Journal of Positive School Psychology http://journalppw.com
2022, Vol. 6, No. 4, 3200 – 3206

REVIEW: SOFT COMPUTING BASED RECENT STRATEGIES, TOOLS & TRENDS FOR THE OPTIMIZATION OF TEST SUITE

[1]Kanika Budhwar, [2]Pardeep Kumar Bhatia, [3]O.P Sangwan

[1]PhD Research Scholar, CSE, GJUS&T, Hisar, kanikabudhwar@gmail.com
[23]Professor, CSE, GJUS&T, Hisar

Abstract

Verification of the functionality of software under the constraints of user requirements may be termed as software testing that is quite challenging activity associated with the process of software development. Dynamic changes may be applied to developed software thus may lead to the regression testing that should be conducted with minimal set of test cases but it suffers from various issues i.e. test suite size, test case selection and priority wise its execution etc. In this paper, various concerns and remedies correlated with test suite optimization will be explored.

Keywords: SoftwareTesting, Test Case Optimization, Soft Computing, Regression Testing.

1. INTRODUCTION

Software Testing is used to ensure that whether the developed solution fulfill the client's requirement or not as well as it is also conducted to trace out the bugs. Software upgradation raises the need to verify its effects over the existing functionality and this goal is achieved using regression testing. Following things can be ensured by regression testing under the constraints of software updation as shown in Figure 1:

- Bug Fixing
- Quality Assurance
- Software Reliability

Figure1: *Software Quality assurance by Regression Testing*

Using regression testing, modification in software may be reversed, a particular test case can be selected and as per requirement, priority of a test case can be set, as shown in Figure 2.

Figure 2: *Regression Testing Types*

At least suite can be planned(having number of test cases) to validate the regular variations in software built may increase the overall testing and development cost as well as redundant test cases may be introduced. So there is need to reduce the size test suite to overcome from all these issues.

II. NEED FOR TEST CASE OPTIMIZATION

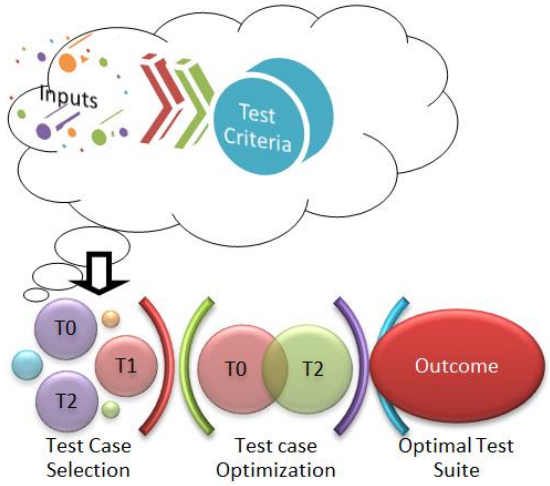

Figure 3: Activities associated with Test Suite

Software may be updated in continue manner and as per each update cycle, there may be requirement to generate a new test case. Later on stages, there will be a bundle of test cases in a specific test suite and it will raise the questions i.e. how to:

- Minimize of number of test cases?
- Select a particular test case, as per requirement?
- Execute test cases, priority wise?

A test suite may be designed for above scenarios that may intake single/multiple inputs and Figure 3 shows critical activities associated with a test suite i.e. Test case selection/ optimization and obtaining an optimal test suite [1-5] etc.

III. SOFT COMPUTING BASED APPROACHES

Soft Computing deals with the intelligent methods those can be used to solve the problems in real-time environment those are given below [31]:

- Artificial Neural Networks(ANN)
- Fuzzy Logic
- Probabilistic Reasoning
- Evolutionary Computation
- Machine Learning(ML)
- Support Vector Machines(SVM)
- Bayesian statistics

Next section IV highlights the contribution of researchers in relevant field using common approaches (discussed above).

IV. LITERATURE SURVEY

K. W. Al-Sabbagh et al. [6] introduced a noise optimization framework to reduce the inconsistent training modules used by Big data platform. It uses machine learning approach with selective regression testing and performs classification over input training sets (curated & non-curated). Finally, its outcome is evaluated for curated sets on the basis of various parameters i.e. F-Score/precision value/ Recall in contrast of non-curated sets. Experimental results indicate that it improves the overall system performance by predicting the failure of test cases.

