A Cat Swarm Optimization Based Test Case Prioritization Technique to Perform Regression Testing

Ms. Manaswini B, Rama Mohan Reddy A

Abstract: Developing the software is a very essential in now a day. Software development lifecycle has phases like requirements, designing, coding, testing and deployment. 50% of the time will be consumed by the testing phase. Software testing is a cost effective phase. Testing phase will be done at various levels with different kinds of the tests for identifying the defects based on the user requirements. One of the major and important test was the regression test. This test will be performed after adding the additional functionalities to the existing software. The different techniques to perform the regression test are selection of test cases, prioritizing the existing test cases, reset all etc. This paper proposed a novel algorithm to perform the regression test based on the prioritizing the test case technique by using the cat swarm optimization algorithm. The results that performed on the open source applications like jtopas, jmeter was shown the effectiveness of the proposed algorithm in the parameters of execution time and false detect rate.

Index Terms: Software, Testing, cat swarm, optimization, Regression test, and Test case prioritization.

I. INTRODUCTION:

The process of developing the software was done by using the lifecycle of SDLC. 50% of the software development life cycle cost will be expensed by the testing phase[1][24]. In many organizations the software is very essential. To deliver the good and efficient software to the customers is tremendous task. So, before delivering the product to the customer, for identification of the hidden defects/false the software testing will be performed. At different levels various tests will be performed with the different kind of tests. Few of the tests like unit test, integration test, system test, acceptance test, regression test, alpha test, beta test, performance test, load test, stress test, usability test, security test, portability test. The major and essential test was the regression testing[24]. The regression test was important and cost effective testing. This was performed after adding the extra/new facilities to the existing software. To the new facilities/new advantages to the existing software may leads to various faults due to adding the extra functionalities to overcome this issue the regression testing will be performed. Already many researchers has proposed various approaches for decreasing the regression test cost, the frequently used approaches are “Regression test case prioritization”, “Selection of Test cases”.

Few of the selection of testcase methods are in the papers [3][4][5][6][7][8][9][10][11][12] and few of prioritizing test case approaches are used to increase the fault detect rate are in the papers [3][13][15][16][17][18][19][20][21][22][23].

Prioritizing the test cases will depend on the permutations of the different test suites which has maximum value of the objective function. Regression testing is used to perform the testing and to ensure that there are no any errors in the software after adding the extra functionalities. It is a type of an integration testing which requires more time to perform testing. Regression testing can be performed in various methodologies like retest all [25][26], test selection[25][27][28], test case reduction [25][29][30] and test case prioritization [25][31][32]. Due to the regression testing software quality was enhanced, the test coverage will be increased checks the impact of added code etc. [35] The general practice for the regression test is to re-execute all the type of test suite that which consumes more cost[36]. To perform the cost effective and high detection rate the current paper used an approach based on the cat swarm based optimization algorithm.

A. CatSwarm Approach:

Based on the various behaviors of various cats chu et. al. proposed a technique called as a cat swarm optimization technique. The cats generally have two phases seeking phase and tracking/tracing phase to move the cats in the solution criteria. In the seeking phase the cats will find the factors like seeking memory pool, seeking range for selected dimensions, and number of coordinates will be changed and the self-coordinates for the estimation of movement of the cats where as in the tracing mode it checks for the best value by that calculate in seeking mode. The seeking and tracking repeatedly performed until we get the good solution[37]. In this paper proposed and implemented an cat swarm approach to perform the regression test by prioritizing the test suites technique using the cat swarm optimization algorithm. The remainder of the paper is, in the next section related work was discussed, in section III the proposed work was discussed, in section IV the results and discussion was discussed and finally in section V conclusion was discussed.

II. RELATED WORK:

In paper, [1] authors proposed an algorithm for regression testing based on the genetic algorithm along with the multi-fitness value.
A Cat Swarm Optimization Based Test Case Prioritization Technique to Perform Regression Testing

It considers the metrics like flow of control with various conditions, statements which covered by the test case. The proposed was compared with other existing techniques and results proved that the proposed was superior compared to the existing once. In paper [2] proposed a method for arranging the tests in the prioritized manner. This method is the combination of the programming method and mahalanobis taguchi method. Here considered the parameters like runtime of various tests and prioritizing manner of tests, the experimental results were performed on 300 industrial based software test cases and proved that it was more effective when compared to conventional methods.

In paper [3] authors proposed a heuristic technique by using the lexicographical ordering notation to increase the fault detect rate by arranging the test case in an order and it proved that good in terms of fault detect rate when compared to the existence once.

In paper[24] authors proposed a method to perform the regression test, based on the Ant colony optimization algorithm for the distribution environment which was done on the Hadoop framework.

In paper [25] authors proposed an algorithm for arranging the test cases in an order based on the importance of the test case by using the ant colony optimization technique. By considering the factors like detection of faults, time of execution and severity of faults and the effectiveness was proved in terms of the APFD value.

In paper [33] author proposed an approach to improve the effectiveness of the regression test by using the metric fault severity.

In paper [34] author proposed a technique to prioritize the test cases based on the test process by identifying the faults and by considering the metric APFD.

In this paper [35] author proposed an algorithm based on the grouping technique to perform the regression test by prioritizing the test cases with the metric scope of the code.

In this paper [36] proposed an approach that enhance the effectiveness and efficiency of regression technique and this technique will decrease the cost of the regression test.

