A Framework for Optimizing Software Regression Test Case based on Modified-Ant Colony Optimization (M-ACO)

Abrahmi Silvarajoo[1]*, *Mohammed* Hazim Alkawaz[2]

[1]School of Graduate Studies Sciences, Management and Science University, Shah Alam,Selangor,.Malaysia
[2]Faculty of Information Sciences and Engineering, Management and Science University, Shah Alam,Selangor,Malaysia

Abstract. Software Regression Testing (SRT) helps with the modified code to ensure that no other new defects have been trained and significantly impact the existing code. SRT is a time and cost taking process. Test Case Prioritization (TCP) helps to reschedule test cases (TC'S) and picks according to the prioritization, which allows the software testers to maximize the number of faults detected within a given time frame. Ant Colony Optimization (ACO) is an optimization algorithm motivated by the searching behavior of insects. This paper proposes a simple framework for the novel Modified-ACO (m-ACO) to prioritize the TC's. m-ACO is planned as the adjusted type of the ACO method, which reschedules the execution sequence of TC'S by changing the wonder of characteristic insects for choosing their food source. The proposed Framework will help the software testers save their time to execute SRT TC's. It has been tested for Zasta Billing Web Application and compared the time taken to complete the SRT manually and using the proposed Framework.The proposed Framework helps to reduce the time taken from 90 days to 45 days to achieve the SRT.

1 Introduction

ST measures the quality of the software [1]. The focal point of quality is to find and remove defects [2]. Inside the test cycle, the testing procedure that guarantees that the highlights are not influenced by the progressions made in programs is known as SRT [3]. The objective of SRT is to ensure that any new adjustments of the product ought not unfavorably influence the current capacities, and there is no further shortcoming covered up in the embedded code [4].SRT is regularly performed throughout the programming improvement and upkeep life cycle [5].Nonetheless, as the product turns out to be huge, the quantity of the test cases will increment after some time, and the expense of performing SRT could turn out to be increasingly costly[6]. An enormous scope ST, a complete SRT frequently takes weeks or even a long run [7]. For this situation, the tester needs to sort the TC's to ensure that the higher priority can be executed right on time as expected under the circumstances [8]. This paper

* Corresponding author: abrahmisilvarajoo@gmail.com

proposes a framework for TCP's novel technique, "m-ACO." Research has been conducted for previous m-ACO studies, and we have come up with a user-friendly framework for m-ACO, which will help the software testers prioritize the TC's very quickly. Also, previous studies only focused on generating the algorithm and executing it in a small SRT environment[9].But, in this study, the proposed algorithm has been discussed in detail. It has been tested for a startup company web application, Zasta Billing Web Application. m-ACO is planned as the adjusted ant colony optimization method, which reschedules the execution sequence of SRT TC'S by changing the wonder of characteristic insects for choosing their food source.

2 Literature Review

SRT is one sort of ST performed with current software changes [10]. It is crucial to give conviction that such changes, which areas of late introduced, don't impede the lead of the unaltered existing bits of the software. Retesting all,TC Selection (TCS),TCP,and TC Reduction (TCR)/TC Minimization (TCM) are regression testing activities[11-12].Retest all is a re-execution of all TC's in a specific capacity test suite. It is an excellent skill, yet be that as it may, it can devour a great deal of time. Instead of re-executing the whole test suite, regression TCS is intelligent to choose a specific piece of a test suite to be run. TCP is the strategy to organize the rundown of TC's, empowering TC's with higher need, chosen based on some predefined paradigm, to be executed before the experiments with a lower need, to meet the required execution objective [13]. So TCP strategy does not dispose of any TC's, thus keeping away from the downside of the TCM procedure.TCM procedures eliminate the repetitive TC's to limit the number of TC's in a test suite. The hybrid approach is a combination of TCS and TCP. Different prioritization criteria might be connected to the regression test suite to meet a specific measure. TC's can be prioritized as far as random, optimal, statement coverage total or additional, branch coverage total or additional, failure rates, or total fault, exposing the TC's potential [14].

