

IPOG Modified Design Technique for Effective Testing

Shwetha M. S., Girijamma H A

Abstract: Software testing is a very crucial, effective and efficient stage in Software Development Life Cycle. As Customers satisfaction and reliability is very essential, this can be achieved by testing phase. The cost can be reduced when testing time is decreased. Hence, combinatorial method is a very effective and well-proved method where high quality of software can be delivered with less time. It is very exhaustive and hard phase to check all the combination of the input parameters given to authenticate the proper functioning of software system before delivering. Many issues are triggered in an application by the interaction of one or more parameters. Hence it is significant to check all the combination of N or fewer parameters in all N-way combinational input. This way of combinatorial testing will yield high guarantee software system such that all the faults have been discovered effectively. Manual Testing of this type of combinatorial inputs is impossible so there are few standard algorithms such as IPOG-C later defined as IPOGD, etc. In this paper, we are presenting the performance of Combinatorial Testing Technique called IPOG-Modified Design method with the IPOGD Technique and Manual way of test case generation. The results are evaluated for N-way combinational inputs of seven parameters. Evaluation of results shows that the IPOG-Modified Design Technique yields better performance than the IPOGD Technique and manual technique for the same input data set. Over all the IPOGD and IPOGMD Combinatorial testing methods can reduce cost, improves efficiency in software testing for numerous applications.

Keywords: Combinatorial Software Testing Methods – Manual & IPOGD & IPOGMD

I. INTRODUCTION

Software development is a manufacturing process of developing software system based on the requirements of the client. Software Development follows the SDLC process to develop the software. Thus testing is essential activity during development of software which analyzes the software to ensure that the actual behavior of the software correctly stick to its specification. Software testing is done by execution of the software on a particular input data set and comparing results actual output with the expected output. The set of program input data and the corresponding actual software output with the expected output is called a test case for the program. These Test cases are very useful to analyze the proper functioning of the developed software system. All these test cases created covers some requirements of the software system such as statements, decisions, definition-use pairs, or paths of interest as used in white-box testing ,

Revised Manuscript Received on December 12, 2019.

Mrs. Shwetha. M.S, Department of Computer Science, R N S Institute of Technology, Bangalore, INDIA. Email: shwetha.ms@in.unisys.com

Dr. Girijamma H A, Department of Computer Science, R N S Institute of Technology, Bangalore, INDIA. Email: girijakasal@gmail.com

coverage of special input values and output values derived from the user requirement specification. Covering unique requirements while testing ensures the better performance of the software system. Testing is a very ex and expensive stage in Software Development Life Cycle. Software Testing is done to achieve quality, reliability and performance of the software system developed. Designing the appropriate test cases for the function module plays very important role to make sure the quality of the software testing.

Software Testing faces extremely hard for complex software. In some cases, it is highly impossible to demonstrate testing that the software works correctly for all the possible inputs. Early detection of errors makes application more secured and reduces the cost of software system developed. Smart testing is highly required to deliver high quality product to the client. Therefore, test design technique is very important phase for maximum coverage which helps to deliver high quality of software. Combinatorial testing dramatically decreases the number of test set combinations. It is very helpful especially when application has multiple parameters. Few Black box techniques are used to list values for individual factors. Practically it is impossible and impractical to perform all the test combinations. Maximum coverage is done by using CT. It can be used from Unit testing to User Acceptance Testing. It saves time 30 to 40% in identifying and documenting test cases. Secondly, it identifies more effective and efficient tests cases. Thus, more number of bugs are detected during testing phase. Test data generated by this method is so effective and it finds twice as many bugs found per hour compared to manual testing. Sophisticated and proven algorithm is used. Multiple studies shows that approximately 85% of software faults can be identified by 2-way solutions. This approach is extremely effective and straightforward to test all kind of different applications [1][3].

