A Hybrid Approach of Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) Algorithm for Test Case Optimization

Sultan Singh Saini, Vipin Jain
Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan Jaipur-302017, (INDIA)
Email: sainisultan@gmail.com
Received 12.02.2019 received in revised form 15.04.2019, accepted 17.04.2019

Abstract- Software testing is a phase of the software development lifecycle model. It is a special programming framework that tries to find errors. The tester compares the expected or generated results with the original results or previous results. The test method manages the testers and exposes them to the reference points that the testers need to complete. The existing frame work uses Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) Hybrid model for test case prioritization, although the basic destination is statement coverage and fault coverage, because it used to be ABC Algorithm, so it is the convergence rate is moderate. Likewise, ABC algorithm based on greedy method, some great solution cannot be easily ignored because greedy method chosen now seems the best arrangement. Because of this, the proposed method, rather than adding the PSO algorithm and the ABC algorithm, merges the PSO algorithm with the ACO as ensured by the algorithm convergence rate will be described. In addition, regression testing in the case of our application retests all regression testing technology, which means that we will no doubt perform all the test cases in each iteration. This result in extra effort and swallows up a lot of time and assets. In any case, due to the retesting of time and asset requirements, all methods cannot be implemented in any way. To overcome this problem, the proposed method uses a hybrid approach of regression testing that incorporates regression test priorities and regression test choices. This paper shows a hybrid approach that allows the selection and prioritization at different levels. The goal is to use the prioritized grouping after selection. Subsequently, the prioritization and selection arrangement reduces the number of test cases in which the code is executed in this manner, thereby reducing running and testing time by utilizing minimal resources

Keywords – Testing, PSO, ABC, ACO, Test Cases

1. INTRODUCTION

Software testing [1] is a phase of the software developmentlife cycle model. It is the most significantstage of software development. It is a special programming framework that tries to find errors. In software testing, testers use the current production to view the production that has been created. The tester also compares the expected or generated results with the original results or previous results. In software testing, testers measure the quality of product execution. In order to test successfully, the tester should accept the test procedure. The system provides a guide that usually describes what steps a tester should take and when to start testing.

1.1 General Characteristics of Strategic Testing

- To be a strong test, product groups should guide the professional success of formal investigation
- From the beginning of the test section level, outward coordinate the entire PC-based framework
- Test different strategies for a variety of priorities and time
- Testing by the designer (for a wide range of activities) and from the free test group guidance products
- Tests and surveys are a variety of exercises, but must exclude any testing technology failure

1.2 Various levels of Testing

There is four level effort included, and how long it takes. The test method manages the testers and exposes them to the reference points that the testers need to complete.

1.3 Regression Testing

Regression test [2] is at performed during the maintenance phase of software. This is done to check that any updates to the software will not affect the existing software. There are four types of regression testing techniques. The first is to retest all methods. Here, all test cases that exist in the test suite should be executed. It consumes a lot of time and computing resources.

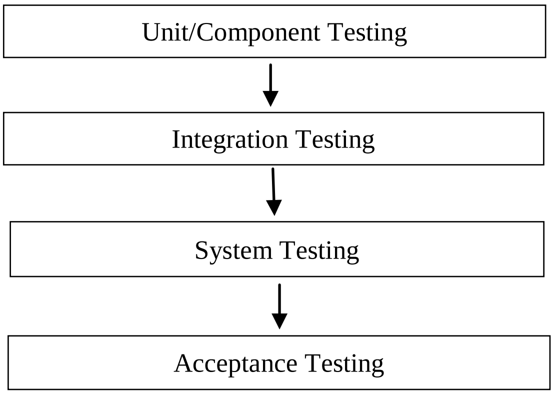

Figure 1 : Various Levels of Testing

1.4 Test Case Prioritization

Test case prioritization [3] schedule test cases in order to increase their ability to meet some performance goal:

- Rate of fault detection
- Rate of code coverage
- Rate of increase of confidence in reliability

1.5 Ant Colony Optimization

Dorigo [4] proposed ant colony optimization, clear behavior by observing nature and draw ants in conclusion.In the chase of the sustenance, the ants leave a pheromone in the way of navigation, finding the shortest path through collaboration and pheromone evaporation. It is expected that at the beginning of the nutrient removal, the ants choose theirway to the food source route arbitrarily, and although they go to the intersection, any conceivable way will keep the pheromone smell.

