INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 2)

Available online at: www.ijariit.com

Enhanced test case prioritization technique using bat algorithm

Anku Sharma
its.anku170@gmail.com
Baddi University of Emerging Sciences and Technologies, Baddi, Himachal Pradesh

Nancy Sehgal
nancy.sehgal@baddiuniv.ac.in
Baddi University of Emerging Sciences and Technologies, Baddi, Himachal Pradesh

ABSTRACT

Software testing is the method of recognising the accuracy and excellence of software package. The main idea is to identify whether the software fulfils the exact desires, requirements and hopes of the customer or we can also say that testing is the execution of a system or application in order to discover software defect, errors or bugs. Regression testing is a type of software testing that guarantees that earlier established and tested software still executes the same way after it is altered or interfaced with other software. Variations may contain software improvements, configuration changes, etc. Throughout regression testing, new software regressions or bugs may be discovered. Sometimes a software change-impact analysis is achieved to decide which areas could be affected by the projected changes. The main idea of regression testing is to guarantee that alterations made to software, such as addition of new features or modifying present features, have not unfavourably affected features of the software that should not change. Most exclusive activities that take place as software is established and preserved is the retesting of the software when it is modified. Since regression testing is essential, but costly, considerable research has been achieved to develop techniques to make regression testing more operative and efficient. There may be inadequate resources to permit for the re-execution of all test cases in regression testing. In this condition, test case prioritization techniques object to increase the efficiency of regression testing by assembling the test cases so that the most valuable are performed first .Various nature-inspired algorithms were introduced which are Ant colony optimization, Particle swarm optimization, Greedy algorithms ,Local beam search and Bat algorithm. Results shows that the performance of Bat algorithm is best out of all algorithms.

Keywords: *Regression testing, Test case prioritization and Bat algorithm.*

1. INTRODUCTION

The technique of finding fault/s in a software application or program so that the application performs according to the end user's requirement is called software testing .It is the method of confirming a system with the purpose of detecting any errors or absent requirement versus the real requirement. Software testing is generally divided into two types - functional testing and non-functional testing. Software testing is the method of calculating a software item to identify alterations between given input and estimated output .Testing measures the value of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

1.1 Testing is done in two types:

- **Manual testing** contains testing a software manually, i.e., without using any automated tool .In this type, the tester takes over the character of an end-user and tests the software to classify any sudden bug.
- **Automation testing**, also known as Test Automation, is when the tester composes scripts and customs alternative software to test the product. This process contains automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

1.2 Methods of software testing:

- **Black box testing** is a technique of software testing that observes the functionality of an application without looking into its interior structures. This technique of test can be beneficial to every level of software testing: unit, integration, system, and acceptance.

- **White box testing** is the full analysis of interior logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to achieve white-box testing on an application, a tester needs to know the interior mechanisms of the code.
- **Grey box testing** is a method to test the application with taking a some degree of information of the interior working of an application. In software testing, the expression the more you know, the better brings a lot of weight while testing an application.

1.3 Regression testing

It is the practise of testing variations to computer programs to make guaranteed that the previous programming still works with the new changes .Changes may include software enhancement, patches, configuration change, etc. It is a standard part of the program development process. Regression Testing is well-defined as a kind of software testing to check that a current program or code change has not affected existing features .It is full or partial selection of previously executed test cases which are re-executed to confirm existing functionalities work well .Test cases which are already developed run against the new version before a new version of a software product is released, to make sure that all the old abilities still work. The reason they might not work is due to changing or adding new code to a program can certainly present errors into code that is not projected to be changed.

1.3.1 Need of regression

- Regression testing increases our possibility of finding bugs made by changes to a software and application.
- It also detects unwanted side caused always by altering the operating environment.
- The set of regression test is greatly valuable for a new way of performing integration testing. This new mode is relatively faster and little unclear than the old way of doing integration testing- but one always need some sort of set of regression test to do it.

1.3.2 Regression testing techniques:

- Retest All
- Regression Test Selection
- Test Case Prioritization

1.3.2.1 Retest All

In this technique, the complete existing group of test cases is re-executed.This process includes vast time and resources which makes it very expensive.

