

A Neoteric Approach of Prioritizing Regression Test Suites Using Hybrid ESDG Models

Punam Rattan¹, Mohit Arora², Manik Rakhra³, Vanshika Goel⁴, Anuj chaudhary⁵,
Aman Mishra⁶, Mohammad Shabaz⁷

¹Department of Computer Applications, CT University, India.

²School of Computer Science and Engineering, Lovely Professional University, Phagwara, India.

E-mail: mohit.15980@lpu.co.in

³School of Computer Science and Engineering, Lovely Professional University, Phagwara, India.

⁴School of Computer Science and Engineering, Lovely Professional University, Phagwara, India.

⁵School of Computer Science and Engineering, Lovely Professional University, Phagwara, India.

⁶School of Computer Science and Engineering, Lovely Professional University, Phagwara, India.

⁷School of Computer Science and Engineering, Lovely Professional University, Phagwara, India.

ABSTRACT

Testing of software is performed during development phase of software development life cycle for detection of bugs or errors in earlier phases and to make sure that modifications made in software change the main specifications of the system. Regression testing in software testing is a keyholder. It is part of software testing which is done in order to make sure that changes or modification don't make any changes with the existing behavior of software. It makes sure that when bugs are fixed or a new functionality is added, it shouldn't affect the original functionality of the software. It plays a crucial role in software development and maintenance. A priority timestamp has been assigned to the suite for enhanced execution. It is used to order the sequence of test cases in a manner that detection of faults at earlier phases is maximized. Extended System Dependence Graphs (ESDG) model with genetic algorithm technique is a proposed method for prioritizing test cases in the area of object-oriented software based on fault detection rate and its impact. APFD metric, used to measure efficacy of test cases, need to optimized in this context. This paper proposes a technique using ESDG model and Genetic Algorithm for regression suite prioritization for object-oriented software and APFD metric measures effectiveness of new test cases. We made an attempt to mine test cases from original test repository, that might be utilized for modified program testing and prioritizing elected regression test cases which increase fault detection rate at earlier stages.

KEYWORDS

Test Case Prioritization, Test Case Generation, ESDG Model, Object Oriented Software.

Introduction

Regression testing is software testing and software maintenance which is performed when changes are made in current version of software. Target of regression testing is checking that changes made in software do not alter behavior of current and rest of the part of software. It is challenging and complex process because of changing trends in software development. It is a style that focuses on retesting after changes are made. Traditionally regression testing reuses the same regression test cases. Test suites are large in size as size of software increases and it became costly to run all test suites. Different techniques are used to increase value of test suite. Minimization techniques eliminate and minimize obsolete test cases to reduce test cases. Selection techniques select tests which are relevant to recent changes. Prioritization techniques schedule and order tests to increase effectiveness to achieve performance. The main performance goal is fault detection rate. Detection of faults must be done in the early stages of testing whole the execution of various test cases. A prioritization metric used which calculates betterment of technique used. This research implements a technique by using ESDG model and Genetic Algorithm for prioritizing test cases for object-oriented software and APFD metric measure effectiveness of new test cases.

Regression test prioritization techniques reorder test cases which are executed after changes in code during testing. It reorders execution of test cases to make sure that defects are detected as soon as possible. At earlier stages reordered test cases are effective for finding faults. Suppose a test suite called as T' has six test cases having order {T₁ T₂, T₃, T₄, T₅, T₆}. Fault detection by T' in P is considered as given in table below.

Table 1. Example Scenario for Regression Test Case Prioritization

Test Case	Number of Faults Detected	Detected Faults	Time Taken (in mins)
T ₁	7	F1, F2, F4, F5, F6, F7, F8	9
T ₂	1	F1	1
T ₃	2	F1, F5	3
T ₄	3	F2, F3, F7	4
T ₅	3	F4, F6, F8	4
T ₆	3	F2, F4, F6	4

Suppose total time is 12 minutes. As it is needed to detect maximum faults at earlier stages it would be better to order test case with consideration of number of faults. For test case order T₁, T₄, T₅, T₆, T₃, T₂ for twelve-minute time constraint only test case T₁ has time to run and can find total of seven faults. Time is main concern and test cases are to be ordered with execution time. Test tuples TC₁ = {T₁}, TC₂ = {T₂, T₃, T₄, T₅}, TC₃ = {T₂, T₁} and TC₄ = {T₅, T₄, T₃} are considered for execution.

