

Effective Test Suite Optimization for Improving the Coverage Standards Using Hybrid Wrapper Filter Memetic Algorithm

Dr. P. Karuppusamy,
Professor,
Department of EEE,
Shree Venkateshwara Hi-Tech Engineering College,
Erode, India.
Email: pkaruppusamy@ieee.org

Abstract: The continuous progress and developments in the technologies related to the database and computer has heightened the significance of selecting the features. The most common methods that are used in choosing the features often come with the peril of choosing the features subsets that are unsuitable with the opted algorithm for the induction. Few methods that include the induction procedure for validating the subsets in the feature, despite its prediction capability is computationally more intensive. So to sort out the solution for the aforementioned problems, the proposed method in the paper utilizes the hybrid method for choosing the features clubbing the wrapper filter along with the structure of the Memetic to have an improved coverage-centered test-case escalation. The hybridized procedure optimizes the test suite integrating the call-stack along with it. The experimental observations obtained also exhibits the performance enhancement and the effective ness of the hybridized procedure that leads to cost factor minimization.

Keywords: Test Suite, Data Base, Computer, Coverage, Feature Selection, Wrapper Filter-Memetic Algorithm

1. Introduction

The array of active-calls related to the threads of the stack centered architecture is coined as the call-stack. The return type, name space qualifiers, name and the parameter list of the procedure is listed into the call-stack. The method to be followed and the instruments to be utilized in performing a dynamic collection of the concurrent multitasking GUI applications call-stacks is put forward in the paper.

The method put forward also attempts to evaluate the application of graphical user interface, by using the tool employed to minimize test suite. This tool minimizes the test-cases to the maximum when compared to the prevailing test-case. The library entries utilized by the graphical user interface is also included. The experiment conducted for the method put forth exhibits the efficacy and the viability by gathering the call-stacks as the coverage standards for the application of graphical user interface.

The graphical user application that are incident obsessed are not similar and equal as they demand new standards of coverage. The basic motivation for performing the test case minimizing using the call stack is to remove the test suite that is equivalent to another test suite possessing same incident, with the capability of producing same array of call-stacks, this is done to protect the assets, minimizing the cost and benefiting in terms of compressing the valued context data.

The swift development and the progress in the database and the computer methodologies and has heightened the significance of the choosing the features. The filter and the wrapper method if used separately result in the peril of choosing the features subsets that are unsuitable with the opted algorithm for the induction and are computationally intensive respectively. The filter method is “computationally cheap” as it does not include any algorithm for induction and the wrapper method has high prediction accuracy than the filter.

The proposed method clubs both the wrapper and the filter for choosing the features and uses the Memetic structure. The Memetic structure developed by R. Darwin’s is the combination of genetic and local search algorithm, it belong to the family of the population centered metaheuristics based scheme. “The meme denotes an equivalent to the gene to maintain a population pool comprising the several solutions simultaneously for the problem”

The hybrid procedure is basically an efficient method for conducting the optimization over the test suite, along with development of “coverage based test case minimization incorporated with the Memetic structure relied escalation and the experimental results exhibits the performance enhancement and the effectiveness of the hybridized procedure that leads to cost factor minimization.

The effective test suite optimization for improving the coverage standards using hybrid wrapper filter-Memetic (HWFM) algorithm. Is planned with the section 2 providing the details of the methods employed in the test case minimization, followed by the proposed algorithm in section 3, experimental validations in the section 4 and the conclusion in section 5.

2. Related works

Basili, Victor R., et al [1] put forth a “TAME project” to enhance the software ecosystem. Korel, Bogdan. Et al [2] designed a method to generate the test data using the software that is automated employing the “alternate variable method” that closely resembles the “hill climbing.” Gendreau, Michel, et al [3] designed an innovative “insertion as well as post optimization procedure” for the TSP taking into consideration only the candidate solutions that are nearby. Jukes, F. et al [4] to sort out remedies for the

concurrent problem utilized the "Genetic Algorithm with Variable Range of Local Search for Tracking Changing Environments." And optimized the dynamic functions. G. Rothermel, R. et al [5] utilized the integrated optimization techniques to perform the test case prioritizing.

Corne, David et al [6] proposed novel concepts based on the optimization, he introduced the scheme of multi-objective optimization. Knowles et al [7] put forth "A comparative assessment of memetic, evolutionary, and constructive algorithms for the multiobjective d-MST problem." Krasnogor et al [8] conducted the "Self generating metaheuristics in bioinformatics" Zhu et al [9] and [10] performed the "Markov blanket-embedded genetic algorithm for gene selection" and the "Wrapper-filter feature selection algorithm using a memetic framework."