D. Marijan et al. [7] introduced a perdition model to define the priority of regression test cases. It uses training and learning strategies both help to condense the test case selection interval for prioritization. Analysis shows that it outperforms in terms of effective fault detection as compared to traditional regression testing. It can be enhanced to optimize the test case redundancy.

Y. Chen et al. [8] developed a deep neural network based scheme for test case selection. It

takes number of test cases as input and uses greedy approach for test case selection/prioritization on the basis of their behavior patterns. Analysis shows its performance in terms of effective fault detection as compared to random techniques.

S. Rathaur et al. [9] introduced a machine learning based linear regression scheme to track the bug density in software. Software metric (Line of code/no. of programmers & code commits etc.) is prepared and predicates are used to analyze bug density w.r.t. this metric. Analysis shows that it has higher prediction accuracy for bug detection over decision tree regression scheme.

D. S. Silva et al. [10] proposed a test case prioritization method using the combination of ant colony algorithm with fuzzy logic. It considers various parameters (i.e. execution interval/no. of detected faults/criticality etc.). for performance analysis of test suites and finally, these are selected using greedy approach. Experiments show that it is compatible with the traditional regression testing and it can be also utilized to define the relationship between software metric and fault occurrence.

A. Thakur et al. [11] presented a scheme for regression testing that performs automated slicing over test suites to set priority of test cases. Analysis shows that it offers in terms of higher fault detection/optimal execution interval as compared to manual test case prioritization.

M. Azam et al. [12] developed a automated test case generation/prioritization scheme by merging Genetic Algorithm (GA) with fuzzy logic framework. GA defines the boundary values and partitioning whereas test case prioritization is done by fuzzy logic. Analysis shows that it is more efficient and accurate as compared to traditional testing tool (Selenium).

A. S. Verma et al. [13] presented a test case optimization method using butterfly algorithm that considers fault detection ratio as performance parameter to minimize the size of a specificregression test suite. Comparison study with BAT methods shows that it outperforms in terms of optimal execution time/ cost/ /efforts etc. It can also be utilized for prioritization of test cases.

T. Tan et al. [14] implemented a K-means based method for test case optimization. It also uses fuzzy membership function to select the unique test cases from a cluster. Results shows that it can optimize the redundant test cases as well as offers higher efficiency. It can be further enhanced using GA.

M. Khari[15] did an investigation to find out the various constraints that affect the automated test case generation. Analysis shows that complexity/project duration, cost, size are various factors that are directly associated with the development cost, quality and efforts etc. This analytical data can be further utilized to improve the performance of test suites.

Z. Wang et al. [16] enhanced the existing particle swarm optimization (PSO) algorithm for the automation of test case generation. Experiments show that it can deliver higher performance with optimal time as compared to others (traditional PSO/GA).

S. Sheoran et al. [17] presented an artificial bee colony method to optimize the test case for data flow. It performs path search at global/local level and finally, duplicate paths are eliminated and rest of all paths are verified to locate the errors. Finally, its outcome is used to define the optimal test suite.Analysis shows that it offers optimal size of test suite.

A. Bahrampour et al. [18] enhanced the test suite generation method by merging both GA and PSO algorithms. It offers maximum coverage by refining the search at local/global level thus results in size reduction of test case. Analysis shows that extended solution outperforms in terms of higher fault detection ratio/optimal size/cost.

P. Ramgouda et al. [19] merged the crow search with chaotic fruitfly optimization method for the interactional test suite optimization. Analysis shows its performance in terms of higher coverage size/time.

S. Kassaymeh et al. [20] merged the slap method with neural network for the prediction of bugs in software. Training data sets are used to define weights/biases and finally, an optimizer is used to predict optimal values in given input. Analysis shows it outperforms in terms of higher accuracy/ specificity/sensitivity with optimal error rate etc. as compared to traditional methods.

T. M. Nithya et al. [21] used the combination of simulated annealing (for global search) and gradient descent method (for local search) to optimize the test case selection for regression testing. Analysis shows offers minimal size of test suite with maximum coverage value.