1.6 Particle Swarm Optimization

It is a computational technique that optimizes problems by iteratively trying to enhance candidate solutions for a given quality metric. It does this by providing a set of candidate solutions, and these particles move to the position and velocity of the particle up to solve the problem in the search space. Each particle motion Subject to Its location near the most significant impact of names, but at the same time be directed to chase the most space of the location name, these positions are updated to a better location. PSOaffectedby the area's most well-known positions, but also to guide the search space of the most well-known locations that were updated for better position.

2. LITERATURE REVIEW

In Abraham Kiran Joseph, Dr. G. Radhamani [5], the performance of the proposed method is compared with all swarm intelligence algorithms, and it has been observed that the performance of the PSABC algorithm is much better than the ABC, ACO and PSO algorithms. It was also observed, as compared with ACO, ABC and PSO algorithm, the proposed algorithm has more coverage range.

The research effort is also aimed at minimizing costs and time to regression testing. The results show that PSABC performs better than ABC, ACO and PSO.

Anjali Chaudhary and Tarun Dalal [6] used ACO to select test cases and prioritize them. The goal here is to find the space and time complexity of the ACO algorithm. As used herein, the tool is ant colony optimization. The motivation of the ACO algorithm is the foraging behavior of ants. The ants here are blind, and in order to obtain food sources, they use antennas and pheromonetests. When searching to find food, leaving a liquid called ant pheromone on the path. Other ants reach the food source through the shortest path with the help of this pheromone. But as time goes on, the pheromone test will also evaporate. Therefore, all ants will get the path with the largest amount of pheromone. Ants use their scent to detect pheromones on the path.

Kamna Solanki et al. [7], has assessed "m-ACO"("modified ACO") comparing the new test case prioritization technique for regression testing. "M-ACO" By changing the selection criteria for natural food sources ants to give priority to test cases, in order to enhance the diversity of failure. The code for the suggested technique for determining the test case "m-ACO" priority has been implemented in the Perl language. In this paper, Genetic Algorithm (GA), Bee Colony Optimization (BCO) algorithm and ACO algorithm are used to compare and evaluate the proposed "m-ACO" test case priority technology. Learn two measures have been used, i.e. Average Percentage of the Detected Failure (APDF) and the PTR ("complete failure of the desired percentage of coverage of the test suite") to measure the proposed "m-ACO" effective technique Sex. The proposed technical "m-ACO" generating optimal or near optimal solutions. Compared with GA, BCO ACO and methods, the proposed "m-ACO" technology proved its efficiency. The proposed techniques to improve the ACO method by altering the food source selection criteria. Future native ants work in this direction would be to use some well-known relative evaluation of the proposed "m-ACO" technology.

3. PROBLEM DESCRIPTION

3.1 Existing Problem

In the previous frame, the PSO uses the hybrid model ABC test to prioritize, while the base statement covers the destination and fault coverage because it uses the ABC algorithm, which has a moderate convergence speed. Similarly, based on the ABC algorithm using the greedy method, some good solutions are not easily overlooked because the greedy method chooses the best permutation that now appears. The proposed method combines the ABC PSO algorithm and algorithm. The PSO algorithm combines with the ACO algorithm, which will be described because of its ACO guaranteed convergence speed.

In addition, regression testing in the case of all regression testing techniques we use retest means that we will undoubtedly execute all test cases in each iteration. This leads to extra effort and engulfs a lot of time and assets. In any case, due to retesting time and asset requirements, all methods cannot be implemented in any way. To overcome this problem, the proposed method uses a hybrid approach for regression testing that combines regression test priority and regression test choices.