The ACO method fundamentally depends on the insightful strategy utilized by the ants to discover the food source [15]. ACO is an optimization algorithm for distinguishing proof of optimal balance between different factors to limit the search space in an inevitable derivation of the final solution. It is a metaheuristic swarm intelligent-based tool that imitates the conduct of ants searching for sustenance. Ants can discover the optimized way to specific nourishment source from their home by indirect interaction with every other individual from their colony, for example, sending a message to all other members of their colony through environment [16]. This marvel is known as stigmergy. The decision procedure of ants is impacted by the pheromone trails saved by other ants on the ground since ants proceed onward the way, which has a more elevated amount of pheromone when contrasted with different methods.

This paper proposes a framework for an m-ACO, which eases the software testers to perform their SRT.m-ACO,is a changed ACO structure.When real ants discover a nourishment source, it returns with sustenance to their nest, store the food in the nest and afterward go to a similar nourishment source again until the nourishment source wraps up.
On the off chance that that way is most brief, at that point, inevitably, all ants will begin following a similar trail until that sustenance source wraps up and it declines the diversity of the kept nourishment in a given time.The previous studies only focused on developing an algorithm without a proper framework explanation, and the developed algorithm does not get implemented well. Thus, this paper will help researchers to have an appropriate view of the Framework for the m-ACO algorithm, which helps prioritize the TC's using the Average Percentage Faults Detection (APFD) metric. It has been implemented for a project in a startup company Zasta Infotek.

3 Methodology

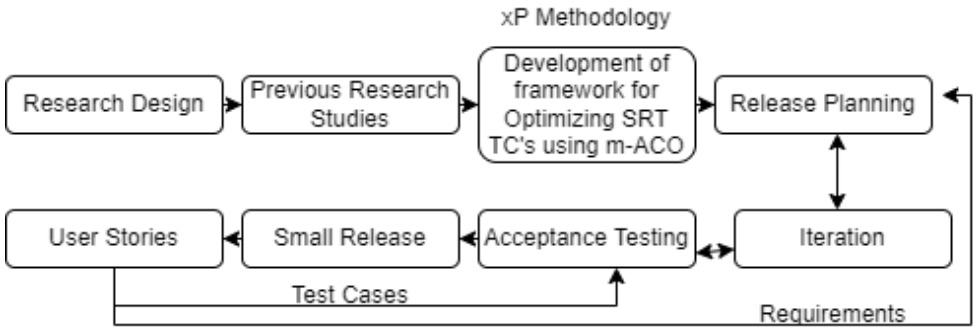

Fig. 1. Research Methodology

Figure 1 depicts the complete diagram of the research methodology, which starts from designing the research to completing the development of the proposed m-ACO framework. The current research was conducted in a few phases, as detailed in the following:

1. The first phase includes studying current RT problems, previous research studies-related to the proposed framework, gathering information on SRT, TCP, and forming the Literature Review.
2. The second phase includes collecting data to develop the proposed framework by conducting an unstructured interview with ST Project Manager from startup company Zasta Infotek Private Limited based from India.
3. The third phase includes developing the proposed framework for optimizing SRT TC's using the M-ACOa mechanism using XP methodology.
4. The fourth phase includes testing the proposed framework through acceptance testing by performing experiment analysis for test suites developed for a web page called Zasta Billing Web Application.
5. The fifth phase includes the discussion of the results obtained from the experiment analysis and the formulation of both the conclusion and recommendation

The type of research methodology that has been used is the quantitative research method which is experimental research.It has been conducted for Zasta Infotek Private Limited, India, and tested for a project called Zasta Billing Web Application(ZBWA). ZBWA is an online invoicing application for small businesses, freelancers, and accountants. The framework has been developed using Angular-typescript. Next is the use of Firebase. It helps to store the database for the framework. A laptop/PC with at least 4GB RAM has been used to develop this framework.

4 Results

The framework has been successfully developed. The result has been collected by comparing the time taken for SRT conducted for Project ZBWA in Zasta Infotek Private Limited, based in India.SRT has been carried out for ZBWA using the developed framework and compared the time taken for SRT with the M-ACO framework and without it. The time

to complete the SRT using the M-ACO framework is 45 days, whereas without M-ACO is 90 days. The screenshots explain the main framework functionalities.

