique of white box testing. That provides an intelligent water drop testing technique for software testing. In [25] an ant colony optimization (ACO) based scheduling technique is presented in that technique, ACO's feature a set of software agents and artificial ants is

presented to provide solution for software testing in that case. In that artificial ants are used to provide solution for the problem and each and provide solution in constructive manner by collection component step by step to form a solution. Thus as mentioned in the related work, a data mining technique is can be provide a significant way of test case categorization. For propose work a data mining technique with an optimizing mechanism can be presented.

2.1 Existing technique (ACO Technique)

Ant colony optimization technique is applied while determining the test case generation. Where the natural ant behavior is used to opt test cases. Pseudo code describes the complete process using ACO technique.

ACO Algorithm:

Input: Data Input Program, TC Test condition i-n;

Output: Test case generation table -GenTCi-n;

Begin:

```
{
Load the complete program code by line by line
For each line proceed:
Apply preprocessing technique and parse the conditional
statement clusters.
Apply construct Ant Solutions ();
Extracting Possible test conditions matched with available}
Apply Local Search ();
Generate the Test cases based on Ant Solutions and refine for
publish.
Apply Pheromones ();
} END;
```

Upon processing a dataset the following optimized data is received for the result process.

Optimized Data using Existing algorithm

Optimized Using Existing Algorithm
(("%d %d %d", &a, &b, &c);
((((a+b)>c)&&(c+a)>b)&&(b+c)>a))
((a>=0)&&(a<=100)&&(b>=0)&&(b<=100)&&(c>=0) &&(c<=100))
((a==b)&&(b==c))
((a==b) b==c) c==a))

In order to calculate the precision, recall, accuracy and detection rate we have various parameter values from the dataset progress:

Which are TP, FP, TN, and FN, which is getting in the form of confusion matrix, using which the result parameters calculated using the following formulae

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN)$$

$$\text{Precision} = TP / (TP+FP)$$

$$\text{Recall} = TP / (TP+FN)$$

The above formulae terms usage for result computation.

Upon processing this data further the observed values are:

$$TP= 38, FP = 63, TN = 19, FN = 16$$

$$\text{Accuracy} = (38+19 / (38+63+19+16))*100 = 41.30$$

$$\text{Precision} = (38 / (38+63))*100 = 37.62$$

$$\text{Recall} = (38 / (38+19))*100 = 70.37$$

3. PROPOSED WORK

As discussed in the related work section data mining techniques can provide assistance for automated testing which can enhance the performance of the testing programs.

Dataset: In our approach here further a input sample program is taken [27] example. In [2] a test case reduction technique using clustering is presented. In clustering data objects can be group into cluster in which each object having similar properties with respect to other objects in the cluster. In a combination of software engineering and data mining technique is used to provide reduction in the test cases which improves the performance of the testing. WEKA is a machine learning tool which is written in java is used to implement the whole process

Test case dataset [28] table is further described here which is used in our experimental setup and solution:

Table-1 Large Dataset for experimental Setup

Dataset Name	Input program purpose	Location
Triangle	Triangle classification	http://www.ijarcse.com/docs/papers/Volume_4/2_February2014/V4I2-0396.pdf
Bockyroberts	Multiple Tutorial Programs	https://github.com/buckyroberts/
Career monk	Data structure algorithm programs	https://github.com/careermonk/

Thus this is the programming large dataset which further consider for the experimental setup and analysis as described in table-1 and considered below program.

3.1 Software Test Cases Table

There are various possible test cases associate with the any input dataset program. Where the accuracy and efficiency of the program can be tested, that what cause it may fail while dealing with the program execution. Here is some sample test case which often consider while doing programming with any of the Object oriented programming language. In usual a test case given with the function or its behavior, further more test case ID, description , remarks etc. can be embed with the sample test case.

Table-2 Sample Test Case Table

Scenario	Expected Test Case
Verify the mobile number 10 digit numeric value.	If(Length(mobile)==10) True; If (Mobile!=numeric) False;
if ((a==b)&&(b==c))	If(a>0,b>0,c>0) & not belongs to Alphabet category.
((a>=0)&&(a<=100)&&(b>=0) &&(b<=100)&&(c>=0))	If(variable != alphabets) OR If(Variable all solution in same format)
if(((a+b)>c)&&((c+a)>b)&&((b+c)>a))	Non-Negative and non-zero verification.
Email format verification	If(email.contains("")), If(email.contains("@domain name"))

As per described in above table the sample test cases are always available with the any of the small programming.