3.2 Proposed Solution

The proposed method is a hybrid approach that allows for selection and prioritization at different levels. The goal is to use the prioritized grouping after selection. Subsequently, the prioritization and selection arrangement reduces the number of test cases in which the code is executed in this manner, thereby reducing run and test time by utilizing minimal resources. The proposed method is divided into four phases. The main stage is a polyethylene - based The acoustic Ming fan circumference. This strategy aggregates test cases into a cluster. Once the test cases are brought together, the following assignments are priorities within the cluster. Here, each group is selected and a hybrid PSACO(PSO + ACO) algorithm is implemented to discover the priority test suite for each cluster. When we have a cluster and determine when the priority of test cases, are now following assignment from each cluster to select the most convincing test, then perform a similar procedure for the remaining clusters. The selection test case is provided as output to the last stage, the inter-cluster priority. Here again choose the most convincing test cases, regardless of their cluster ID. So I left a test suite optimized.The proposed method is divided into four phases as follows:

1. Poly class
2. The group prioritization order
3. Selected test case selection
4. Cluster among priority level

4. METHODOLOGY

4.1. Proposed Methodology

The proposed approach is a hybrid approach that is divided into four phases. In the first phase i.e. CLUSTERING the test cases will be grouped into various clusters based on their statement coverage. For this randomly select ‘c’ cluster centers will be selected. Then calculate the distance between each data point and cluster center. The data point will be assigned to the cluster whose distance from the cluster center is minimum of all the cluster centers. In this way the test cases will be housed into various clusters. Now, in the next phase i.e. Intra-CLUSTER PRIORITIZATION a hybrid PSACO (PSO+ACO) will be applied on each test case to prioritize the test cases. Randomly start with a test case and store this test case in set S. Then probabilistically decide which test case to be visited next. Store the next test case too set S. This process of recalculating the probabilities and moving the test case in S will go on until all the test cases are covered. The test cases of set S will be the prioritized test cases of that cluster. Once all the test cases are covered the next task is to sort the clusters in decreasing order of their priority. Then the highest priority cluster will be selected and the Influential trace for each test case of that cluster will be evaluated. For performing the selection based on the influential traces of test cases belonging to same cluster needs to be compared. The tenets for choice are:

1. For two test cases having same "Influential Trace" as far as variables creating or utilizing same variable, any of the two can be utilized.
2. For two experiments having distinctive "Influential Trace" both are to be incorporated for execution.

The selected test given by the above function, and as it is the final phase of the input, i.e. INTER-the CLUSTER Prioritization. Here, the impact tracking of all selected test cases will be compared to find the final optimized test case. The output of this phase will be the final optimized test case that will be executed to test the software.

5. PERFORMANCE ANALYSIS

5.1 Comparison of the Proposed Approach with the Existing Approach Based on the Number of Optimized Test Cases

To compare the proposed methods with existing methods. The same parameters are used to compare the two methods, i.e. both methods use the same fault coverage and statement coverage, and it has been observed that the proposed method saves more test cases than existing methods. Furthermore, the proposed method uses statement coverage and fault coverage more efficiently than existing methods. Two technologies are surgery graphical representation as follows:

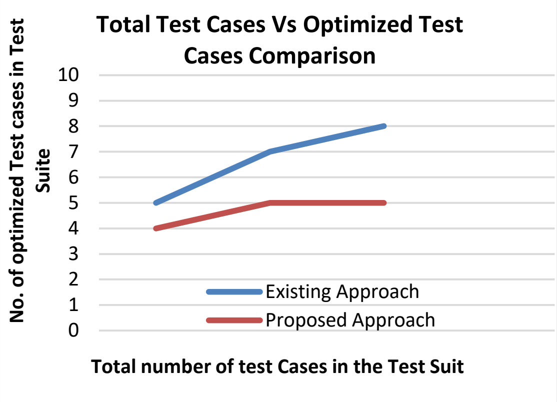

Figure 2: Total Test Cases v/s Optimized Test Cases Comparison

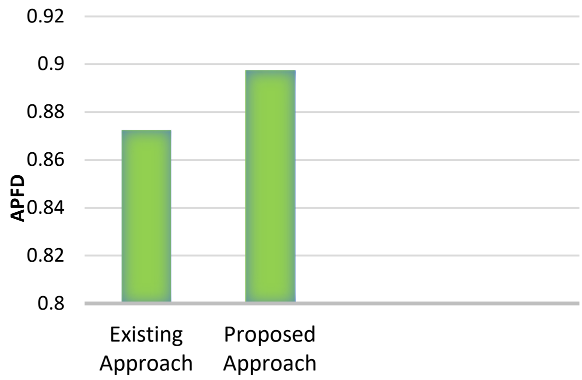

Figure 3: Plot between APFD metrics of Proposed Approach and Existing Approach

5.2 Determining the Effectiveness of the Proposed Approach

The prioritization of experiments has different goals. One of the goals is to correctly identify faults in the test suite at a reasonably expected time. This is used to enhance the fault identification capabilities of the test suite. For this reason, there is a metric called APFD (Average Percent Failure Detection).APFD refers to the standard by the Ro-thermal proposed.

