



## **Fault and Future Aspects Based Dynamic Programming Approach to Estimate Path Sequence**

Kulvinder Singh Handa<sup>1</sup> Mehak Kalra<sup>2</sup>

<sup>1</sup>Faculty of Computer Engineering, <sup>2</sup>Research Scholar,  
Department of Computer Engineering,

University Institute of Engineering and Technology,  
Kurukshetra University, Kurukshetra, Haryana, INDIA.

E-mail (s): [kshanda@rediffmail.com](mailto:kshanda@rediffmail.com), [mehakkalra.17@gmail.com](mailto:mehakkalra.17@gmail.com),

**Abstract:** Testing plays a very important role in software development life cycle. Reliability of software depends upon how better the testing process is performed. Path testing is a very beneficial white box testing that play an important role to select the actual code test sequence as well as to perform the code optimization. The code optimization is actually performed in terms of detection of unusable code from the test sequence. There are number of aspects according to which the code sequence is decided. This paper proposed an algorithm using dynamic programming approach that will work on the basis of fault tolerance as well as the future aspect of a test case to estimate path sequence.

**Keywords:** Structural testing, Path testing, Dynamic programming

### **I. Introduction**

Testing is a process of executing a program with the aim of finding errors [1]. To perform testing, test cases are designed. A test case is a particular made up artificial situation upon which a program is exposed so as to find errors. A good test case is one that finds undiscovered errors. If testing is done properly, it uncovers errors and after fixing those errors software is being developed according to specifications [2]. Main purpose of software testing is to give the best quality for the product. Role of software testing involves checking that there should be no discrepancy in the software development process anywhere throughout the process. Quality should not get sacrificed at all [3]. Structural testing takes into account the code, code structure, internal design, and how they are coded. In structural testing tests are actually run by the computer on the built product [4]. Path testing consists of two steps: (i) select a finite set of paths to be exercised. (ii) Generate test data to execute the selected paths [5]. This paper presents an algorithm to generate optimized path sequence. First, estimating the cost of all possible test cases and then by using dynamic programming approach minimum cost path sequence will be generated. The cost of each test case will be decided on the basis of fault occurrence chances in that code block as well as on the basis of future test chances of code during maintenance.

### **II. Dynamic Programming**

Dynamic programming is a method for solving complex problems by breaking them down into simpler subproblems. It is applicable to problems exhibiting the properties of overlapping sub-problems which are only slightly smaller and optimal substructure [6].

Working methodology of dynamic programming approach is given as:

- Divide a problem into subproblems: Main problem is divided into a number of smaller, similar subproblems. Solution to main problem is expressed in terms of the solution for the smaller subproblems. Stage wise solutions start with the smallest subproblems.
- Construction of Table for Storage: Once the solution to a constrained version of the problem is computed, it stores that solution in a table until the solution is no longer needed by any future computation.
- Combining using Bottom-up: To obtain the solutions to subproblems of increasing size solutions of smallest subproblems is combined. The process is continued until we arrive at the solution of the original problem [7].

Dynamic programming involves selection of optimal decision rules that optimizes a certain performance criterion:

- The Principle of Optimality: An optimal sequence of decisions is obtained iff each subsequence must be optimal. It means if the initial state and decisions are optimal then the remaining decisions must constitute an optimal sequence w.r.t the state resulting from the first decision.
- Polynomial Break up: The original problem is divided into several subproblems. The division is done in such a way that the total number of subproblems to be solved should be a polynomial or almost a polynomial number [7].

### III. Related Work

A brief review of some important contributions in the field of path testing and dynamic programming from the existing literature is presented in this section.

Bruno *et al.* [5] worked on the GEO Applied to Path Testing & had shown that GEO is an attractive option for test data generation. Although its final path coverage was not the best for all the benchmarks evaluated, it consumed much less time than the SGA on parameter adjust. It was also faster for the most complex benchmarks, due to the way it stored its solutions, which was simpler and less complex than SGA's.

In this paper fundamental working principles, major area of applications strengths, drawbacks, comparison study of dynamic programming technique has been presented by Biswajit [7].

In this paper Zijian *et al.* [8] proposed a fast path selection approach based on graph partition. First, a critical path graph (CPG) was generated to implicitly enumerate almost all candidate critical paths, and then the CPG was partitioned into several sub graphs which contain limited numbers of paths using two graph partition approaches. After that, Monte Carlo simulation was applied on each sub graph for path selection. At last, according to the partition topology of the CPG and path sets selected from each sub graph, a path set for this original CPG was generated using Union and Cartesian product operations for testing SDDs. Further Zijian *et al.* [9] carried out the work in which the path selection problem was converted to a minimal space intersection problem, and a greedy path selection heuristics was proposed, the key point of which was to calculate the probability that the paths in a specified path set all met the delay constraint. Statistical timing analysis technologies and heuristics were used in the calculation. Experimental results show that the proposed approach was time-efficient and achieved a higher probability of capturing delay failures in comparison with conventional path selection approaches.