Here a dataset input taken triangle classification problem a C language program and further examine the auto test cases generation on processing the program as dataset.

```
#include<stdio.h>
#include<conio.h>
void main ()
{
int a, b, c, valid;
printf("\nEnter the value of three sides: ");
scanf("%d %d %d", &a, &b, &c);
valid=0;
If ((a>=0)&&(a<=100)&&(b>=0)&&(b<=100)&&(c>=0)
&&(c<=100))
{
if (((a+b)>c)&&((c+a)>b)&&((b+c)>a))
{
valid=1;
}
}
if (valid==1)
{
if ((a==b)&&(b==c))
printf("\nEquilateral triangle.");
else if ((a==b)|| (b==c)|| (c==a))
printf("\nIsosceles triangle.");
else
printf("\nScalene triangle.");
}
else{
printf("\n Invalid input (out of range ");
}
}
```

In this paper a PEB Recursive Ant Colony system based optimization technique is presented which provide optimize test cases for the automated testing.

3.2 Parallel Early Binding Recursive Ant Colony Optimization Algorithm

1. Test case generation presenting different dataset (Triangle classification, Bocky Roberts, Career monk) etc. Apply Ant colony optimization algorithm to generate test cases.
2. Partition test cases in to sub-classes.
3. Perform thread creation for each subclass and process for execution in parallel.
4. Perform thread based ant colony optimization for each sub class, apply early binding execution stops.

5. Collect results from each sub-class.
6. Merge results of the sub-classes.
7. Form final optimal result to generate test cases.

3.2.1 Proposed PEB-RACS: In our approach here further a input sample Program

Input: Sample input program, GenTCi-n.

Output: Test case generation.//optimal test cases

Steps: Begin – Apply PEB-RAC

Construct Ant Solutions ()

Local Search on Ant Solution;

Perform Pheromones;

Generate subset partition program ();

Thread Creation Partition ();

Apply Early Binding ();

}

For each partition using

{

New Thread ();

Obtain Cluster CLi-n;

Further ACO on each thread Ti CLi-n;

Apply early Binding ();

Returns merge final result; //optimal test cases.

}

3.3 ACO Technique

The existing approach which is Ant colony optimization where data is processed at using the approach which is stated in related work, in order to proceed with the technique following steps were involved:

3.3.1 Input (Dataset) Program:

As per discussed the above triangle classification program given by the author in paper, here is the Triangle.C considered.

```
#include<stdio.h>
#include<conio.h>
void main ()
{
int a, b, c, valid;
printf("\n Enter the value of three sides: ");
scanf("%d %d %d", &a, &b, &c);
valid=0;
If ((a>=0)&&(a<=100)&&(b>=0)&&(b<=100)&&(c>=0)
&&(c<=100))
{
if (((a+b)>c)&&((c+a)>b)&&((b+c)>a))
{
}
```

```

valid=1;
}
}
if (valid==1)
{
if ((a==b)&&(b==c))
printf("\nEquilateral triangle.");
else if ((a==b)||(b==c)||(c==a))
printf("\nIsosceles triangle.");
else
printf("\nScalene triangle.");
}
else {
printf("\n Invalid input (out of range ");
}

```

3.3.2 Preprocessing Procedure

Upon taking the program as input the complete structure is analyzed and parse using the scanner and further entity such as separator, operators, keywords and such other components are removed and finally the remaining part is pass to the next phase.

3.3.3 Apply ACO Technique

After preprocessing of triangle program coming to the step where ACO is applied with the algorithm and following optimal result is obtain where the optimized statements retrieved.

3.3.4 Result Obtain (Optimal)

Upon applying the ACO technique, following optimal data is retrieved which shows the different mathematical and other available statement is considered.