1.3.2.2 Regression Test Selection

In this technique of regression testing, instead of picking the whole test-suite for re-execution, a portion of the test-suite is selected for execution. In this, the test cases are categorized under reusable test cases and outdated test cases. The re-usable test cases are used in the scheduled regression cycles, whereas out-dated test cases are not used in successive cycles.

1.3.2.3 Test case prioritization

Test case prioritization techniques list test cases for the process in an order that tries to increase their usefulness in meeting some performance goal. Various objectives are possible; one includes the rate of fault detection --- a degree of how rapidly faults are noticed within the testing process. An enhanced rate of fault detection during testing can offer a faster response on the system under test, and let software engineers begin modifying faults earlier than might otherwise be possible.

Definition: The Test Case Prioritization Problem: Given: T, a test suite, PT, the set of permutations of T, and f, a function from PT to the real numbers.

Problem: Find T'ϵ PT such that (T'') (T $\forall$ PT) (T''$\neq$T') [f(T')$\geq$f(T'')].

In this definition, PT signifies the set of all possible prioritizations of T, and f is a function that, applied to any such ordering, yields an award value for that ordering. (For simplicity, and without loss of generality, the definition assumes that higher award values are preferable to lower ones.

Table 1: Test case prioritization techniques

Label	*Mnemonic*	*Description*
T1	random	randomized ordering
T2	optimal	ordered to optimize rate of fault detection
T3	st-total	prioritize on coverage of statements
T4	st-addtl	prioritize on coverage of statements not yet covered
T5	st-fep-total	prioritize on probability of exposing faults
T6	st-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
T7	fn-total	prioritize on coverage of functions
T8	fn-addtl	prioritize on coverage of functions not yet covered
T9	fn-fep-total	prioritize on probability of exposing faults
T10	fn-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
T11	fn-fi-total	prioritize on probability of fault existence
T12	fn-fi-addtl	prioritize on probability of fault existence, adjusted to consider previous test cases
T13	fn-fi-fep-total	prioritize on combined probabilities of fault existence and fault exposure
T14	fn-fi-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted on previous coverage
T15	fn-diff-total	prioritize on probability of fault existence
T16	fn-diff-addtl	prioritize on probability of fault existence, adjusted to consider previous test cases
T17	fn-diff-fep-total	prioritize on combined probabilities of fault existence and fault exposure
T18	fn-diff-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted on previous coverage

1.4 Bat algorithm

The **Bat algorithm** is a metaheuristic algorithm for global optimization. It was motivated by the echolocation performance of microbats, with fluctuating pulse rates of emission and loudness .The Bat algorithm was developed by Xin-She Yang in 2010.

1.4.1 Behaviour of bats

Bats are attractive animals. They are the individual mammals with wings and they also have advanced ability of echolocation. It is predictable that there are about 996 unlike species which account for up to 20% of all mammal species . Their size series from the tiny bumblebee bat (of about 1.5 to 2g) to the giant bats with wingspan of about 2 m and weight up to about 1 kg. Microbats classically have forearm length of about 2.2 to 11cm. Most bats practises echolocation to a certain grade; among all the species, microbats are a well-known example as microbats use echolocation generally while megabats do not.Microbats use a type of sonar, called, echolocation, to detect prey, avoid obstacles, and find their roosting crevices in the dark. These bats emit a very loud sound pulse and attend for the echo that bounces back from the surrounding objects. Their pulses differ in belongings and can be linked with their hunting strategies, dependent on the species. Most bats use short, frequency-modulated signals to sweep through about an octave, while others more often use constant-frequency signals for echolocation. Their signal bandwidth differs depends on the species, and often enlarged by using more harmonics.

If we idealize some of the echolocation appearances of microbats, we can advance various bat-inspired algorithms or bat algorithms. For simplicity, we now use the following estimated or flawless rules:

- All bats use echolocation to sense distance, and they also 'know' the alteration between food/prey and background barriers in some magical way;
- Bats fly casually with velocity vi at position xi with a secure frequency fmin, varying wavelength λ and loudness A0 to search for prey. They can routinely adjust the wavelength (or frequency) of their emitted pulses and adjust the degree of pulse emission r ∈ [0, 1], depending on the proximity of their target;
- Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A0 to a minimum constant value Amin.