It is very important to check software implementation errors, which contribute more for system security vulnerabilities. Performing Exhaustive testing is highly impossible which takes lot of staff time and involves enormous cost for testing. Testing process also depends on the Criticality of Software system developed. If the system is less critical and budget constraints often limits the amount of testing which yields risk in residual errors that leads to system failures. Today software is forced to develop complex software to meet the multi-functional requirements of the clients. This complex Software uses more number of input parameters, which leads

to more and more input interaction with multiple dependencies between the input parameters. This yields more faults due to the interaction.

Combinatorial is a branch of mathematics deals with the fundamental principles of counting. Permutations & combinations are important counting principles adopted in most of the research domains. If one event occurs, m -way and another event occurs n -ways. Then $(m \times n)$ ways that the events can occur in sequence. The given fundamental principle is applicable for the testing process. Software system quality is evaluated which depends on the execution of the test cases. Test cases are developed based on the user requirements. Considering the parameters for developing the Test suite combinatorial method plays an important role. Manual Testing is highly impossible to generate all the possible test cases for the parameters considered for testing the software. Usually in manual testing all the possible combination of inputs are considered for testing purpose (permutation of all input data set). This type testing is highly impossible and measuring with respect to time is not possible so Automatic testing tools are invented to generate the test case, automate the testing and analyze the output for the reliability and Security of the software system developed. [6][8].

II. METHODOLOGY

A. System Architecture

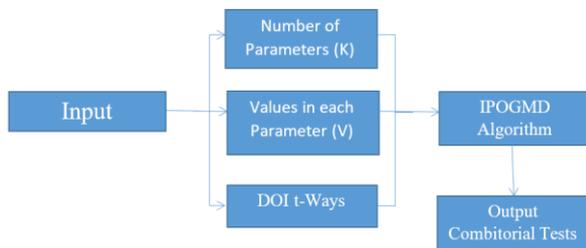


Fig 1: IPOGMD System Architecture

Fig 1 shows the IPOGMD (In-parameter-order general modified Doubling) algorithm which uses different greedy methodology to develop covering arrays (CA). IPO method covers the CA in terms of parameter order. In every iteration, the algorithm adds a one parameter at a time. IPOG has two stages, which extends the CA horizontally and vertically and then it doubles. IPOGMD implemented IPOG strategy with recursive construction method.

B. Software Testing Techniques

Many different software Techniques are adopted in the industries. In this paper we are comparing the performance of IPOG (In-Parameter-Order General) with the performance our defined algorithm called IPOGMD (In-Parameter-Order General Modified Doubling) Testing Technique for Combinatorial Inputs.

III. IPOGMD: BRIEF DESCRIPTION THE IPOG-D STRATEGY

IPOG Design strategy is an efficient test generation techniques used in multi-way testing of software system. It is an extension of pairwise testing method for multi-way. This Technique includes explicitly enumerating entire probable t -way combinations, which is impractical when more number of combinations and constraint resources. When there is increase in strength t -way, number of test cases grows exponentially. This is referred as combinatorial explosion. Hence it is expensive with respect to storage and time to enumerate all the combinations [10].

Minimization of test generation should be maintained with respect to space and time. In pair-wise testing, we get modest number of combination for larger system configuration. Hence, focus of current work is to reduce and generate smaller size of the test set. It gives very little attention on space and time.

T -way testing has been considered as a well proved technique to generate test data. It does not require any knowledge regarding implementation. Thus it is a lightweight process as input parameters are identified and the possible combination of the given parameter. Next point it has high capability as it very effective for different kinds of application. Previous research shows that faults in different industrial application can be detected by 6-way interaction. It means to say that all faults will be exposed when 6-way testing is performed. The process of generation of test data becomes key feature in software industry [2][4].