McMaster, et al [11] conducted the "Call-stack coverage for GUI test suite reduction." Praditwong, et al [12] performed the "Software module clustering as a multi-objective search problem." Ding et al [13] put forth the "Automatic generation of software test data based on hybrid particle swarm genetic algorithm." Singh, et al [14] designed "An integrated approach to test suite selection using ACO and Genetic Algorithm." Arcuri, et al [15] devised "A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering." Marchetto, et al [16] performed the test case ordering using the multi-objective scheme. Raj, Jennifer S., et al [17] formed the "Virtual Structure for Sustainable Wireless Networks in Cloud Services and Enterprise Information System." Smys, S., et al [18] has discussed the possible technologies available in the computing and as well as communication. Rahimunnisa, K et al [19] put forth the "Hybridized Genetic-Simulated Annealing Algorithm for Performance Optimization in Wireless Adhoc Network." Shakya, Subarna, et al [20] presented the multi-objective optimization that is adaptable using the bio-inspired procedures.

3. Proposed Work

The proposed procedure clubbing the wrapper and filter in the choosing the feature along with the Memetic to escalate the test case that are coverage centered is organized with two phases (i) Minimization of Test suite and (ii) Optimizing Test Suite.

- (i) Minimization of Test-Suite: to have the minimization of test-suite the call-stack is gathered [11], and the amount of repeated test case is minimized. The stacks are gathered dynamically from the applications of the GUI with the concurrent multitasking capability. Even the libraries utilized by the GUI are included in the process of identifying the replications. The "multiple function calls" were developed in the application of GUI to manage every event, and was provided as the input for the generation of the test-case. Further the information's

structured such as the “event managers”, the utilized methods and the values put forward by the “event managers” were reclaimed followed by conducting the examination over the “event manager.” Every declared incidents are investigated to receive the methods confirmed. These procedures are performed while the actions on the particular button are performed. Based on the coverage of every event the test case was generated. This causes the replication while similar procedures are executed. Using this similar test for the same events are generated. The approach that is stack centered is utilized to produce the test case and all the incidents are managed and stored by the stack. The computation of the entire set was done merging all the call stacks that are unique, based on the observation from each test case over the suite. The values are reserved in the stack with the event managers, procedure followed and the parameters used. The coverage of the call stack and the incident according to the performance are compared. “Call stack-reduced suites were substantially larger than suites reduced by other criteria. Thus, it seemed possible that the call stack coverage may have been preserving more fault detection solely on the basis of including more test cases”.

- (ii) Optimizing Test Suite: The Memetic [6, 9, 14] based feature selection is done for choosing the gene, that is utilized in optimizing the test suite. Choosing features is done using the genetic wrapper and the filter method along with the Memetic structure to sort the “feature space effectively with a high prediction accuracy”. The algorithm empowered by the basic structure of the evolution algorithm integration of the local search. “The algorithm initializes the population of test cases, develops a mating pool by sampling the population with the fitness proportional probability. Performs the crossover by pairing the test case randomly for the generation and inclusion of new off springs. Performs the mutation with the low probability mutation operator to the offspring set” Replace the worst solutions in the population with the generated offspring’s. Repeat the process until the stopping criteria is met.

The outcome of minimized test suite is initialized as population in the optimization stage. New test are produced during the cross over and the “mutation operator injects the new material into the population to modify the existing solution at a low rate and generates a new solution. A fitness function evaluates the test case for reproduction by steadily improving the fitness values with the each generation until a solution is found”

4. Performance Analysis

The system hybridized is validated using various applications, the test case coverage are mainly checked. The cases are computed to sort out its capacity in identifying the faults and evaluating the number of covered codes using the test-case in a test suite while executing. The higher coverage test case are empowered with more capacities to detect faults than the low coverage test-case. Two different

application with the test cases ranging from 1800 – 2000 was utilized in validating the proposed system. The results below in figure.1 shows the test suite for execution status for the first application.

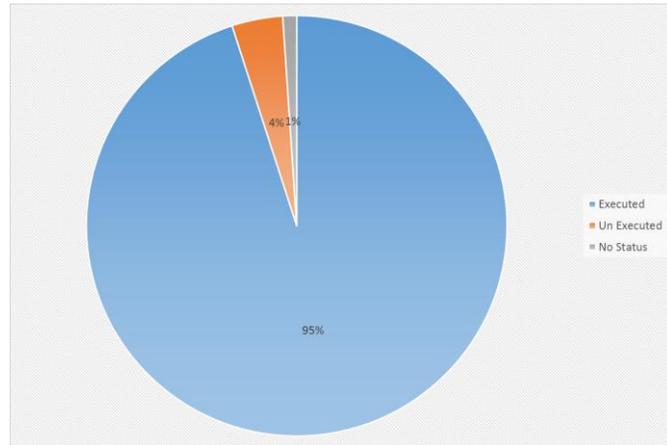


Figure.1 Execution Status of Application 1

The figure shows that among the 1789 test cases in the suite only 1700 cases were executed and the 70 cases were not executed and 19 cases did not provide any status. Figure.2 shows the functional coverage of the application