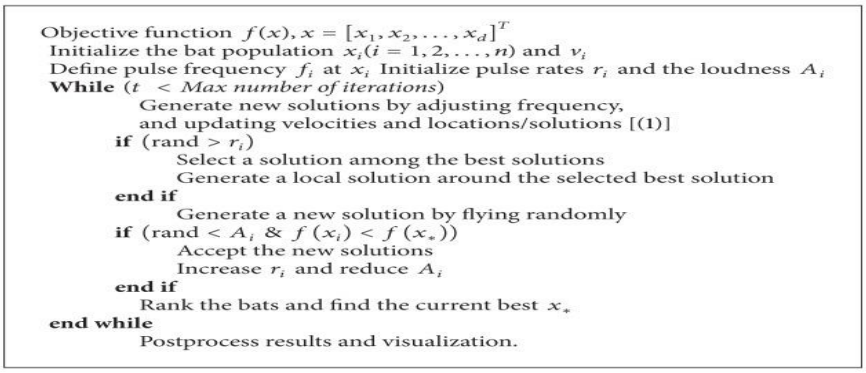

Figure 1: Bat Algorithm

2. LITERATURE REVIEW

Muhammed Maruf Öztürk, et.al (2017) planned in this paper an advanced bat-inspired test cases prioritization algorithm (BITCP).There are four different approaches involved here in order to display the comparison between the planned and existing methods. It is seen through the results reached that the proposed BITCP approach performs better than the existing methods. [12].

Paolo Tonella et al (2006), Test case prioritization having a positive effect on testing cost especially when the test execution time is long.In this paper a new test case prioritization technique is described which take advantage of user knowledge through machine learning algorithm, case-based Ranking (CBR).This technique provide a prioritized test case ordering at low cost. Other big advantage of this approach is that test suit size is moderate at or below 60 test cases per suit which also minimum cost [13].

Zheng Li et al(2007), This paper represent five algorithms for the sequencing.The data and analysis described in this paper indicate that the Greedy Algorithm works much inferior to Additional Greedy, 2-Optimal, and Genetic Algorithms. Also, the 2-Optimal Algorithm eliminates the weakness of the Greedy Algorithm and Additional Greedy Algorithm.[14]
Praveen ranjan srivastava (2008), In this paper a new test case prioritization algorithm is presented which recover the performance of regression testing. This test case prioritization algorithm calculates the average faults found per minute. The main objective of this paper is to determine the effectiveness of prioritized and non-prioritized case with help of APFD [15]

Ruchika Malhotra et al (june 2010), The technique purposed in this paper is important due to the following reasons:

- The proposed technique rises confidence in the correctness of the modified program.
- The test cases selected using the proposed technique will identify and locate errors in the modified program easily.

Hyunsook Do et al (2010), This paper reveals the effect of time constraint on test case prioritization. Time constraints become a reason for improved prioritization techniques and improved maintenance and testing processes.[18].

Gary Y. Chen et al (2010)**,** This paper offers a framework which automates the process of test case arrangement and management through an optimization method. This framework will support software test engineers to solve problems related to prioritizing test cases. In this paper ant, colony optimization technique is used for ordering the test case [19].

Shifa-e-Zehra Haidry et al(2012), Some existing test case prioritization technique reflect that test can be run in any order. But due to functional dependency that exist between the test case means the one test case must be performed before another. This paper presents test case prioritization techniques that use dependency information from test suit to prioritize the test suit. In open dependency the test case t1 must be executed at some point before t2. But it is not essential to execute immediately before t2, but on the other side in closed dependency structure the dependency between test case t1 and t2 is specified means that t1 must be executed before the t2. The test case that has more dependents the coverage value is higher of those test cases. This paper offers a solution to test case prioritization problem when dependency exists between the test cases [20].

3. CONCLUSION AND FUTURE SCOPE

BA customs echolocation and frequency tuning to solve problems. Though echolocation is not directly used to mimic the true function in reality, frequency variations are used. This capability can provide some functionality that may be similar to the key feature used in particle swarm optimization and harmony search. Therefore, BA possess the advantages of other swarm-intelligence-based algorithms. BA has a capability of automatically zooming into a region where promising solutions have been found .It has a quick convergence rate, at least at early stages of the iterations, compared with other algorithms. BA uses parameter control, which can vary the values of parameters (A and r) as the iterations proceed. This provides a way to automatically switch from exploration to exploitation when the optimal solution is approaching.
We can improve test case prioritization in terms of APFD, accuracy and time.