Multi-way testing can be implemented effectively by using IPOG-D algorithm. It is a method, which is derived from IPOG, and a recursive approach called D-construction, which reduces enumerated combinations. For 3-way interaction, D construction increases the number of parameter twice and hence the name called double. However, the limitation is, domain size should be same. It generates for an input of 3-way $ts1$ and 2-way for 3 Boolean parameters $ts2$ for k parameter. The generated $ts2$ tests data set covers all the $2k$ parameters.

Consider test set Ts has 6 Boolean parameters. Each column in is duplicated in $ts1$ and placed in Ts next to the original column in.

For example, the fourth row (0 1 1) in test set $ts1$ becomes the second row (00 11 11) in test set ts . The resulting test set is referred to as the expanded copy of test set $ts1$. The Upper part of the test set ts is an expanded copy of test set $ts1$. Advantage is that enumeration is reduced [10][11].

IV. IPOGMD: BRIEF DESCRIPTION OF IPOG-MODIFIED DESIGN STRATEGY

IPOGMD is one of the optimized and effective technique to derive small set of test data in combinatorial testing. When more number of parameter is present in an application, a test

set size increases that will lead to explosion of combinations. To resolve this problem IPOGMD is introduced. IPOGMD also uses horizontal growth and vertical growth to derive required test data. IPOGMD minimizes logically and removes repetitive combination to derive optimized test set. Hence, it generates small size test data compared to IPOGD. So, achieves greater coverage. In fewer tests. As it generates small size test data, which saves time during test execution cycle. High quality of the application can be delivered to the clients by using this method. Evaluation of IPOGMD is very effective as it saves execution time. As it generates small test set, it is effective for higher strength parameter. As IPOGMD generates very small set of data, it is more effective with respect to test case writing and execution phase. Hence, it saves much time in Software Testing Life Cycle phase and by using this method, it is possible to deliver high quality product to clients.

IPOG Modified Design algorithm is an extended version of basic IPOG-D Techniques used in most of the software Testing Environment. In this system, considering 2K parameters which is more or equal to $t(2K \geq t)$ where “t” is called the degree of Interaction coverage. This method derives t-way test set for k number of parameters and a(t-1)-way data test set for k parameters. It uses doubling construction technique. This feature creates the optimized data test set. This method is deterministic and recursive in nature and will be efficient up to 6-way of Degree of Interaction coverage. CA are recursively-built from least covering arrays and always-exact data set is generated when the system configuration is same [10].

V. EXPERIMENTAL SETUP

Due to large volume of Combination data Set, it is highly impossible to test the application manually. So ACTS Tool version 3.1 is used to check the total number of Test cases yields by different Testing methods, it is evaluated with the IPOGMD method of Testing, and Results are evaluated. A software testing is performed taking seven parameters and for 2-way to 6-Way combinatorial testing. Maximum six way of combinatorial testing can be achieved [4]. ACTS Testing Tool executed over windows Platform on i3-Series of Computer Configuration system. Our Test bed considers seven parameters to test by considering only two parameter once (2- way), similarly 3-way to 6-way. These experiments are run for sufficient time and results are tabulated.

VI. RESULT AND DISCUSSION

The results were obtained by considering seven parameters with combinatorial ways of giving inputs to Manual Way of Test case generation, Automation tool usage results for IPOGD algorithm with the our Technique Called IPOG Modified Design algorithm for combinatorial inputs are 2, 3, 4, 5 & 6 combinations of input parameters (Degree of Interaction Coverage).

A. Test Cases Generation

Computed the Test cases for all the three methods under consideration and analyzed the performance with respect to the Test Case generation.

Table I: Results obtained for N-way Combinatorial Data Set-Number of Test Case Generated

Degree of Interaction Coverage	Manual	IPOGD	IPOGMD
2	42	268	27
3	210	316	117
4	840	704	531
5	2520	2816	2006
6	5040	5120	4096

The experimental results showed that IPOG Modified Design algorithm performs better in generation of Test Cases than IPOGD algorithm. It generates minimum number of Test cases, which satisfies all the input requirements. Now a day, we use Automatic testing tools, which detect the faults faster than the manual methods. These kinds of algorithm implementation surely reduce human efforts with faster detection of faults, saves lots of time and cost in testing process. The graphical illustrations are shown in fig 2.