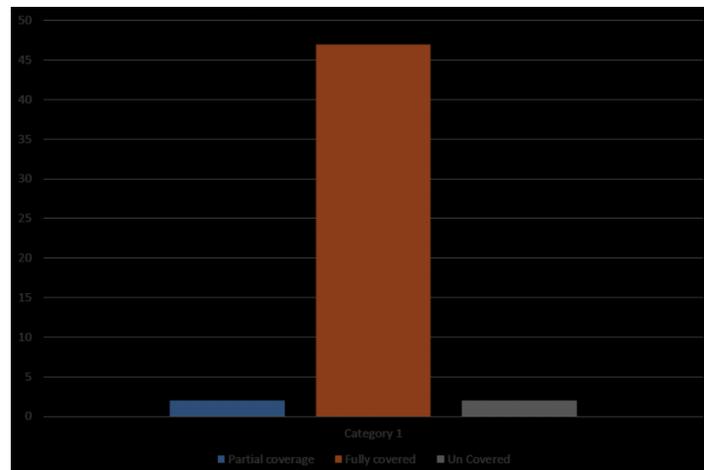


Figure.2 Functional Coverage of Application 1

The functional coverage is estimated using the following equation 1

$$\sum_{s \in \text{statements}} \begin{cases} ns, & s \in \text{code covered} \\ 0, & \text{else} \end{cases} \quad (1)$$

When all statements are covered it is termed as the fully covered and when partial statements are covered it is called as partial coverage or otherwise denoted as uncovered functions.

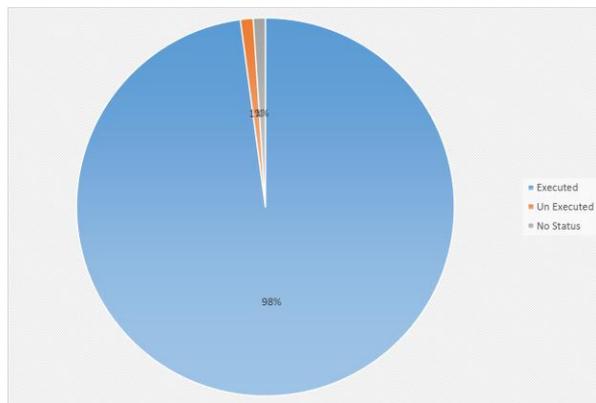


Figure.3 Execution Status of Application 2

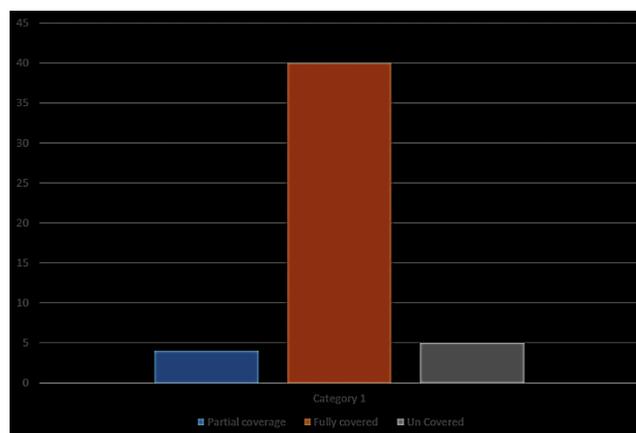


Figure.4 Functional Coverage of Application 2

The same process is repeated for the second application and results are observed. Among 1856 test cases, 1800 are executed and 20 are unexecuted and balance are reported with no status.

5. Conclusion

The technique used in the paper presents the tool to gather the call-stacks from concurrent multi-tasking graphical user interface applications, along with the library entries of GUI .The proposed system minimized the test suite and the optimized the test suite using the hybridized Memetic frame work to improve the coverage standards by proper selection of features using the call-stack as the coverage standard. The experimental results observed based on two different application ensured the capability of the proposed method in terms of the coverage. In future the paper is to develop a better coverage standard that minimizes the amount of coverage demands while maintaining the necessary qualities of the active calls in the stacks. This done to further improve the effectiveness of the Memetic.