It gives the number of normal faults identified during the life of the test suite. The higher the self-esteem of APFD, the higher the fault detection rate. However, this metric can only be used when the tester can obtain prior knowledge of the fault.

The APFD is given by:

APFD = 1 – (TF1 + TF2 + …………… + TFm)/nm +1/2*n

Where, n is the aggregate number of test cases in the test suite;

m is the aggregate number of faults detected in the test suite;

TFi is the position of the first test case in test suite that reveals fault i.

APFD Value for the proposed approach is calculated as:

APFD = 1 – (31/400) + 1/40= 0.8975= 0.9

APFD Value for the existing approach is calculated as:

APFD = 1 – (41/400) + 1/40= 0.8725

In this manner, it is clear that from the above calculation, the method proposed APFD estimated count than the current method APFD estimated counts more prominent. Therefore, the proposed method has better fault detection capability than the existing methods. Along these directions, the proposed method is more efficient than the existing methods. A similar survey of the proposed method and the current methodology shows the following:

6. CONCLUSION AND FUTURE WORK

In this work, a method is proposed to increase the income convergence speed and minimize execution time, in order to select and prioritize test cases provide the best solution. The proposed method is divided into four phases. The first is the cluster; the second is the intra-cluster priority, followed by the test case selection and inter-cluster priority. In the second phase, the PSACO algorithm was used. Although related to the ACO algorithm, the ants here are test cases that change over time. Here, the main goal of ants is to find test cases with the highest fault coverage so that faults can be identified at an earlier stage. The ant's plan is to find areas with higher node coverage. Based on the proposed method, the best results for test case execution are obtained. The PSO and ACO (Cause ACO) is to use a combination of ACO security algorithms permit convergence speed and the PSO algorithm of dynamic adaptability. In addition, ACO algorithm features can produce high quality solutions. Due to this promising nature of the ACO and PSO algorithms, the proposed method combines PSO and ACO compared to existing methods advantages to provide more optimized test cases. The performance of the proposed method was evaluated and compared to the prior art. Also measure the effectiveness of the method proposed by APFD. The values are calculated for the proposed method as well as the existing methods, and it has been observed that the proposed method is more efficient than the existing method because the proposed method has an APFD value greater than the APFD value of existing methods. This work in the examination into account objective is covered by a valid statement, fault coverage; speed up the convergence as well as students to a more accurate and optimized test suite. In the future I hope other applications of swarm intelligence algorithms on the same method to check efficiency.

REFERENCES

[1] SiripongRoongruangsuwan, JirapunDaengdej, “Test Case Prioritization Techniques”, Journal of Theoretical and Applied Information Technology (2010).

[2] Colorni, Dorigo, Maniezzo, and Trubian, “Ant System for Job-Shop Scheduling”, Belgian Journal of Operations Research, Statistics and Computer Science (1999), 34, 39–53.

[3] S.Elbaum, A.G. Malishevsky, and G. Rothermel, “Test case prioritization: a family of empirical studies” IEEE Transactions on Software Engineering (2002), 28, 2, 159-182

[4] Dorigo and Socha, “An Introduction to Ant Colony Optimization”, IRIDIA - Technical Report Series (2006).

[5] Abraham Kiran Joseph and G. Radhamani, “A Hybrid Model of Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) Algorithm for Test Case Optimization”, Int. J. Comput. Sci. Eng. (2011), 3, 5.

[6] Anjali Chaudhary, TarunDalal, “Evaluating the Effectiveness of Regression Test Selection using Ant Colony Optimization” (2015).

[7] Kamna Solanki, YudhVir Singh and Sandeep Dalal, “A Comparative Evaluation of m-ACO Technique for Test Suite Prioritization”, Indian Journal of Science and Technology (2016), 9, 30.