The testers can upload, view, and directly calculate the time taken by uploading the test case report by filling up the necessary columns. The tester has two options to choose for the upload type. When the tester chooses the new upload type, they can fill up the report name, select the iteration number, and upload their report. Once the details have been filled in, they can submit them to prioritize their TC's. When the tester chooses the existing upload type, the tester selects the report name, the iteration number and uploads their document. Once the details have been filled in, they can submit them to prioritize their TC's. The tester needs to fill up some mandatory information in the template such as TC number, description, priority, type, status, created date, and execution time. The tester can upload either new or existing TC documents with the stated mandatory fills as above.

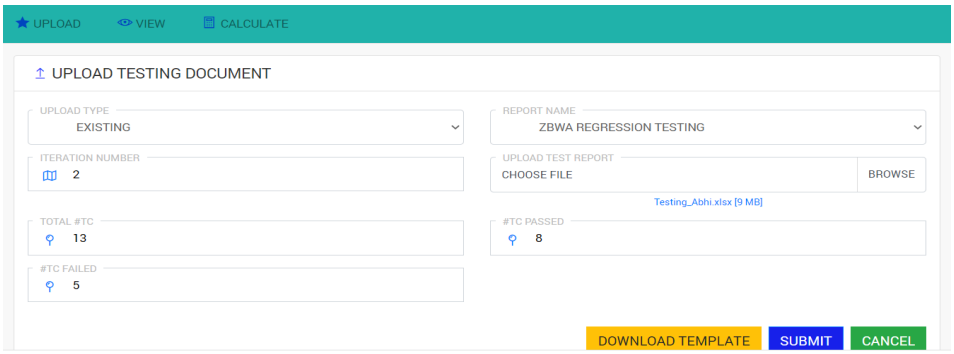

Fig.2. Example of updating existing document

When the tester chooses an existing TC's document, the iteration number, Total TC's, TC passes, and failure will be automatically filled according to the TC document that has been uploaded. After all the necessary details are filled in, the tester can submit the details. The tester can search and view theTC report that has been submitted in the view component. To calculate the priority, the tester needs to choose theTC report submitted previously.

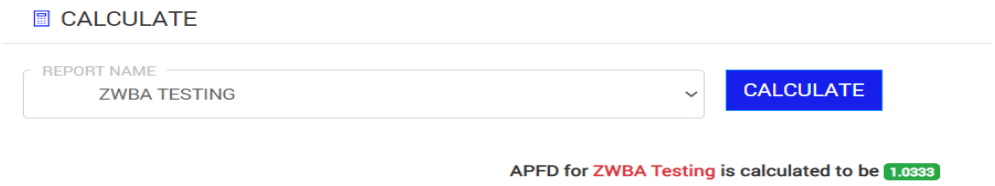

Fig.3. Test Case Priority Calculation

The priority determined based on time-dependent rate computation was utilized as an approval boundary. The m-ACO strategy for TCP determined the appropriateness of every node based on optimal values of the code covered, faults detected, and time utilized for execution. A new report with a prioritized list will be generated. The artificial ants in a pseudo-random way moved toward the fault-carrying modules and determined the appropriateness. A pheromone factor was scored to draw in other antlike measures. Just those pheromone trails that had a place with the most practical code module became more grounded in later cycles by progressive appearances of more antlike actions. The pheromone trail of

less-visited modules became more vulnerable by a consistent rate. Ants might, in any case, visit these progressively weakening paths, however, with lower likelihood. Two parameters were determined: Average percentage of faults detected and Percentage of Test Suite Required for Complete Fault Coverage (PTR). The primary measure utilized for focusing on test cases in this proposed m-ACO method depended on accomplishing total or most excellent fault coverage. Then the outcomes were investigated utilizing the APFD metric [17]. The APFD metric is used to assess the average fault detection rate per test suite execution level.

$$APFD = \frac{1 - TF1 + TF2 + TF3 \ldots\ldots TFm}{mn} + \frac{1}{2} \quad (1)$$

The notation for APFD calculations are,T: test suite to be evaluated,m :number of faults in an application under test,n:total test cases in a test suite,TFj: location of the first test case in test suite T that reveals fault j.