References

- [1] Basili, Victor R., and H. Dieter Rombach. "The TAME project: Towards improvement-oriented software environments." *IEEE Transactions on software engineering* 14, no. 6 (1988): 758-773.
- [2] Korel, Bogdan. "Automated software test data generation." *IEEE Transactions on software engineering* 16, no. 8 (1990): 870-879.
- [3] Gendreau, Michel, Alain Hertz, and Gilbert Laporte. "New insertion and postoptimization procedures for the traveling salesman problem." *Operations Research* 40, no. 6 (1992): 1086-1094.
- [4] Jukes, F. Vavak-TC Fogarty-K. "A Genetic Algorithm with Variable Range of Local Search for Tracking Changing Environments." In *Parallel Problem Solving from Nature-PPSN IV: International Conference on Evolutionary Computation. The 4th International Conference on Parallel Problem Solving from Nature Berlin, Germany, September 22-26, 1996. Proceedings*, vol. 114, p. 376. Springer Science & Business Media, 1996.
- [5] G. Rothmel, R. Untch, C. Chu, M. Harrold, Test case prioritization: an empirical study, in *Proceedings of International Conference on Software Maintenance (IEEE Computer Society, 1999)*, pp. 179–188
- [6] Corne, David, Marco Dorigo, Fred Glover, Dipankar Dasgupta, Pablo Moscato, Riccardo Poli, and Kenneth V. Price. *New ideas in optimization*. McGraw-Hill Ltd., UK, 1999.
- [7] Knowles, Joshua, and David Corne. "A comparative assessment of memetic, evolutionary, and constructive algorithms for the multiobjective d-MST problem." In *GECCO-2001 workshop program*, pp. 162-167. 2001.

- [8] Krasnogor, Natalio. "Self generating metaheuristics in bioinformatics: The proteins structure comparison case." *Genetic Programming and Evolvable Machines* 5, no. 2 (2004): 181-201.
- [9] Zhu, Zexuan, Yew-Soon Ong, and Manoranjan Dash. "Markov blanket-embedded genetic algorithm for gene selection." *Pattern Recognition* 40, no. 11 (2007): 3236-3248.
- [10] Zhu, Zexuan, Yew-Soon Ong, and Manoranjan Dash. "Wrapper-filter feature selection algorithm using a memetic framework." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37, no. 1 (2007): 70-76.
- [11] McMaster, Scott, and Atif Memon. "Call-stack coverage for GUI test suite reduction." *IEEE Transactions on Software Engineering* 34, no. 1 (2008): 99-115.
- [12] Praditwong, Kata, Mark Harman, and Xin Yao. "Software module clustering as a multi-objective search problem." *IEEE Transactions on Software Engineering* 37, no. 2 (2010): 264-282.
- [13] Ding, Rui, Xianbin Feng, Shuping Li, and Hongbin Dong. "Automatic generation of software test data based on hybrid particle swarm genetic algorithm." In *2012 IEEE Symposium on Electrical & Electronics Engineering (EEESYM)*, pp. 670-673. IEEE, 2012.
- [14] Singh, Gurinder, and Dinesh Gupta. "An integrated approach to test suite selection using ACO and Genetic Algorithm." *International Journal of Advanced Research in Computer Science and Software Engineering* 3, no. 6 (2013).
- [15] Arcuri, Andrea, and Lionel Briand. "A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering." *Software Testing, Verification and Reliability* 24, no. 3 (2014): 219-250
- [16] Marchetto, Alessandro, Md Mahfuzul Islam, Waseem Asghar, Angelo Susi, and Giuseppe Scanniello. "A multi-objective technique to prioritize test cases." *IEEE Transactions on Software Engineering* 42, no. 10 (2015): 918-940.
- [17] Raj, Jennifer S., and S. Smys. "Virtual Structure for Sustainable Wireless Networks In Cloud Services And Enterprise Information System." *Journal of ISMAC* 1, no. 03 (2019): 188-204.
- [18] Smys, S., Robert Bestak, Joy Iong-Zong Chen, and Ivan Kotuliak, eds. *International Conference on Computer Networks and Communication Technologies: ICCNCT 2018*. Vol. 15. Springer, 2018.
- [19] Rahimunnisa, K. "Hybridized Genetic-Simulated Annealing Algorithm for Performance Optimization In Wireless Adhoc Network." *Journal of Soft Computing Paradigm (JSCP)* 1, no. 01 (2019): 1-13.
- [20] Shakya, Subarna, and Lalitpur Nepal Pulchowk. "Intelligent and Adaptive Multi-Objective Optimization in WANET Using Bio Inspired Algorithms." *Journal of Soft Computing Paradigm (JSCP)* 2, no. 01 (2020): 13-23.

Authors Biography

Dr. P. Karrupusamy, is the Professor and Head, of Department of Electrical and Electronics Engineering, Shree Venkateshwara Hi-Tech Engineering College, Erode, India. His research area includes Computer Networks, Chaos Theory, Particle Swarm Optimization, Probabilistic Computing, Fuzzy, Bio- Inspired Computing, Data Visualization, Fault Diagnosis, Robotics, Internet of Things, Neuro Computing, Information Retrieval Human-Machine Interface and Network Security.