

Application of Orthogonal Experimental Design for the Automatic Software Testing

Hao Wu

Department of Computer Science
Zhuhai College of Jilin University, JLUGH
Zhuhai, China
e-mail: haowu_mouse@hotmail.com

Abstract—According to inputting different combination of conditions so as to produce different impacts, software testing designs a large number of test cases. If the implementation of an overall test, due to the limit of the combination of conditions, it is difficult to carry out. In order to generate high quality test cases as early as possible to improve the efficiency of software testing, it is designed a generation tool of the automatic software testing case on orthogonal experimental design. For the test data, the use of that tool design test cases. The practice shows that a small number of test cases are generated, the error detection ability is strong, and it greatly improves the efficiency of software testing.

Keywords- *orthogonal experimental design; test case; software testing*

I. INTRODUCTION

Along with the more powerful software function and the improvement of software complexity, software development process is not easy to be controlled. Software quality problem has become an important factor to influence the development of computer application technology. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. The design of tests for software and other engineered products can be as challenging as the initial design of the product itself. A rich variety of test case design methods have evolved for software.

There are many applications in which the input domain is relatively limited. That is, the number of input parameters is small and the values that each of the parameters may take are clearly bounded. When these numbers are very small, it is possible to consider every input permutation and exhaustively test processing of the input domain. However, as the number of input values grows and the number of discrete values for each data item increases, exhaustive testing becomes impractical or impossible.

Orthogonal experimental design can be applied to problems in which the input domain is relatively small but too large to accommodate exhaustive testing. The orthogonal experimental design is particularly useful in finding errors associated with region faults—an error category associated with faulty logic within a software component.

Based on the research of orthogonal experimental design method, it considers that to be applied to design test cases. Through designing an automatically generated tool to generate test cases, it can be effective in reducing the number of test cases, so as to ensure accomplishing software testing

in the lower cost and the lower risk. The ultimate aim is to effectively improve testing results and the efficiency of software testing.

II. MAIN METHODOLOGY

A. Test Case Design

The design of tests for software and other engineered products can be as challenging as the initial design of the product itself. A rich variety of test case design methods have evolved for software. These methods provide the developer with a systematic approach to testing. More important, methods provide a mechanism that can help to ensure the completeness of tests and provide the highest likelihood for uncovering errors in software.

B. Orthogonal Experimental Design

1) Orthogonal Experimental Design Concepts

Orthogonal experimental design is an important branch of statistical mathematics, based on the probability theory, the mathematical statistics, and the standardized orthogonal table to arrange the test plan. According to the modern algebra of Galois theory, Orthogonal experimental design is a scientific test design method, which selects the right amount of representative points or uses cases from a large number of experimental data, so as to arrange experiments or tests reasonably.

Orthogonal experimental design usually determines the standard of test results pros and cons, which is called indicators that may affect tests as factors, and the impact factor as factor levels. In designing the optimal test, it must have a reasonable indicator and a reasonable reference to selecting factor and the corresponding level, in order to complete the specific test purpose.

2) Orthogonal Experimental Design Characteristics

Orthogonal experimental design is the study of multi-factor and level of design method, through the part of the test to find out the optimal level combination. To complete test requirements needed less number of experiment. The greater the number of factors and levels, the more obvious the advantage of this method is. The distribution of data points is uniform.

It is available to use corresponding range analysis method, variance analysis method and regression analysis method to analyze the test results, and can lead to many valuable

conclusions, better reliability. Orthogonal experimental design, base on the orthogonal table, is efficiency, rapid and economic test design method.

C. Orthogonal Table

1)

orthogonal Table Concepts

Orthogonal table is a kind of two-dimensional digital form with the neat comparability and the balanced collocation. The neat comparability means that each factor of each level is completely the same in the same piece of orthogonal table. Since the test of each factor of each level and other factors involved in the probability of each level test is exactly the same, it ensures that each factor level is ruled out other factors level of interference in the greatest degree. Therefore, it can be most effective in comparison and prospect, and be easy to find a good test condition. The balanced collocation means that any two columns (two factors) level collocation (horizontal formation of digital for) is exactly the same in the same piece of orthogonal table. This helps to ensure that the test condition is balanced dispersion in the level of combination of factors completely, thus which has a strong representation, and is easy to get good test conditions.