Optimized Using Extsting Algorithm
("%d %d %d", &a, &b, &c);
((a+b)>c)&&((c+a)>b)&&((b+c)>a))
((a>=0)&&(a<=100)&&(b>=0)&&(b<=100)&&(c>=0) &&(c<=100))
((a==b)&&(b==c))
((a==b) (b==c) (c==a))

Optimal data using ACO Technique

3.4 PEB-RACS Technique

PEB-RACS system based on the fact that performance of the ant colony system for the small size problem is better but in case of large scale data set it degrades in the performance. In PEB-RACS partition scheme similar to the merge sort is presented. In that technique all the nodes of the data set are divided in to two small sub sets and perform the operation of optimization and in that way separate operation for these subsets is performed and desired performance have been reached.

3.4.1 Input (Dataset) program:

In order to process the proposed algorithm further the same Triangle.C program is considered which is given below.

```

#include<stdio.h>
#include<conio.h>
Void main ()
{
int a, b, c, valid;
printf("\nEnter the value of three sides: ");
scanf("%d %d %d", &a, &b, &c);
Valid=0;
If ((a>=0)&&(a<=100)&&(b>=0)&&(b<=100)&&(c>=0)
&&(c<=100))
{
if (((a+b)>c)&&((c+a)>b)&&((b+c)>a))
{
valid=1;
}
}
if (valid==1)
{
if ((a==b)&&(b==c))
printf("\nEquilateral triangle.");
else if ((a==b)||(b==c)||(c==a))
printf("\nIsosceles triangle.");
else
printf("\nScalene triangle.");
}
else{
printf("\n Invalid input (out of range ");
}
}

```

3.4.2 Preprocessing

Preprocessing technique extract the conditional statement data. The process parses the complete data and work on the keywords, constants, separators and other removable constant statements. As in case of our program input, only the if, while, operators related activity come out after the pre-processing technique on dataset to generate auto test case.

3.4.3 Apply PEB-RAC Technique

Further after getting data from the preprocessing technique, PEB-RAC is applied and the optimal data is obtain from the PEB-RAC.

3.4.4 Result (Optimal Data)

Upon processing the same dataset input program the following optimized result is observed

Optimized Data using proposed algorithm

Optimized using Proposed Algorithm
Group 1: $(((a==b), (b==c), (b==c)), (c==a))$
Group 2: $((a+b), (b+c), (c+a))$
Group 3: $(((a>=0), (a<=100), (b<=100), (b>=0), (c<=100)), (c>=0))$
Group 4: $[(\square\%d\%d\%d\square, \&a, \&b, \&c);]$

Thus a method, which combines automated testing with the PEB-RACS system, is presented which improve the performance of the existing technique and provide better performance as compare to the existing technique.

In order to calculate the precision, recall, accuracy and detection rate we have various parameter values from the dataset progress:

Which are TP, FP, TN, and FN, which is getting in the form of confusion matrix, using which the result parameters calculated using the following formulae

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN)$$

$$\text{Precision} = TP / (TP+FP)$$

$$\text{Recall} = TP / (TP+FN)$$

The above formulae terms usage for result computation.

Upon processing this step the further values computed are –

$$TP = 41, FP = 58, TN = 26, FN = 12$$

$$\text{Accuracy} = ((41+26) / (41+58+26+12)) * 100 = 60.90\%$$

$$\text{Precision} = (41 / (41+58)) * 100 = 41.41\%$$

$$\text{Recall} = (41 / (41+12)) * 100 = 77.35\%$$

3.5 Different Dataset outcomes

As per our work performance and applying same procedure with different dataset programs, the following result is obtain using Bocky Roberts and career monk program, the confusion matrix parameter value is obtained.

Bocky Roberts: the obtained value is performed using the ACO technique –

$$TP - 27, FP - 97,$$

$$TN - 17, FN - 2.$$

$$\text{Accuracy} - 21.77, \text{Recall} - 93.10, \text{Precision} - 30.76$$

PEB-RAC Technique: the following result with Bocky Robert is obtained –

$$TP - 43, FP - 28,$$

$$TN - 7, FN - 2.$$

$$\text{Accuracy} - 60.56, \text{Recall} - 95.55, \text{Precision} - 62.5$$

Career Monk: the obtained value is performed using the ACO technique –

$$TP - 29, FP - 51,$$

$$TN - 19, FN - 6.$$

$$\text{Accuracy} - 45.71, \text{Recall} - 82.25, \text{Precision} - 36.25$$

PEB-RAC Technique: the following result with Career monk is obtained –

$$TP - 45, FP - 37,$$

$$TN - 7, FN - 6.$$

Accuracy – 54.73, Recall – 88.23, Precision – 54.87

4 RESULT ANALYSIS

All the experiments were performed using an i5-2410M CPU @ 2.30 GHz processor and 4 GB of RAM running windows 8. The discussed feature selection algorithms were implemented using language Java.