III. PROPOSED SYSTEM:

In the proposed system cat swarm optimization algorithm was used for arranging the test cases in the prioritizing priority based on the fitness value that which calculated by using the metrics like coverage scope, fault detection rate and execution time for the various tests. The proposed algorithm was shown in Algorithm 1. The flow of the algorithm was shown in the Figure 1, Figure 2, and Figure 3.

The proposed algorithm will work in two stages Seeking stage and Tracking/Tracing stage. In the seeking stage it calculates the Fault detection rate, coverage scope, and execution time and Fitness value by using the formulas (1) and (2) for every test case as shown in the Table 1. The flow of the seeking phase was shown in the Figure 2. Whereas in the Tracing phase based on the fitness values of individual test cases all the tests were arranged in the decreasing order. The flow of the tracing phase was shown in the figure 3.

The sorted test case will be in the priority order which is the output of the Algorithm 1. The overall flow of the proposed algorithm was shown in the Figure 1.

Algorithm 1: Cat swarm optimization based regression testing

<table>
<thead>
<tr>
<th>Input:</th>
<th>T_i</th>
<th>T_1, T_2, T_3, T_4, ..., T_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>T_i</td>
<td>T_1, T_2, T_3, T_4, ..., T_n</td>
</tr>
</tbody>
</table>

1: Start  
2: Repeat  
3: For every Test case T_i  
4: Seeking phase  
5: Fault detection Rate F_D_i  
6: Execution time of TE_i  
7: Coverage Scope TC_i  
8: Calculate Fitness value FV_i  
9: Tracing Phase  
10: Based on fitness value arrange the test cases in order.  
11: End for  
12: Until all the test case finish.  
13: End cat swarm optimization based regression testing

Where

T_i : Test cases  
T_p : Test cases in prioritizing order.  
n : Total Number of test cases.  
F_D_i : Fault Detection rate for individual test case  
TE_i : Execution time of the test case for individual test case.  
TC_i : Coverage Scope  
FV_i Fitness Value.  
HFD_i : Higher Fault detection rate  
LTE_i : Less Execution time  
HTC_i : More coverage Scope

Fault detection Rate

\[ F_D_i = \frac{\text{Number of faults detected by Test case } T_i}{\text{Execution time of Testcase } T_i} \] (1)

Fitness Value

\[ FV_i = HFD_i \cap LTE_i \cap HTC_i \] (2)

The Table 1 consists of the fifteen test cases with the faults for every individual test case that obtained from the open source application Jtopas and Jmeter.
Table 1: Test cases with the Faults

<table>
<thead>
<tr>
<th>Faults</th>
<th>Test Cases</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
<th>F13</th>
<th>F14</th>
<th>F15</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T8</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T9</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T11</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T12</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T13</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T14</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T15</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Figure 1: Flowchart for proposed algorithm.

Figure 2: Flowchart of Seeking Phase.
A Cat Swarm Optimization Based Test Case Prioritization Technique to Perform Regression Testing

Figure 3: Flowchart of Tracing Phase

IV. RESULTS AND ANALYSIS:

The proposed Algorithm was applied on the opensource applications like jtopas and jmeter. The results was shown in the Figure 4, Figure 5 and Figure 6 in terms of execution time and fault detect.

Figure 4: comparison of the existing and proposed systems on the Jmeter application in terms of fault detection rate.

Figure 5: comparison of the existing and proposed systems on the Jtopas application in terms of fault detection rate.

Figure 6: Fault detection rate of the proposed system on both the Jtopas and Jmeter applications

Table 2: shows the order of the test case before and after the proposed system
Table 3: Faults identified and execution Time for the individual Test case

<table>
<thead>
<tr>
<th>Test case</th>
<th>Faults Detected</th>
<th>Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>T2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>T3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>T4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>T5</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>


Original Order | Prior Order
---|---|
T1 | T14
T2 | T9
T3 | T12
T4 | T3
T5 | T6
T6 | T4
T7 | T7

<table>
<thead>
<tr>
<th>T6</th>
<th>T15</th>
</tr>
</thead>
<tbody>
<tr>
<td>T7</td>
<td>1</td>
</tr>
<tr>
<td>T8</td>
<td>4</td>
</tr>
<tr>
<td>T9</td>
<td>3</td>
</tr>
<tr>
<td>T10</td>
<td>4</td>
</tr>
<tr>
<td>T11</td>
<td>2</td>
</tr>
<tr>
<td>T12</td>
<td>1</td>
</tr>
<tr>
<td>T13</td>
<td>1</td>
</tr>
<tr>
<td>T14</td>
<td>2</td>
</tr>
<tr>
<td>T15</td>
<td>2</td>
</tr>
</tbody>
</table>


V. CONCLUSION:

The work in this paper was based on the software testing particularly to perform the regression testing based on test case prioritization technique. To accomplish that here proposed algorithm based on the normal behavior of the cats called as a cat swarm optimization algorithm. This algorithm was performed in two phases called as seeking and tracing phase which will help to arrange the testcases in the prioritized order based on the fitness value. The proposed algorithm was implemented on the open source applications like jtopas and jmeter. The Results had shown the effectiveness of proposed algorithm in terms of the execution time and the fault detection rate.

REFERENCES:


A Cat Swarm Optimization Based Test Case Prioritization Technique to Perform Regression Testing