The orthogonal table is the foundation of the orthogonal experimental design, which forms as follow:

$$L_n (m^k) \quad (1)$$

The L is the symbol of the orthogonal table, and its symbol is expressed as follows:

- n: The number of trials arranged by the orthogonal table. That is the number of rows in the orthogonal table in directly corresponding to the number of production test cases.
- m: The most number of factors arranged in the orthogonal table. That is the number of columns in the orthogonal table in directly corresponding to the input parameters in the software module.
- k: The each factor level number is to get the largest number in each single factor. The orthogonal table contains values from 0 to the level number - 1 or from 1 to the level number, and each factor level values is corresponding to the input parameter.

In the process of the orthogonal experimental design, it can make use of the existing orthogonal table, and also use a mathematical model to generate an orthogonal table. The structure of the orthogonal table mainly uses the theory of the abstract algebra, and there are many existing standardized orthogonal tables, such as $L_4(2^3)$ 、 $L_8(2^7)$ 、 $L_9(3^4)$ 、 $L_{16}(4^5)$, etc. Using these orthogonal tables, the general experimental design can be completed. But for some experiments, due to the factor and the level of the testing modules are quite complex, it may not be able to select the proper one from these orthogonal tables. Then it should use a mathematical model to construct a new orthogonal table, involving more abstract algebra knowledge and as well as

many orthogonal table structure, and so far that have not been resolved.

2)

orthogonal Table Nature

- The frequency of different digital in each column is equal, such as $L_4(2^3)$ that there are two different figures in each column, and which appears twice.
- If two figures are seen as the ordered pairs in any two lines, and the occurrence number of each ordered pairs appears equal. For example, there are four ordered numbers of $L_4(2^3)$ and each of them appear once.

Due to the orthogonal nature of the table to arrange the experiment, the various levels of each factor are collocation balanced, and the experimental number is small, but there is a strong representation. The intuitive analysis, the variance analysis and effect analysis can all analyze the size of the impact of various factors on the experiment results, and to determine the best experiment plan. Effect analysis can also estimate the trend to the index values, and it can guide the experiment process conversely.

3)

orthogonal Experimental Design for Test Cases Strategy

By orthogonal experimental design for test cases strategy, there are simple designs and also complex ones. In the real work, it often encounters very complicated situations, because of the increase of the system scale.

There are two key strategies on designing the orthogonal experiment:

a) To Determine Factors

That all may affect the program result should be taken into account. That factors are not interactive with others can be added after the formation of use cases.

b) To Determine Levels

The limited value level is taken all. The infinite value level is determined by the boundary value analysis, the equivalence class analysis and the error determination method.

III. ORTHOGONAL EXPERIMENTAL TESTING CASE DESIGN

To design software test cases by orthogonal experimental design can greatly reduce the number of test cases, so as to improve the efficiency of software testing. That is the most suitable for the software module that a number of input parameters are together to determine the output results. As the software structure illustrated in Figure 1, three input parameters like 1, 2 and 3 together determine the result of the output parameter 1, and three input parameters like 3, 4 and 5 together determine the result of the output parameter 2, then to choose the orthogonal table should be respectively suitable for three factors as 1, 2, 3 and 3, 4, 5.

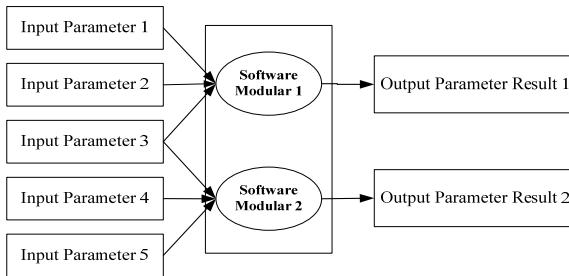


Figure 1. Software modules of Orthogonal Array to Design Test Cases.

A. Test Data Description

It illustrates the orthogonal experimental design to design test cases as example that it sets input interface on the software query module of report issued in the financial management and control project. For example, adding a query can filter out the corresponding budget statement in the units of the issued report, through setting the input parameters on this interface, which together determine the budget statement.