In Regression test prioritization technique two things are considered. One is granularity and other is prioritization strategy. Throughout life of large software product number of test cases increases when next release of software comes to market. Retesting cost of test cases is high so testers discard obsolete test cases for constructing reduced test suite at reasonable cost. The scope of study is to propose a technique used for prioritizing test cases for object-oriented software with use of ESDG model and Genetic Algorithm. APFD metric measure effectiveness of test cases generated. ESDG model and Genetic Algorithm are used in the proposed technique to order test case. ESDG Graph is a graph used for object-oriented software and genetic algorithm is search algorithm based on genetic principles. APFD metric is a prioritization metric used to measure rate of detecting faults of regression tests and for calculating the effectiveness of regression test prioritization technique and test cases generated. System dependence graph (SDG) is used to create models of procedural program. Afterwards it was extended to create modes of object-oriented software and given name extended system dependence graph ESDG. ESDG is used to create models for object-oriented program. It is extension of system dependence graph which create model of procedural programs. It is used to represent control and data dependencies and different types of dependencies. Analyzing statements using ESDG model help in identifying changes in simple statements, method call statements and polymorphic method call.

Related Work

Kaur, A. et al. [1] have proposed an approach which incorporates code-coverage based optimization using genetic algorithm. Genetic algorithm is automating test case prioritization and effectiveness of approach is represented with Minimal Acceptable Code Coverage metric. Suman et al. [2] have proposed approach using genetic algorithm for prioritization of regression test cases and this approach dynamically prioritizes test cases on the basis of code coverage completeness. Generation of dewy test cases is performed through the use of PMX with cyclic crossover. Overall aim is at reducing count of test cases lined up for execution when modifications are made in code. Panigrahi, C. et al. [3] have proposed another technique for regression test selection on basis of analysis of program code for object-oriented software and UML diagram for changed classes of object-oriented program. Dependency graph is made for the original program code. On changing the program, updating the model so constructed is done to know changes. This model represents the flow of control and various data dependencies of object-relationships. In order to determine affected model elements due to change in program, forward slice is constructed. Every element of the changed model is considered as a criterion for slicing. The test cases of changed elements of the model present within the program model, those in transitions and those due to the affected methods are selected to perform regression testing on. This technique select average of 28 % extra fault-finding test cases in comparison to dependency graph model approach. It increases 38% of size of test suite for regression testing.

Musa, S. et al. [4] have presented test prioritization technique for regression testing for object- oriented programs which makes use of dependency graph model and determines affected program using genetic algorithm. This model selects optimized test cases. The goal is to find changes from body of method due to dependence on data,