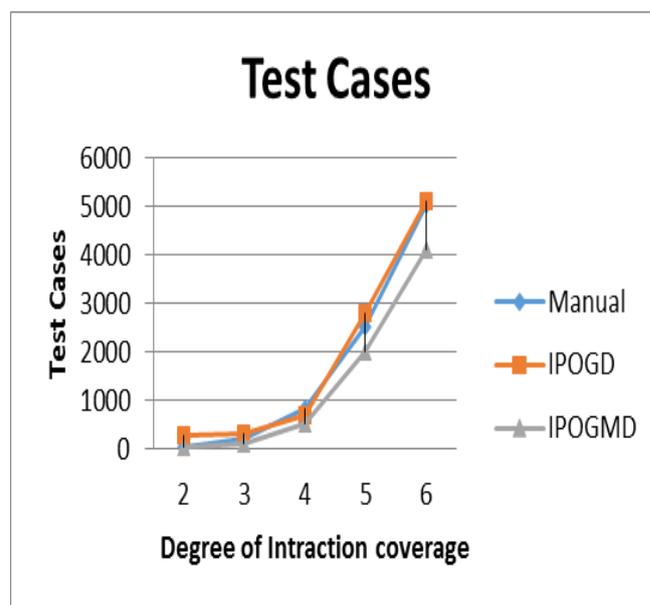


Fig 2: Test Cases generation using combinatorial algorithms

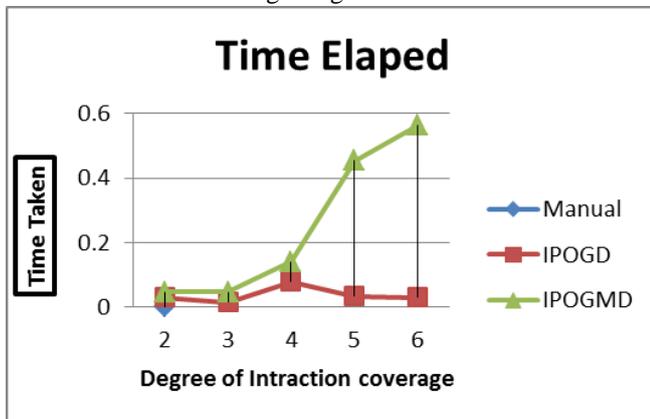
B. Time taken to generate the Test Cases

The time taken for generation of Test cases executed for above said algorithms are tabulated in table 2. The time complexities in computing Test cases are generated using the ACTS tools for the experiment conducted for same data set.

Table II: Time taken to generate test cases for IPOGD & IPOGMD Techniques

Degree of Interaction Coverage	Manual	IPOGD (Sec)	IPOGMD (Sec)
2	Test case generation time cannot be measured.	0.031	0.047
3		0.015	0.047
4		0.078	0.141
5		0.033	0.453
6		0.031	0.563

The experimental results showed that IPOG Modified Design algorithm takes much more time for generating fewer test cases than the IPOGD method. The graphical illustration is shown in figure 3. Manual way of finding the time for generation is not possible and it is of no use. Analysis of this time factors surely help researches to find how fast the automation tools working using these methods.

**Fig 3: Time taken to Test Cases manual, IPOGD & IPOGMD algorithms**

VII. CONCLUSION

IPOG-Modified Design method generates very lesser number of Test cases than IPOGD method and manual way of generation of Test cases for different Combinatorial input. The Performance of IPOG Modified Design technique helps in testing the application faster and effectively. Empirical data shows 100% faults are detected by maximum 6-way interaction. The Experiment shows that the outcome of IPOGMD is correlated with the output of IPOGD and Manual method shows Correlation Co-efficient Value of 99.84%, which implies our testing methods, is in orientation with the other standard methods. The Time taken to generate test cases for same data set shows the IPOGMD method outperforms the IPOGD. The future work of the proposed IPOGMD algorithm will be carried out for different set of parameters check for N-Ways of combination. Analyzing the performance for all the ways of Testing with additional performance parameters can be considered for quantifying the performance of the IPOGMD algorithm.