The input parameters are as follows:

- Budget cycle: Annual budget, Month budget
- Budget time: 2010、2011、201008、201108
- Budget classification: Sale power budget, Purchase power budget, Transportation power budget.
- Unit: A provincial grid company, a city, State grid.
- Report/Issue: Reported query, Issued query.
- Reported/Issued number: Reported number, Issued number.

B. Argumentation of Selection Orthogonal Experimental Design

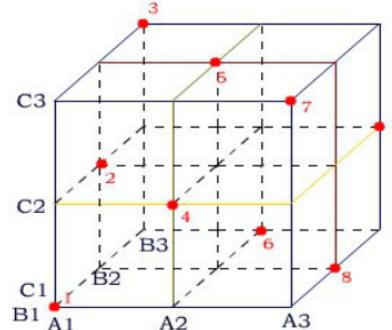
1) Comprehensive Test

Comprehensive testing is to put all the parameters and domains into one test collocation. For example, the test of 4 factors and 3 levels is needed to do testing use cases of 3 of 4 times square equal to 81.

The Advantage of the comprehensive test is that it can achieve the comprehensiveness of the test, analyze the effect of various factors and the interaction, and also choose the optimal level combination. However, the disadvantage of this approach is to require the test too many times and the workload is relatively large. For the test of m factors and n levels, the total number of test cases is n of m Times Square, which is difficult to fully achieve in the actual test work in some cases.

2) Orthogonal Experimental Design

When orthogonal experimental testing occurs, an $L_9(3^4)$ orthogonal array of test cases is created. The $L_9(3^4)$ orthogonal array has a balancing property. That is, test cases are dispersed uniformly throughout the test domain, as illustrated in Figure 2. Test coverage across the input domain is more complete.

Figure 2. $L_9(3^4)$ orthogonal array

To illustrate the use of the $L_9(3^4)$ orthogonal array, consider the send function for a fax application. Four parameters, P1, P2, P3, P4, are passed to the send function. Each takes on three discrete values.

Given the relatively small number of input parameters and discrete values, exhaustive testing is possible. The number of tests required is $3^4=81$, large, but manageable. All faults associated with data item permutation would be found, but the effort required is relatively high.

The orthogonal array testing approach enables us to provide good test coverage with far fewer test cases than the exhaustive strategy. An $L_9(3^4)$ orthogonal array for the fax send function is illustrated in Table I.

TABLE I. $L_9(3^4)$ ORTHOGONAL ARRAY

Test case (n)	Test parameters (m)			
	P1	P2	P3	P4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Because of the balanced collocation, a factor of any levels and other ones are combined with once, and the only once. It can be conducive to analyze the testing results on account of that.

C. Designing Test Cases on Orthogonal Experimental Method

According to the above analysis, it selects the orthogonal experiment scheme. It designs the test as long as each of the investigated factors are optionally corresponding to the orthogonal table column, and then the number of each

column is corresponding to that of one-to-one factor levels. In this way, the combination of each line of each level constitutes a test condition.

According to the above analysis, the first step is to determine the factor. Given that there are total 6 factors in this module, there are 2, 3, 4 kinds of level. Based on the principle to select the orthogonal table, to select an orthogonal table of 6-factor and 5-level, the number of test cases generated is 25. It is clear that the number of test cases generated by this table is very large, and then it should cut factors and corresponding levels.

According to the product requirement specification, it can analysis the correlation between various factors. Because the relevance of the budget time and the budget cycle is close, it can cut out the factor of the budget time. Similarly the correlation of the reported / issued factor and that of number is also very close, and then can also be cut out. Thus the input parameters remain four, which are as following:

- Budget cycle: Annual budget, Monthly budget
- Budget classification: Sale electricity budget, Purchase electricity budget, Transmission electricity budget
- Unit: T grid provincial company, t Grid city, State grid
- Reported / issued number: Reported number, Issued number

According to the above analysis conditions, to determine four factors, that is the budget cycle, the budget classification, the unit and the reported / issued number, as well as its corresponding level of 2, 3, 3 and 2. The number of experiments is not less than $3 \times (3-1) + 1 = 7$ (s), therefore, it can be considered to use the orthogonal table $L_9(3^4)$ with 4 factors and 3 levels. The factor A, B or C can be corresponding to the orthogonal table $L_9(3^4)$ in any three columns arbitrarily, for example, A, B and C are placed in 1,2 and 3 column, and then test them line by line and in unlimited order. Mapping the determined factors and levels with the orthogonal table $L_9(3^4)$, and filling null value, it can draw a set of test case table, such as shown in table 3.