control and object relation and orders them based on their fitness with help of genetic algorithm. The count of statement changed determines fitness of the test case for analysis through regression testing. This approach provides evidence of feasibility of approach and increase fault detection rate and reduce regression testing effort. Mall, R. et al. [5] have proposed a technique for object-oriented software where an intermediate graph model for program is constructed. Model is updated whenever code is modified. Constructed model represents dependencies and information of different types of object-relationship dependencies. The changed nodes in model are determined with construction of forward slices corresponding to each modified element of the model. There are test cases which covers large number of changed elements of model is given more priority. This approach increases APFD metric value by 25.70 % in comparison with other approach. Abu Bakar et al. [6] have another technique concerning object-oriented programs that has dependency graph model responsible and analysis of the affected program is performed through genetic algorithm. This tactic selects test cases on the basis of optimization from test cases. The goal is to find changes from body of method on account of data dependence, control dependence and object related dependence and orders them based on their fitness with help of genetic algorithm. The count of statement changed determines fitness of the test case for analysis through regression testing. This approach provides evidence of feasibility of approach and increase fault detection rate and reduce regression testing effort. The order is measured using APFD metric to calculate effectiveness of approach. This approach provides better efficiency and effectiveness in the results of regression testing. Brian A. Malloy et al. [7] have discussed about representation of the object-oriented programs based on program dependency graph and includes control flow and data flow information of the code. Mishra P, et al. [8] have proposed technique for test case prioritization with the help of genetic algorithm. Their technique separates the test cases which are not worthy and remaining test cases prioritizes subsequence of original test cases such as new test cases are to be executed in constraint of time. It has higher fault identification rate. Srivastava, P. et al. [9] have presented method to optimize software testing efficiency by finding critical path clusters of programs. It is done by development of variable length genetic algorithm which optimizes and choose path clusters that are weighted with path criticality. Parts of program can be tested but they can be error prone. Identification of critical paths increases the testing efficiency. Mor, A. [10] has worked on APFD metric for measuring fault detection rate during prioritization of test cases. Various forms of APFD metric are used. All these metrics used are reviewed to assess fault detection rate for prioritized test cases with respect to coverage criteria for various other techniques. Arora, V. et al. [11] have discussed that graphical methods that works on the structure of the system which helps in testing of logics of the program. Control flow graph sequences the executed instructions of program. System Dependence Graph shows program which has multiple procedures procedural calls. Analysis of flow graphs and dependence graphs is done based on various dependencies. S. Raju et al. [12] have done the development and validation of requirement-based system level test case prioritization method of fault detection at earlier stage. Set of prioritization factors are also designed for proposed system. T. Prem Jacob et al. [13] have discussed that selected of test cases are used to prioritize test cases on basis of count of modified lines that the test case covers. The test case which covers the greatest count in the modified lines is given higher preference thereafter being the first to be executed and then that in which modified lines have the least coverage are given less priority that shall be executed at the last in a manner such that deadline time is not reached. To prioritize test cases genetic algorithm is used which takes as input the information of test cases and produce procession of test case lined up for execution such that maximum modified codes are covered. Baharom, S. et al. [14] have presented test prioritization for object-oriented program for regression testing based on extended system dependency graph model for changed program. This approach is based on selecting test cases by dependency analysis of source codes by using genetic algorithm. The aim is identifying changes in body of method and selects test cases on basis of changed statements and orders them on basis of their fitness with help of genetic algorithm. Number of affected statements determines good test cases for regression testing. This approach increases fault detection rate and reduce regression testing effort in comparison with retest-all technique. Ghani, A et al. [15] have presented techniques for test case prioritization using regression test case in object-oriented programs through the relationship with dependency graph model and utilizing genetic algorithm analyze the affected program. This tactic selects test cases depending on the optimization from test cases. The goal is to find changes from body of method on account of data dependence, control dependence and object related dependence and orders them based on their fitness with help of genetic algorithm. Number of changed statements determines the fitness of test case for regression testing. This approach provides evidence of feasibility of approach and increase fault detection rate and reduce regression testing effort. The order is measured using APFD metric to calculate effectiveness of approach. This approach provides better efficiency and effectiveness in the results of regression testing. Krishnamoorthi, R et al. [16] have proposed test

prioritization technique with help of genetic algorithm. This technique prioritizes subsequences of original test cases in order to make sure that new test cases are executed in constrained of time and will have higher fault detection rate when compared with rates of prioritized test cases. The metric established is Average of the Percentage of Faults Detected (APFD) which determines how effective the fresh test cases are. [17] is chapter which gives time constrained prioritization technique for regression testing proposed with genetic algorithm. In this scheme time allotted for execution and coverage information is considered for all test cases. It is observed that genetic algorithm is used as test generation, program transformation, and software maintenance resource allocation and so it is used to design time constrained test case prioritization. APFD metric and ATEI metric are used to determine effectiveness of new test cases. Donglin Liang et al. [18] have presented SDG for object-oriented program which is more precise as compared to other representations and is more efficient to construct as compared with other approaches. SDG help the different objects by distinguishing data members for them and provide path to representing object parameters. Owing to such system dependency, the graph concept for object slicing is introduced and algorithm for implementation of such concept. Rahul Gupta et al. [19] have discussed that the purpose of regression testing is to ensure the fixation of bugs while contemporary functionality introduced during the software's upgradation does not affect established functionality inherited before the current version. Regression testing is most complaining activity of software development and maintenance. There are few resources that during regression testing allow the re-execution of all test cases. An algorithm is proposed for prioritizing test cases based on fault detection rate and fault effect. The algorithm recognizes fault being exhaustive during initial testing stage. APFD metric is used to determine efficiency of approach. Praveen Ranjan Srivastava et al. [20] have proposed algorithm for prioritizing test cases which calculates average faults found in a time unit. This algorithm determines effectiveness of prioritized and non-prioritized cases. Chhabi Rani Panigrahi et al. [21] have proposed approach on basis of model to prioritize test cases for object-oriented software's. This model represents different object-oriented features. For prioritization of test cases the dependencies among test case is considered.

Proposed Methodology

The conglomerate methodology is presented in following two sections 3.1 and 3.2. The same has been applied to object-oriented programming paradigms in eclipse IDE and its associated sequence dependency graphs has also been generated.