ACKNOWLEDGMENT

Authors are thankful to RNSIT Research center and VTU University for providing excellent environment to pursue

research work. Special thanks to Principal Dr. M. K. Venkatesha and Staff of CSE/ISE for their support and guidance in review meetings. I appreciate Dr. B.I.D Kumar for his time and effort for his valuable review comments.

REFERENCES

1. D. M. Cohen; S. R. Dalal; M. L. Fredman; and G. C. Patton; The AETG System: An Approach to Testing based on Combinatorial Design. IEEE Transactions on Software Engineering, 1997, 23, pp. 437–443.
2. Williams A W. In: Determination of test configurations for pair-wise interaction coverage. In: Proceedings of 13th International Conference on the Testing of Communicating Systems. Ottawa, Canada, 59–74, 2000.
3. Kuhn D R; Wallace D; Gallo; Software fault interactions and implications for software testing. IEEE Transactions on Software Engineering, 2004, 30, 6, 418–421.
4. Y. Lei; R. Kacker; D. R. Kuhn; V. Okun, and J. Lawrence. In: IPOG: A general strategy for T-way software testing. In: 14th Annual IEEE International Conference and Workshop on Engineer of Computer Based Systems (ECBS 2007), 26 – 29 March 2007, Tucson, Arizona, USA, 549–556, 2007
5. Yu Lei; Raghu Kacker; D. Richard Kuhn; Vadim Okun and James Lawrence; IPOG/IPOG-D: efficient test generation for multi-way combinatorial testing. Software Testing Verification and reliability 2008, 8, 125-148.
6. R. Kuhn; R. Kacker; Y. Lei and J. Hunter; Combinatorial Software Testing. Computers, 2009, 42, 8, 94–96.
7. M. I. Younis; K. Z. Zamli; M. F. J. Klaib; Z. C. Soh; S. C. Abdullah and N. A. M. Isa; Assessing IRPS as an Efficient Pairwise Test Data Generation Strateg; International Journal of Advanced Intelligence Paradigms (IJAIIP), 2010, 2, pp. 90-104.
8. C. Nie; H. Leung; A survey of combinatorial testing; ACM Comput. Surv., 2011, 43, 2, 1–29.
9. Shwetha.M.S; Dr.Girijamma H A; Survey on advanced combinatorial testing and applications; International Journal of Applied Engineering Research (IJAEER) , 2015, 10, 86 pp.482-486.
10. Shwetha M S; Dr. Girijamma H A. In: Recast IPOG-D Algorithm with Constraint Handling for Combinatorial Testing. In: Proceedings of the IEEE conference - ICRAECT 2017, March 14-17, 2017.
11. Shwetha.M.S; Dr. Girijamma H A; Reduce the Cost of Software Testing by Implementing Combinatorial Testing Techniques; Journal of Advanced Research in Dynamical and Control System, 2018, ISSN 1943-023, 13, Special Issue, pp 354-365.

AUTHORS PROFILE



Mrs. Shwetha. M.S currently working with Unisys software India Ltd. Bangalore and having 14 years of experience in Software testing. She is pursuing, Ph.D in VTU, under the guidance. Dr. Girijamma H A, Prof. R.N.S Institute of Technology, Bangalore. She is doing Research in the area of Combinatorial Testing.



Dr. Girijamma H A received Ph. D from Kuvempu University in 2012. She is Professor at Department of Computer Science and Engineering, R.N.S Institute of Technology, Bangalore. Her current research domain includes fuzzy logic, medical image processing and machine learning.