S. Nayak et al. [22]used bee algorithm to demine the test case prioritization. Its size is optimized using fuzzy logic.Comparison with traditional methods of prioritization (Reverse/Randomand no-prioritization) shows that it offers higher fault detection ratio but it is not suitable for large scale input datasets.

H. Pei et al. [23] presented a dynamic random testing solution for the test case prioritization and optimal resource allocation over cloud platform. Test case selection method is used to mark current test case profile and a virtual environment is assigned for execution. It can manage multiple test profiles which are updated as required. Analysis shows that it offers higher fault detection ratio in contrast of traditional method.

A.P. Agrawal et al. [24] used heuristic algorithm for test suite optimization. It selects only consistent test cases for regression testing. Its performance was examined with numerous parameters (execution time/ fault coverage /reduced optimized test suite size) as compared to other methods (Greedy/ Additional Greedy/Harrolds–Gupta–Soffatest suiteetc.).

W. Liu et al. [25] proposed ant colony based test reduction method that forms a virtual colony in which each case is added as node and a search is performed using pheromone values and finally, optimal set of test cases are selected. Experiments show it can minimize the testing cost.

O. Banias[26] provided a test case optimization solution using dynamic programing. Test cases prioritization is done randomly using test pool to locate maximum faults in software. Analysis indicates that it is suitable only for medium size projects and it suffers from memory space utilization and it is still an open issue.

A. Bajaj et l. [27] used bio inspired method to define test case prioritization and its outcome is fed to adaptive algorithm for test case selection and finally, hybrid approach is enforced to filter out in consistence test cases. Analysis shows that it offers maximum coverage with optimal fault coverage loss.

M. N. Abadeh [28] used incremental regression testing for web based applications. It utilizes GA and multi-objective function to search faults. Analysis shows its performance in terms of effective fault detection as compared to traditional regression test suite.

J. Joo et al. [29] introduced a metric based prioritization scheme that performs sorting using a probability of error propagation w.r.t each test case. Its outcomes are used to identify unique test cases along with their fault finding intensity. Analysis shows its efficiency in terms of fault finding ratio w.r.t. metric. However, minimization of redundant test cases is still an open issue for this scheme.

A. Vescan et al. [30] deployed fuzzy logic for the prioritization of test cases. It builds a metric using various parameters (Faults/Execution Time) and only optimal values are adapted for test suiteusing fuzzy clustering. Analysis shows its performance in terms of maximization of faults/code coverage with expectable time execution.

Table:1 *shows the recent development introduced by various researchers in the relevant domain.*

Author(s)	Contribution
K. W. Al-Sabbagh et al. [6]	Machine learning approach with selective regression testing
D. Marijan et al. [7]	Regression test case priority estimation using perdition model
Y. Chen et al. [8]	test case selection using deep neural network
S. Rathaur et al. [9]	Bug density tracing though machine learning based linear regression
D. S. Silva et al. [10]	Prioritization of test case by fuzzy logic based ant colony algorithm
A. Thakur et al. [11]	Prioritization of test case through automated test suite slicing
M. Azam et al.	Genetic Algorithm (GA) with

[12]	fuzzy logic framework for test case automation
A. S. Verma et al. [13]	Butterfly algorithm for test case optimization
T. Tan et al. [14]	K-means based method for test case optimization
M. Khari[15]	Study for automated test case generation
Z. Wang et al. [16]	PSO algorithm for the automation of test case generation
S. Sheoran et al. [17]	Artificial bee colony method to optimize the test case for data flow
A. Bahrampour et al. [18]	Test suite generation method by merging both GA and PSO algorithms
P. Ramgouda et al. [19]	Crow search and chaotic fruitfly optimization method
S. Kassaymeh et al. [20]	Prediction of Software bugs using Slap method and neural network
T. M. Nithya et al. [21]	Test case selection using combination of simulated annealing and gradient descent method
S. Nayak et al. [22]	Test case prioritization using bee algorithm
H. Pei et al. [23]	Dynamic random testing solution for the test case prioritization and optimal resource allocation over cloud platform
A.P. Agrawal et al. [24]	heuristic algorithm for test suite optimization
W. Liu et al. [25]	Ant colony based test reduction
O. Banias [26]	Dynamic programing for test case optimization
A. Bajaj et l. [27]	Test case prioritization
M. N. Abadeh [28]	Incremental regression testing for web based applications
J. Joo et al. [29]	Metric based test case prioritization scheme
A. Vescan et al. [30]	Fuzzy logic for the prioritization of test cases