4. REFERENCES

[1] Timothy C. Lethbridge, Robert Langaniere, “Object-oriented software engineering practical software development using UML and Java” 2010, Tata McGraw Hill Eduction Private Limited.
[2] Gregg Rothermel, Roland Huntch, Chengyun Chu, “prioritize the test case for Regression testing”, 2001, IEEE Transactions on Software Engineering.
[3] Hema Srikanthi, Laurie williamsi, Jason Osborne, “system test case prioritization of new regression test case”, 2011, Elsevier B.V.
[4] Sebastian Elbaum, Gregg Rothermel,y Satya Kanduri,z Alexey G. Malishevsk” Selecting a Cost-Effective Test Case Prioritization Technique”, 2004, research gate ACM publications, volume 4, issue 10
[5] Lingming Zhang, Ji Zhou, Dan Hao, Lu Zhang, Hong Mei, “Prioritizing JUnit Test Cases in Absence of Coverage Information”, 2009, IEEE .
[6] S.raju , G. V. Uma, “Factors Oriented Test Case Prioritization Technique in Regression Testing using Genetic Algorithm”, 2012, European Journal of Scientific Research ISSN 1450-216X Vol.74 No.3.
[7] Ryan Carlson, Hyunsook Do, Anne Dento, “A Clustering Approach to Conference on Software Maintenance (ICSM)”, 2011, Science direct publications
[8] Bora, T.C., dos Coelho, L.S., Lebensztajn, L.: Bat-inspired optimization approach for the brushless DC wheel motor problem. IEEE Trans. Magn. 48, 947–950 (2012)
[9] Hasanebi, O., Teke, T., Pekcan, O.: A bat-inspired algorithm for structural optimization. Comput. Struct. 128, 7790 (2013)
[10] Ayari, K., Bouktif, S., Antoniol, G.: Automatic mutation test input data generation via ant colony. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation—GECCO 07, p. 1074. ACM Press, New York (2007)

[11] Biswas, S., Kaiser,M.S., Mamun, S.A.: Applying Ant Colony Optimization in software testing to generate prioritized optimal path and test data. In: 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), p.16. IEEE (2015)
[12] Muhammed Maruf Öztürk, "A bat-inspired algorithm for prioritizing test cases", 2017, Springer, Vietnam J Comput Sci.
[13] Paolo Tonella, Paolo Avesani, Angelo Susi, "Using the Case-Based Ranking Methodology for Test Case Prioritization", 2009, 22nd IEEE International Conference on Software Maintenance (ICSM'06).
[14] Zheng Li, Mark Harman, and Robert M. Hierons, "Search Algorithms for Regression Test Case Prioritization", 2010, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. 33, No. 4, APRIL 2007.
[15] Praveen Ranjan Srivastava, "TEST CASE PRIORITIZATION" 2008, Journal of Theoretical and Applied Information Technology.
[16] Ruchika Malhotra, Arvinder Kaur and Yogesh Singh, "A Regression Test Selection and Prioritization Technique" 2010, Journal of Information Processing Systems, Vol.6, No.2
[17] Siripong Roongrungsuwan, Jirapun Daengdej, "TEST CASE PRIORITIZATION TECHNIQUES", 2010, Journal of Theoretical and applied information Technology, JATT&LLS.
[18] Hyunsook Do, Siavash Mirarab, landan Tahvildari, and Gregg Rothermel, "The effect of time constraint on test case prioritization" 2010, IEEE TRANSACTION ON SOFTWARE ENGINEERING ,VOL.36
[19] GRAY Y.CHENL, JAMIE ROGERS, "Arranging software test case through an optimization method", 2010, IEEE Chung Yuan Christian University, Industrial & Systems. Engineering, Chung Li, Taiwan University of Texas – Arlington, Industrial & Mfg. Systems Engineering, Arlington, TX - USA.
[20] Shifa-e-Zehra Haidry and Tim Miller, "Using Dependency Structures for Prioritisation of Functional Test Suites", 2012, IEEE.