Proposed Algorithm

The proposed algorithm has been discussed below in three broad categories. In the first step, we will determine the nodes affected because of any critical updates/release changes. Then, apply genetic algorithm followed by prioritization of created test cases.

Determine Affected Nodes

Step 1: ANS \Downarrow NULL

Step 2: For each node

Step 3: Find nodes which are data, control or dependent due to inheritance and association on node

Step 4: ANS \Downarrow all nodes which are data, control or dependent due to inheritance and association on node

Applying Genetic Algorithm

Step 1: Encode selected test cases

Step 2: Generate 2 populations P1 and P2 from T' (Original Population) Step 3: P1 is set T', P2 is reverse set of T.

Step 4: Calculate fitness function of parent Step 5:

Repeat {

 Child=crossover (P1, P2)

 Child=mutate (child)

Step 6: Calculate fitness function for child
 Step 7: Add child to population
 Step 8: Discard worst chromosome from population
 Step 9: } Until maximum number of iterations

Regression Test Prioritization

Step 1: For every node of ANS do
 Step 2: For every test case t, do
 Step 3: Add test case t to T
 Step 4: $WT \leftarrow wt$.
 Step 5: End
 Step 6: End
 Step 7: Sort test cases of T in decreasing order of their weights in WT.

Flowchart

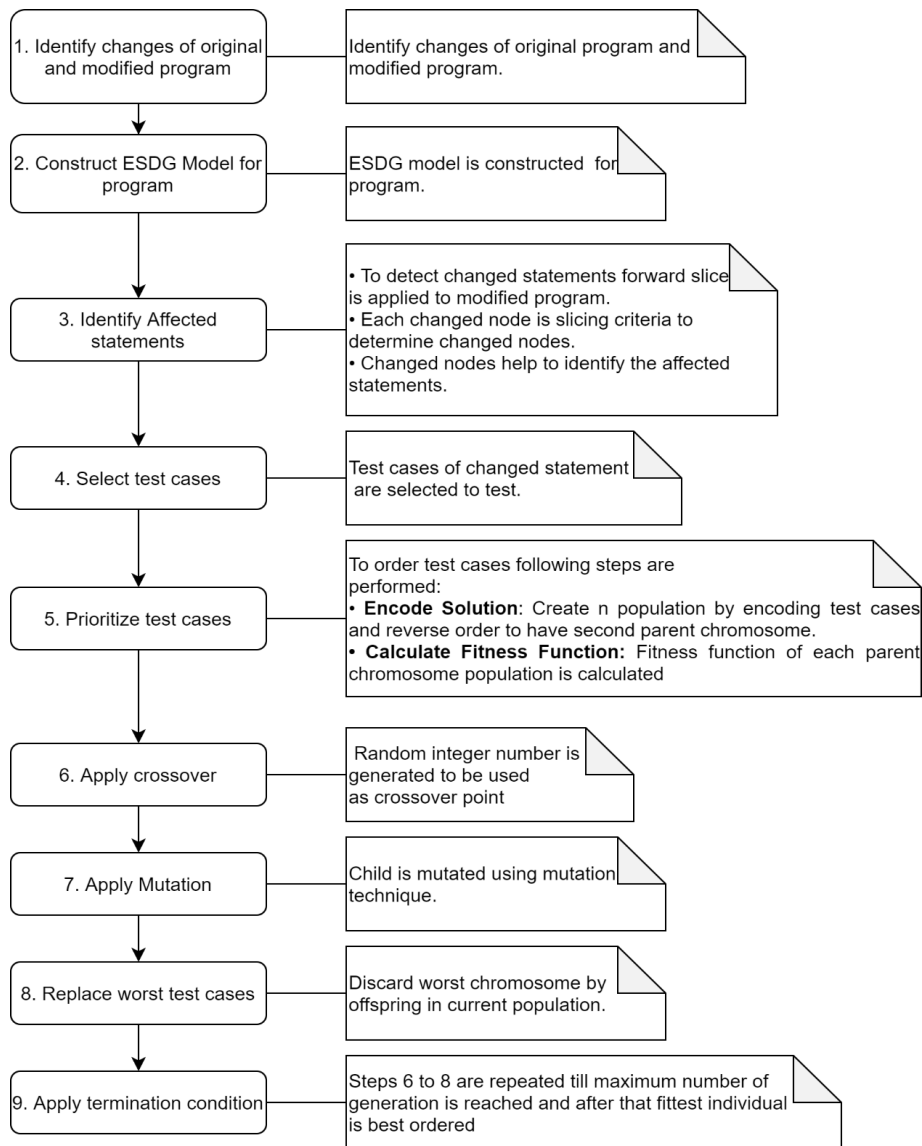


Fig. 1. Proposed methodology flowchart

Results and Discussions

Some sample screenshots of various ESDG graphs that has been generated after applying above methodology are given in Figure 2 and 3. It shows the cyclic and other dependencies between the various lines of code. The numbers assigned to program lines are vertices of graph.