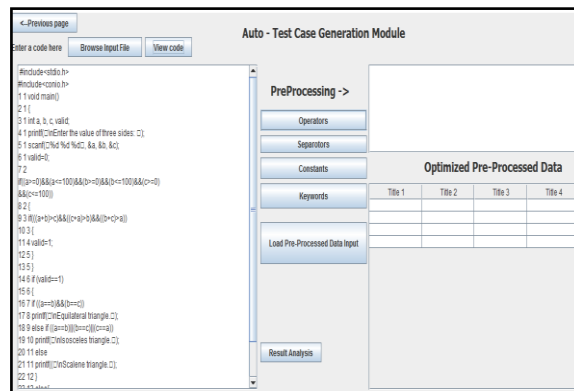


Figure-1: Diagram for Test Case Generation framework

The optimized data on processing with both the technique is also presented in below table such that an optimal data which further required investigating is presented in below screen.

Optimized using Proposed Algorithm
Group 1: $(((a==b), (b==c), (b==c)), (c==a))$
Group 2: $((a+b), (b+c), (c+a))$
Group 3: $(((a>=0), (a<=100), (b<=100), (b>=0), (c<=100)), (c>=0))$
Group 4: $[(\square\%d\%d\%d\square, \&a, \&b, \&c);]$

Figure 2: Optimal pre-processed data after Proposed technique.

A result table is shown in the Table 1 which mentions the comparison for existing technique and proposed technique is presented that shows that proposed method performances well as compare to the existing technique.

Table-3 Comparison Analysis between Existing and Proposed Algorithm

Dataset	Algorithm m System	Accurac y %	Precision %	Recall %
Triangle classification	ACO	41.30	37.62	70.37
	PEB-RAC	60.9	41.41	77.35
Bocky Roberts	ACO	21.77	30.76	93.1
	PEB-RAC	60.56	62.5	95.55
Career monk	ACO	45.71	36.25	82.25
	PEB-RAC	54.73	54.87	88.23

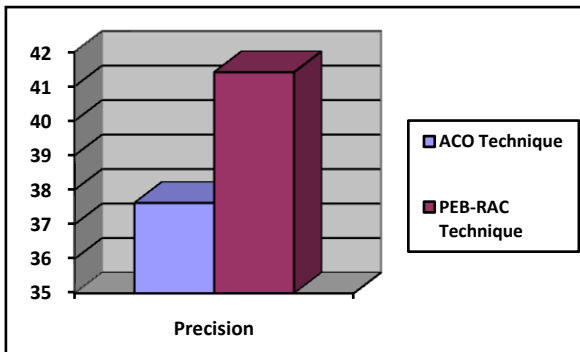


Figure 3: Precision Bar Graph comparison between existing and proposed algorithm using Triangle program.

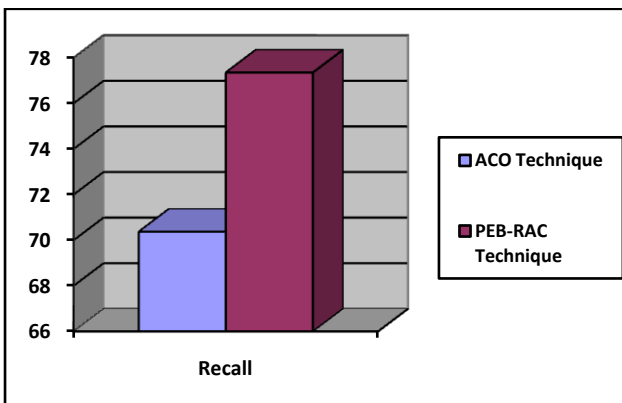


Figure 3: Recall Bar Graph comparison between existing and proposed algorithm using Triangle program.