Lixin [10] proposed a method of dividing the program under test (PUT) into a several program segments with small cyclomatic complexity. This paper analyzed the relationship between the input variables and the fragments gave the approach of processing the split point and illustrated the benefits of the method for generating test data both theoretically and experimentally. For each fragment, the method decreased the complexity of large program while structural testing, increased the efficiency of the test data generation. Joachim [11] in the paper has introduced dynamic programming technique by way of example with the help of the computer algebra system Maple.

### IV. Proposed Work

To find the path sequence first the activity chart of software will be created. It can be of a module or the software. This model will be represented in the form of graph with exactly one initial state and one or more goal stages. Then the weights will be assigned to each edge on the basis of fault occurrence as well as the future perspective of that code block. The proposed work will then perform the dynamic programming approach to estimate the significant path sequence.

#### Algorithm: An algorithm to find minimum cost path

1. {
2. Define Number of nodes having Code block with Test Cases called T1,T2...Tn
3. Define Number of Faults Called F1,F2....Fm
4. Associate the Faults with Test Cases
 

$T1 = (F1, F2...Fk1)$   
 $T2 = (F1, F2...Fk2)$   
 $\cdot$   
 $\cdot$   
 $Tn = (F1, F2...Fkn)$
5. Assign the Priority to Test Cases on the basis of various factors

6. Define Cost of Each Test Case
  - a. For Priority 1 Assign Constant Cost Called Ccost\*3
  - b. For Priority 2 Assign Constant Cost Called Ccost\*2
  - c. For Priority 3 Assign Constant Cost Called Ccost\*1
7. Total Cost=0
8. for i=n-1 to 1
9. {
10. Find Min Cost node for Test Level (i) + Min Cost node for Test Level (i+1, n)
11. Total Cost=Min Cost(i)+Min Cost(i+1,n)
12. }
13. Return Total Cost
14. }

## V. Conclusions

In this paper an algorithm is generated to find the minimum cost path sequence. The software or module is represented in the form of graph with each node consisting of code block having test cases append to it, then weights are assigned on the basis of fault and future aspects. Dynamic programming approach is used to determine the path sequence.

## VI. References

- [1] Roger S.Pressman, "Software engineering-A Practitioner's approach" sixth edition.
- [2] System Testing and Quality Analysis available at <http://www.freetutes.com/systemanalysis/sa9-testing-quality-assurance.html>
- [3] Article by Mandeep, "Introduction and Importance of Software Testing in SDLC", available at <http://www.onestopsoftwaretesting.com/introduction-and-importance-of-software-testing-in-sdlc/#more-338>.
- [4] "Software testing fundamrntals" available at [www.softwaretestingfundamentals.com](http://www.softwaretestingfundamentals.com)
- [5] Abreu, B T.de, Martins, E., and De Susa, F L., "Generalized Extremal Optimization Applied to Path Testing", ISSRE 2006 Supplementary Conference Proceedings © Copyright 2006 Chillarege Press, 2006.
- [6] "Dynamic Programming" available at [http://en.wikipedia.org/wiki/Dynamic\\_programming](http://en.wikipedia.org/wiki/Dynamic_programming)
- [7] Bhowmik, B., "Dynamic Programming – Its Principles, Applications, Strengths, And Limitations", International Journal of Engineering Science and Technology Vol. 2(9), pp.4822-4826, 2010.
- [8] He, Z., Lv, T., Li, H., and Li, X., "Graph Partition Based Path Selection for Testing of Small Delay Defects", 15<sup>th</sup> Asia and South Pacific Design Automation Conference (ASP-DAC), 978-1-4244-5767-0/10@2010IEEE, pp.499-504, 2010.
- [9] He, Z., Lv, T., Li, H., and Li, X., "Fast Path Selection for Testing of Small Delay Defects Considering Path Correlations", 28th IEEE VLSI Test Symposium, 978-1-4244-6650-4/10©2010 IEEE, pp.3-8, 2010.
- [10] Wang, L., "A Program Segmentation Method for Testing Data Generating Based on Path Coverage", IEEE International Conference on Software Engineering and Service Sciences (ICSESS), 978-1-4244-6055-7/10©2010 IEEE, pp.565-568, 2010.
- [11] Zietz, J., "Dynamic Programming: An Introduction by Example", Department of Economics and Finance Working Paper Series, 2004.