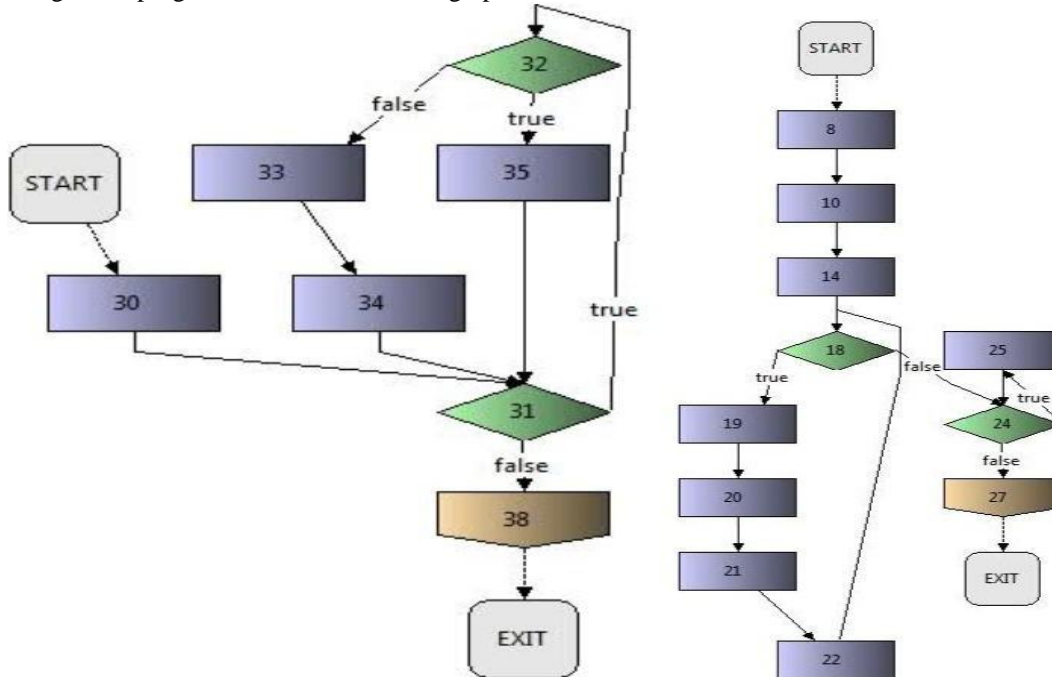


Fig. 2. ESDG model graph-1

The figure 3, given beneath is one example amongst 16 ESD graphs generated for the dataset considered.

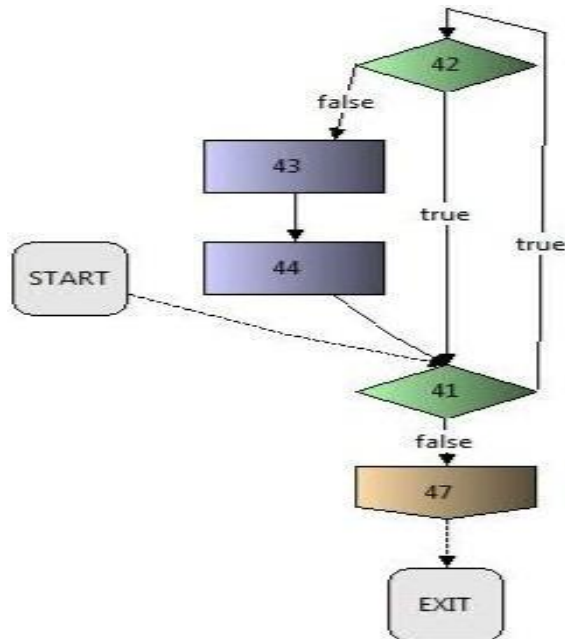


Fig. 3. ESDG graph-2

APFD metric used, measure effectiveness of new test cases. This research proposes approach for regression test case prioritization which orders test case on basis of changed graph. Changed part of graph is considered using dependency. ESDG graph is constructed and changed node is determined by slicing ESDG model. Results of approach increases APFD value when compared with related approach. Regression test prioritization orders test case which is re-executed in regression testing. ESDG graph identifies changes and gives coverage information for every test case. Modified information identifies changed statements. Selected test case is prioritized with genetic algorithm for increasing fault detection rate. Efficiency of approach is calculated using APFD metric.