V. CONCLUSION

Software may be updated frequently to incorporate the new user requirements but these dynamic changes may have impact over the functionality of existing modules, so it arise the need of testing the new features as well as their compatibility with the existing. To ensure the smooth execution of updated application is done through regression testing which has some barriers i.e. test suite size, number of required test case and the sequence in which these tests are executed etc.

Software updation affects over existing functionality, significance of regression testing and its limitations are discussed in this paper. Researchers presented numerous solutions related to test suite optimization/selection and its prioritization which are developed using neural network, GA, PSO, GA, fuzzy logic, dynamic programming, greedy method, butterfly algorithm, prediction model/metric based prioritization, deep neural network based test case selection,machine learning based linear regression scheme to track the bug density, K-means based method, artificial bee colony, crow search with chaotic fruitfly optimization method, simulated annealing (for global search) and gradient descent method, nature inspired methods, heuristic algorithm etc. Currently, the scope of this study covers the open issues associated with regression testing only. In future, investigate will also include the automated testing tools available for other types of software testing (Functional/non-Functional testing).

Reference

[1] M. Khari, P. Kumar, D. Burgos, R. G. Crespo, "Optimized test suites for automated testing using different optimization techniques", Soft Computing, Vol.22, Springer-2018, pp.8341–8352.

[2] M. H. Alkawaz, A. Silvarajoo, "A Survey on Test Case Prioritization and Optimization Techniques in Software Regression Testing", 7th Conference on Systems,

Process and Control (ICSPC), IEEE-2019, pp. 59-64.

[3] A. Kiran, W. H. Butt, M. W. Anwar, F. Azam, B. Maqbool", A Comprehensive Investigation of Modern Test Suite Optimization Trends, Tools and Techniques," in IEEE Access, Vol.7, IEEE-2019, pp.89093-89117.

[4] V. Gupta, Y. Vij, C. Gupta, "Toward Analysis of Requirement Prioritization Based Regression Testing Techniques", System Performance and Management Analytics, SPringer-2018, pp.97-101.

[5] J. F. S. Ouriques, E. G. Cartaxo, P. D. L. Machado, "Test case prioritization techniques for model-based testing: a replicated study", Software Quality Journal, Vol.26, Springer-2018, pp.1451–1482.

[6] K. W. Al-Sabbagh, M. Staron, R. Hebig, W. Meding, "Improving Data Quality for Regression Test Selection by Reducing Annotation Noise", 46thEuromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE-2020, pp. 191-194.

[7] D. Marijan, A. Gotlieb, A. Sapkota, "Neural Network Classification for Improving Continuous Regression Testing", IEEE International Conference On Artificial Intelligence Testing (AITest), IEEE-2020, pp. 123-124.

[8] Y. Chen, Z. Wang, D. Wang, Y. Yao, Z. Chen, "Behavior Pattern-Driven Test Case Selection for Deep Neural Networks", IEEE International Conference On Artificial Intelligence Testing (AITest), Newark, IEEE-2019, pp.89-90.

[9] S. Rathaur, N. Kamath, U. Ghanekar, "Software Defect Density Prediction based on Multiple Linear Regression",2ndInternational Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, IEEE-2020, pp. 434-439.

[10] D. S. Silva, R. Rabelo, P. S. Neto, R. Britto, P. A. Oliveira, "A Test Case Prioritization Approach Based on Software Component Metrics," 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), IEEE-2019, pp. 2939-2945.

[11] A. Thakur, G. Sharma, "Neural Network Based Test Case Prioritization in Software Engineering", International Conference on Advanced Informatics for Computing Research, ICAICR 2018: Advanced Informatics for Computing Research, pp.334-345.

[12] M. Azam, A.urRahman, K. Sultan, S. Dash, S. Naqeeb Khan M. Aftab A. Khan, "Automated Testcase Generation and Prioritization Using GA and FRBS", International Conference on Advanced Informatics for Computing Research, ICAICR : Advanced Informatics for Computing Research, Springer-2018, pp 571-584.