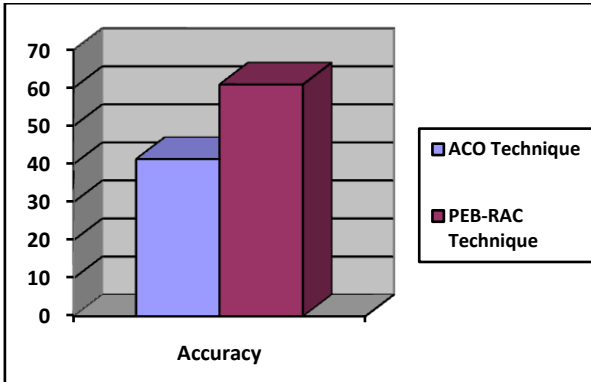


Figure 3: Accuracy Bar Graph comparison between existing and proposed algorithm using Triangle program.

In this graph we have discussed and compared the obtained result analysis obtain from the experimental setup and result part by which we observe the obtained parameter value prove that the proposed system algorithm suite best as compared with existing algorithm.

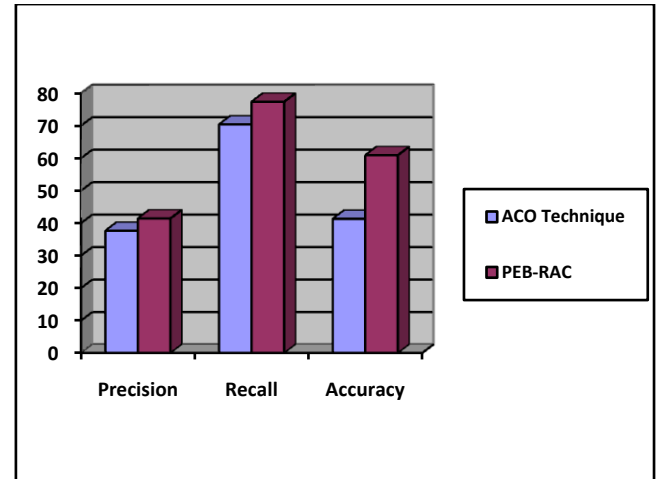


Figure 4: Combine Result Analysis Graphical view using Triangle classification dataset program.

A result analysis for the existing technique and proposed technique is shown in the Table-2. This shows that proposed technique provides better results than existing technique or proposed technique provides optimized no. of use cases as compare to the existing technique.

5 CONCLUSION

Software testing is one of the key phases in software development process. In this paper various issues related to the automated software testing like redundancy in the test cases is presented. To reduces these issues in automated testing data mining techniques are used which enhance the performance of the whole automated testing technique. A PEB-RACS based optimizing technique with automated techniques is proposed. A result analysis for that method is shown, which shows that proposed technique provide better result as compare to the existing technique.

6 REFERENCES

- [1] Paramjit Kaur, Rupinder Kaur “approaches to generating test cases automatically to test the software” IJEAT, Vol. 2 Issue 3, and february2013.
- [2] B. Subhasini, D Jeyamala “reduction of test cases using clustering technique” IJRSET, vol. 3 issue 3, March 2014.
- [3] Sarita Sharma, Anamika Sharma “Amalgamation of automated testing and data mining: A novel approach in software testing” IJCSI Vol. 8, Issue 5, No. 2, September 2011.
- [4] Lilly Ramesh, G.V. Uma “reliable mining of automatically generated test cases from software requirement specification” Vol.7, Issue 1, No. 3, January 2010.
- [5] K. Jain, M. N. Murty, and P. J. Flynn. A Data clustering: review. ACM Computing Surveys, 31(3):264–323, 1999.
- [6] Lilly Ramesh, “Knowledge Mining of Test Case System,” International Journal on Computer Science and Engineering Vol.2 (1), 2009, 69-73.
- [7] Mark Last and MenahemFriedman.”The Data Mining approach to automated software testing.”.Communications of ACM, 2003.