References

- [1] Kaur, A., & Goyal, S. (2011). A genetic algorithm for regression test case prioritization using code coverage. *International journal on computer science and engineering*, 3(5), 1839-1847.
- [2] Suman, S. (2012). A genetic algorithm for regression test sequence optimization. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(7), 478-481.
- [3] Panigrahi, C.R., & Mall, R. (2012). A hybrid regression test selection technique for object-oriented programs. *International Journal of Software Engineering and Its Applications*, 6(4), 17-34.
- [4] Musa, S., Md Sultan A.B., Bin A.A., Ghani, A., & Baharom, S. (2014). A Regression Test Case Selection and Prioritization for Object-Oriented Programs using Dependency Graph and Genetic Algorithm. *Research Invent. International Journal of Engineering and Science*, 4(7), 54-64.
- [5] Panigrahi, C.R., & Mall, R. (2013). An approach to prioritize the regression test cases of object-oriented programs. *CSI transactions on ICT*, 1(2), 159-173.
- [6] Sultan, A.B.M., Ghani, A.A.A., Baharom, S., & Musa, S. (2014). An Evolutionary Regression Test Case Prioritization based on dependence graph and genetic algorithm for object-oriented programs. In 2nd *International conference on emerging trends in engineering and technology*, 22-26.
- [7] Malloy, B.A., McGregor, J.D., & Medikonda, A. An Extensible Representation for Object-Oriented Software.
- [8] Mishra, P.K., & Pattanaik, B.K.S.S. (2013). Analysis of test case prioritization in regression testing using genetic algorithm. *International Journal of Computer Applications*, 75(8).
- [9] Srivastava, P.R., & Kim, T.H. (2009). Application of genetic algorithm in software testing. *International Journal of software Engineering and its Applications*, 3(4), 87-96.
- [10] Mor, A. (2014). Evaluate the effectiveness of test suite prioritization techniques using APFD metric. *IOSR Journal of Computer*, 16(4), 47-51.
- [11] Arora, V., Bhatia, R.K., & Singh, M. (2012). Evaluation of flow graph and dependence graphs for program representation. *International Journal of Computer Applications*, 56(14), 18-23.
- [12] Raju, S., & Uma, G.V. (2012). Factors oriented test case prioritization technique in regression testing using genetic algorithm. *European Journal of Scientific Research*, 74(3), 389-402.
- [13] Jacob, T.P., & Ravi, T. (2013). Optimization of test cases by prioritization. *Journal of Computer Science*.
- [14] Musa, S., Md. Sultan, A.B., Ghani, A.A.A., & Baharom, S. (2014). Regression Test Case Selection & Prioritization Using Dependence Graph and Genetic Algorithm. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 16(3), IV 38-47.
- [15] Musa, S., Sultan, A.B.M., Abd Ghani, A., & Bahrom, S. (2014). Regression test framework based on extended system dependence graph for object-oriented programs. *International Journal of Advances in Software Engineering & Research Methodology-IJSERM*, 1(2), 28-33.
- [16] Krishnamoorthi, R., & Mary, S.S.A. (2009). Regression test suite prioritization using genetic algorithms. *International Journal of Hybrid Information Technology*, 2(3), 35-52.
- [17] Yadav, D.K., & Dutta, S. (2017). Regression test case prioritization technique using genetic algorithm. In

Advances in computational intelligence, 133-140.

- [18] Liang, D., & Harrold, M.J. (1998). Slicing objects using system dependence graphs. In Proceedings. International Conference on Software Maintenance, (Cat. No. 98CB36272), 358-367.
- [19] Gupta, R., & Yadav, A.K. (2013). Study of test case prioritization technique using APFD. *International Journal of Computers & Technology*, 10(3), 1475-1481.
- [20] Srivastava, P.R. (2008). Test case prioritization. *Journal of Theoretical & Applied Information Technology*, 4(3).
- [21] Panigrahi, C.R., & Mall, R. (2010). *Test Case Prioritization of Object-Oriented Programs*, Setlabs Briefings, 9(4).