[13] A. S. Verma, A. Choudhary, S. Tiwari, "Test Case Optimization using Butterfly Optimization Algorithm", 10thInternational Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE-2020, pp.704-709.

[14] T. Tan, B. Wang, Y. Tang and X. Zhou, "An Improved K-means Algorithm for Test Case Optimization" ,IEEE 4thInternational Conference on Computer and Communication Systems (ICCCS), IEEE-2019, pp. 169-172.

[15] M. Khari, "Empirical Evaluation of Automated Test Suite Generation and Optimization", Arabian Journal for Science and Engineering, Vol.45, pp.2407–2423.

[16] Z. Wang, Q. Liu, "A Software Test Case Automatic Generation Technology Based on the Modified Particle Swarm Optimization Algorithm", International Conference on Virtual Reality and Intelligent Systems (ICVRIS), IEEE-2018, pp. 156-159.

[17] S. Sheoran, N. Mittal, A. Gelbukh, "Artificial bee colony algorithm in data flow testing for optimal test suite generation", International Journal of System Assurance Engineering and Management, Vol.11, Springer-2020, pp.340–349.

[18] A. Bahrampour, V. Rafe, "Using memetic algorithm for robustness testing of contract-based software models", Artificial Intelligence Review, Vol.54, Springer-2021, pp.877–915.

[19] P. Ramgouda, V. Chandraprakash, "Constraints handling in combinatorial interaction testing using multiobjective crow search and fruitfly optimization", Soft Computing, Vol.23, Springer-2019, pp.2713–2726.

[20] S. Kassaymeh, Salwani Abdullah, M. A. Al-Betar, M. Alweshah, "Salp swarm optimizer for modeling the software fault

prediction problem", Journal of King Saud University - Computer and Information Sciences,Elsevier-2021, pp.1-14.

[21] T. M. Nithya, S. Chitra, "Soft computing-based semi-automated test case selection using gradient-based techniques", Methodologies and Application, Soft Computing Vol.24, pp.12981–12987.

[22] S. Nayak, C. Kumar, S. Tripathi, N. Mohanty, V. Baral, "Regression test optimization and prioritization using Honey Bee optimization algorithm with fuzzy rule base", Soft Computing, Springer-2020, pp.1-20.

[23] H. Pei, B. Yin, M. Xie, "Dynamic Random Testing Strategy for Test Case Optimization in Cloud Environment", IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), IEEE-2018, pp. 148-149.

[24] A. P. Agrawal, A. Choudhary, A. Kaur, H. M. Pandey, "Fault coverage-based test suite optimization method for regression testing: learning from mistakes-based approach", Neural Computing and Applications Vol.32, Springer-2020, pp.7769–7784.

[25] W. Liu, J. Xiong, "Research on Simplified Method of Combination Test Case Set for Basic Software System", IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), IEEE-2020, pp. 718-721.

[26] O. Banias, "Dynamic programming optimization algorithm applied in test case selection", International Symposium on Electronics and Telecommunications (ISETC), IEEE -2018, pp.1-4.

[27] A. Bajaj, O. P. Sangwan, "Tri-level regression testing using nature-inspired algorithms", Innovations in Systems and Software Engineering, Vol. 17, pp.1–16.

[28] M. N. Abadeh, "Genetic-based web regression testing: an ontology-based multi-objective evolutionary framework to auto-regression testing of web applications", Service Oriented Computing and Applications Vol.15, Springer-2021, pp.55–74.

[29] J. Joo, S. Yoo, M. Park, "Poster: Test Case Prioritization Using Error Propagation Probability", IEEE 13th International Conference on Software Testing, Validation and Verification (ICST), IEEE-2020, pp.398-401.

[30] A. Vescan, C. Şerban, "Towards a new Test Case Prioritization Approach based on Fuzzy Clustering Analysis", IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE-2020, pp.786-788.

[31] B. Pandey, R. Jain, "Soft Computing Based Approaches for Software Testing: a Survey", International Journal of Soft Computing and Engineering, Vol.4 (2), IJSCE-2014, pp.1-8.