- [8] Martina marre and Antonia Bertolino, “using spanning sets for coverage testing”. IEEE transactions on software Engineering, vol.29.
- [9] Zhenyu Chen and BaowenXu.”A novel approach for test suite reduction based on requirement relation contraction”.Communications of ACM, 2006.
- [10] KartheekMuthyala, Rajshekhar Naidu P “A Novel Approach to Test Suite Reduction Using Data Mining” computer science and information system Birla Institute of Technology and Science.PilaniRajasthan,India Vol.2 No. 3 Jun-Jul 2011.
- [11] P. Samuel R. Mall A.K.Bothra “Automatic test case generation using unified Modeling language (UML) state diagrams” Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, West Bengal, India E-mail: philips@cusat.ac.in
- [12] VasilacheS., and Tanaka J., "Synthesis of State Machines from Multiple Interrelated Scenarios Using Dependency Diagrams," Journal of Systemics, Cybernetics and Informatics, Vol.3, No.3, 2006.
- [13] Lilly Ramesh and G.V. Uma “Reliable Mining of Automatically Generated Test Cases From Software Requirement Specification (SRS)” IJCSI Vol. 7, Issue 1,No. 3,January 2010.
- [14] Dave Kelly, “Software Test Automation and the Product Life Cycle”, Symantec Corporation, Volume Number: 13 (1997), Issue Number: 10
- [15] Sarita Sharma,Anamika Sharma ,Computer Science Department ,DAV institute of Management Faridabad Haryana 121001 India ,Amalgamation of Automated Testing and Data Mining: A novel Approach in Software testing, IJCSI Vol 8, Issue 5, No2, September 2011.
- [16] I.S Board, IEEE standard for Software unit Testing,IEEE Computer Society,New York,YK1987.
- [17] J. Han and M. Kamber, "Chapter 5: Mining Frequent Patterns, Associations and Correlations," in Data Mining: Concepts and Techniques, San Francisco, CA, Morgan Kaufmann, 2006, pp. 227-283.
- [18] Karl Nils Gunderson “An Application of Association Rule Mining To Unit Test Selection” Fargo North Dakota June 2013
- [19] Sotiris Kotsiantis, DimitrisKanellopoulos, Association Rules Mining: A Recent Overview, GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71-82
- [20] GagandeepKaur, ShrutiAggarwal , Performance Analysis of Association Rule Mining Algorithms, International Journal of AdvancedResearchinComputerScienceandSoftwareEngineering, Volume 3, Issue 8, August 2013, ISSN: 2277 128X
- [21] Trupti A. Kumbhare, Prof. Santosh V. Chobe Associate Professor, DYPIET, Pimpri, Pune, India “An Overview of Association Rule Mining Algorithms” IJCSIT Vol. 5 (1) , 2014, 927-930
- [22] Data Mining: Concepts and Techniques, 2nd Edition, Jiawei Han and MichelineKamber, Morgan Kauffman, 2006.
- [23] Fayaz Ahmad Khan, Dr. Anil Kumar Gupta, DibyaJyoti Bora Department of Computer Science and Applications, Barkatullah University Bhopal, (M.P) “An Efficient Approach to Test Suite Minimization for 100% Decision Coverage Criteria using K-Means Clustering Approach” IJAPRR ISSN (2350-1294) Vol. II, Issue VII, 2015.
- [24] Huaizhong Li and C. Peng Lam “Software Test Data Generation using Ant Colony Optimization” World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:1, No:1, 2007.
- [25] Laheeb M. Alzubaidy 1, Baraa S. Alhafid “Proposed Software Testing Using Intelligent techniques (Intelligent Water Drop (IWD) and Ant Colony Optimization Algorithm (ACO))” IJCSI, September 2013.
- [26] MinalC.Toley, 2 Prof.V.B.Bhagat “An Application of Ant Colony Optimization for Software Project Scheduling with Algorithm in Artificial Intelligence” IJAIEM, February 2015.
- [27] SapnaVarshney*, Monica Mehrotra,” Automated Software Test Data Generation for Data Flow Dependencies using Genetic Algorithm”, Volume 4, Issue 2, February 2014.
- [28] D. Azar, J. Vybihal,” An ant colony optimization algorithm to improve software quality prediction models: Case of class stability”, December 2010.