5 Conclusion

SRT is the confirmation of beforehand functional software regarding whether it remains so after a change. It is a time and money concentrated interaction.TCP is beneficial and a sensible RT procedure to limit the number of TC's in a test suite by rearranging TC's in need request, which viable builds the likelihood of code coverage and pace of fault discovery through the APFD metric. This paper presented a framework for a new algorithm, m-ACO, an altered form of ACO for the TCP. m-ACO works by changing the procedure utilized by regular subterranean insects to determine their food source. Research has been conducted for previous m-ACO studies, and this paper has come up with a user-friendly framework for m-ACO, which will help the software testers prioritize the TC's very quickly. Also, previous studies only focused on generating the algorithm and executing it in a small SRT environment. But, in this study, the proposed algorithm has been implemented for a SRT Project for a startup company.The future work on the paper will focus on developing the framework as a mobile application to make the testers ease their testing activities.

References

1. Iboyi, V, *Software Quality: A Case for Regression Testing,* (2020)
2. Kaur, S, *A Review of Software Development Life Cycle Models*., International journal of advanced research in computer science and software engineering **5(11)**: 354-360,(2015).
3. Dahiya, O. and K. Solanki,*A Systematic Literature Study of Regression Test Case Prioritization Approaches,* International Journal of Engineering & Technology **7(4)**: 2184-2191, (2018).
4. Bajwa, J. K. and R. Kaur, *An Adaptive Approach for Test Case Prioritization in Regression Testing using Improved Genetic Algorithm*, (2017).
5. Dalal, S. and K. Solanki,*Challenges of Regression Testing: A pragmatic perspective,*International Journal of Advanced Research in Computer Science **9(1),**(2018)

6. Greca, R.,*Orchestration Strategies for Regression Testing of Evolving Software Systems* ,(2020).
7. Gao, D., et al.,*Test case Prioritization for Regression Testing based on Ant Colony Optimization,*2015 6th IEEE international conference on software engineering and service science (ICSESS), IEEE,(2015).
8. Khari, M., et al.,*Test Optimisation: An Approach based on Modified Algorithm for Software Network,*International Journal of Advanced Intelligence Paradigms **17(3-4)**: 208-237,(2020).
9. Hasnain, M., et al.,*Functional Requirement-Based Test Case Prioritization in Regression Testing: A Systematic Literature Review,*SN Computer Science **2(6)**: 1-32, (2021).
10. Ali, M. and U. Khan, *A Survey Paper, Test Cases Prioritization of Regression Testing*, (2021).
11. Kiran, R.,S..,*A literature survey on TCP-test case prioritization using the RT-regression techniques,* Global Journal of research in engineering, (2015)
12. Kumar, A. and K. Singh,*A Literature Survey on Test Case Prioritization.*" Compusoft **3(5)**: 793.,(2014).
13. Kaur, A., *An Approach to Extract Optimal Test Cases Using AI*. 2020 10th International Conference on Cloud Computing, Data Science & Engineering, IEEE. (2015)
14. Kumar, S. and P. Ranjan,*ACO based test case prioritization for fault detection in maintenance phase,*International Journal of Applied Engineering Research **12(16)**: 5578-5586.,(2017).
15. Alkawaz, M.,H. and A. Silvarajoo, *A survey on test case prioritization and optimization techniques in software regression testing,*2019 IEEE 7th Conference on Systems, Process and Control (ICSPC), IEEE,(2019).
16. Solanki, K., et al.,*Impact of m-ACO-Based Test Suite Prioritization on Software Quality,* Applications of Advanced Computing in Systems, Springer: 289-297.,(2021)
17. Solanki, K., et al.,. *Test case prioritization: an approach based on modified ant colony optimization (m-ACO).*2015 International Conference on Computer, Communication and Control (IC4), IEEE.,(2015).
18. Solanki, K., et al.Test *case prioritization: An approach based on modified ant colony optimization*. Emerging Research in Computing, Information, Communication and Applications, Springer: 213-223.,(2016).