TABLE II. THE ORTHOGONAL TABLE CORRESPONDING TO TEST PROGRAM - THE TEST CASE TABLE

Test case	Horizontal combination	Test condition			
		Budget cycle	Budget classification	Unit	Reported / issued number
1	A1B1C1D1	Annual budget	Sale electricity budget	T grid provincial company	Reported number
2	A1B2C2D2	Annual budget	Purchase electricity budget	t Grid city	Issued number
3	A1B3C3D3	Annual budget	Transmission electricity budget	State grid	Reported number
4	A2B1C2D3	Monthly budget	Sale electricity budget	t Grid city	Issued number
5	A2B2C3D1	Monthly budget	Purchase electricity budget	State grid	Reported number

6	A2B3C1D2	Monthly budget	Transmission electricity budget	T grid provincial company	Issued number
7	A3B1C3D2	Annual budget	Sale electricity budget	State grid	Issued number
8	A3B2C1D3	Monthly budget	Purchase electricity budget	T grid provincial company	Reported number
9	A3B3C2D1	Annual budget	Transmission electricity budget	t Grid city	Reported number

The analysis of the above process can get 9 test cases, which can cover the critical points of the demand substantially.

IV. AUTOMATIC GENERATION TOOL ON THE ORTHOGONAL TABLE TEST CASE

There are many advantages in use of the orthogonal experiment to design test cases. For example, that can save test time and control the number of test cases generated, and which coverage is good. But in process of designing orthogonal experimental test cases, to find the orthogonal table is very troublesome, so it leads to selecting the orthogonal table with less factor levels. That problem can not be resolved in the artificial stage. In the actual testing work, it is to use an easier orthogonal table. If it is lack of an automatic method, the application of the orthogonal experimental design on test cases is a very complicated and difficult work, so it is necessary to study and improve the method in using process carefully.

A. Code Framework

This is the part of code framework which is designed orthogonal table test cases strategy to the automatic generated tool.

```
// To define a class of orthogonal table:  
public class UniFormTable  
{  
    private int m_Runs;  
    // Rows number is that of rows of the orthogonal table,  
    i.e. the one of tests.  
    private int m_FactorLevelCount;  
    // If it equals 1, that is the the level orthogonal table  
    private int[] m_Factors;  
    // The factor number is that of columns in the orthogonal  
    table.  
    private int[] m_Levels;  
    // The level number is that any single factor can get the  
    maximum value.  
    private string[] m_TableMatrix;  
    // The data of the orthogonal matrix table.  
    private string[] m_TableMatrixString;  
    // The block of orthogonal matrix table data.  
    .....  
    //Case1: The factors number and the levels number are  
    the same.  
    for (int i = 0; i < UniFormTableList.Count; i++)  
    {  
        if (UniFormTableList[i].FactorLevelCount == 1)
```

```

{ //To find the levels and factors number in the
orthogonal table is equals to the specified one.
if (UniFormTableList[i].Factors[0] == MyFactors.Count)
.....
// Case2: The levels number is equal, but factors number
is not equal.
List<UniFormTable> FitTableList = new List<UniFormTable>();
for (int i = 0; i < UniFormTableList.Count; i++)
{
    if (UniFormTableList[i].FactorLevelCount == 1)
    {
        if (UniFormTableList[i].Levels[0] ==
MyFactors[0].Levels.Count)
        { // To find that is more than or equal to the specified
factors number and is the most close to the standard
orthogonal table.
        .....
        // Case3: Each factors number is not the same.
for (int i = 0; i < UniFormTableList.Count; i++)
{
    if (UniFormTableList[i].FactorLevelCount ==
FactorLevelPair.Count)
.....
    // Outputting test cases:
for (int aa = 0; aa < TestCaseTable.Runs; aa++)
{
    for (int bb = 0; bb < MyFactors.Count; bb++)
    {
        MapOutPut.AppendText(TestCaseTable.TableMatrix[aa,
bb].Trim() + " ");
    }
    MapOutPut.AppendText("\n");
}
}

```

B. Automation Tools Generation

It realizes a software, develops a tool of the research and analyzes on the orthogonal experimental to design test cases.

The software provides a solution strategy of three orthogonal experimental design designing test cases that is the same as the factor number and the level number, not the same as one. That realization principle is to read a file contains a number of orthogonal tables. When the program starts, it loads the orthogonal table file and analyses each orthogonal tables, and then finds out factors number and levels number and stores to the orthogonal table object. To explain the inputting contents and to analyses factors and levels, it can find the matching orthogonal table and product the corresponding test cases.

The main function of the software is to input the factor and the corresponding level, and can automatically analyze the suitable orthogonal table to product the corresponding test cases.

V. CONCLUSION

Software testing need analysis and design test cases from different angles, and thus test the system effectively and scientifically. However, due to the limitation of the testing time and resources, it is impossible to test the system completely that is limited to test. That how to distribute limited resources to the system scientifically is an important

topic in software testing. The Orthogonal experimental design is a good solution to such problems.

Using the Orthogonal experimental design to design test cases is the effective means to reduce test cases, thereby improving the efficiency of software testing. This method is most suitable for the software module that contains a plurality of inputting parameters to determine one outputting results together. It studies the designing scheme of the orthogonal experimental design to design test cases, and puts forward the solving strategies on multi-factor levels. To select the right and strong representative points from a large number of test cases, which are more comprehensive and more objective understanding comprehensive tests, and to select the most optimal level combinations. This method can avoid one-sidedness and blindness testing, thereby can improve the efficiency of the software testing and can reduce the cost of it. Practice has proved that such orthogonal experimental design which is “equilibrium dispersion and neat comparable” is a kind of multi-factor testing and effective method. If it will put together a number of test cases designing techniques, the effect would be better.

With the advancement of technology, the software scale is much larger and the complexity increases more and much higher, which bring the new challenges to testers that must master the scientific test cases designing method to apply to the software testing in the actual work, and the orthogonal test method is the best choice by testers undoubtedly.

ACKNOWLEDGMENT

Thank Zhuhai College of Jilin University for supporting the Focus Professional Development Funds.

REFERENCES

- [1] Dashan SUI and Zhenshan CUI, “Application of orthogonal experimental design and Tikhonov regularization method for the identification of parameters in the casting solidification process”, Acta Metall. Sin.(Engl. Lett.)Vol.22 No.1 pp13-21 Feb..2009.
- [2] Fu, Jianping; Lu Minyan; Ruan, Lian; Huang, Baiqiao; “Application of experimental design in software reliability test ” , Beijing Hangkong Hangtian Daxue Xuebao, vol.34, pp. 1379-1383, December 2008.
- [3] Lan, B; Feng, P.F. ; Wu, Z. J. ; Yu, D. W. “Determination of constitutive equation parameters for orthogonal cutting through pressure bar tests and FEA method” , Ken Engineering Materials, v499, p56-61, 2012, Anti-Fatigue Design and Manufacturing Technologies.
- [4] Chan, W.K.; Cheng, M.Y.; Cheung, S.C. ;Tse, T.H. “Automatic goal-oriented classification of failure behaviors for testing XML-based multimedia software applications: An experimental case study”, Journal of Systems and Software, vol 79, pp.602-612, May 2008.
- [5] Qiao, Wen-Feng; Zhou, Yong-Min; “Optimizing technological parameters of preparation of Bi nanoparticles by orthogonal experimental method”, Lubrication Engineering, September 2008..
- [6] Zhou, De-Jian; Huang, Hong-Yan; Peng, Kai-Qiang; “Study on evaluation method for the heat characteristics of IC devices based on the combination of orthogonal experimental method and response surface method”, Proceedings of International Symposium on High Density Packaging and Microsystem Integration 2007, HDP'07.
- [7] Zhang, Qingbiao; Zhou, Kan; Gui, Yingchun; Yu, Chengda; Sun, Ning; Li, Shizhong; “Simulation research of laminated solid armature based on orthogonal experimental method”, 2011 2nd International

Conference on Mechanic Automation and Control Engineering,
MACE 2011 – Proceedings, 1616-1618.