A BINARY TRANSFORMATION APPROACH TO PAIR-WISE TESTING

CHAGARDPON CHIMPONG

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BECHELOR OF SCIENCE

INTERNATIONAL PROGRAMS, FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

This ma... ercial use.
Forbidden to modify the content, and cite the document when use.

COPYRIGHT 2008

FACULTY OF SCIENCE, COMPUTER SCIENCE INTERNATIONAL PROGRAM

KING MONGKUT INSTITUTE OF TECHNOLOGY LADKRABANG

This thesis is allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

A BINARY TRANSFORMATION APPROACH TO PAIR-WISE TESTING

CHAGARDPON CHIMPONG

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BECHELOR OF SCIENCE

INTERNATIONAL PROGRAMS, FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

This r[illegible] for commercial use.
Forbidden to modify the content, and cite the document when use.

Special Project Title	A Binary Transformation Approach to Pair-wise Testing
Name	Chagardpon Chimpong
Degree	Bachelor's Degree of Science
Program	Computer Science
Academic Year	2008
Special Project Advisor	Assoc.P rof. Dr. Veera Boonjing

Abstract

This special project proposes a new approach to pair-wise testing called a binary transformation. The new approach transforms a non-binary input parameter into a binary parameter to be suitable for any test cases generation algorithm of pair-wise testing. The new approach eliminates the restriction of algebraic algorithm that requires some specific pattern of inputs. This special project uses the algebraic binary parameters test cases generation to generate test cases after transforming of input parameters. In order to evaluate the used of this approach, we did the experiment with some problems and compare the result with others approaches and the exhaustive testing. The result of this approach when compare with the exhaustive testing, the number of test cases is less than the exhaustive testing and the coverage of this approach is more than 50% from the exhaustive testing. However, when compare with others pair-wise testing approaches,t he number of test cases is sometime up to twice of the others approaches.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Acknowledgement

I would like to express my gratitude to all people who gave me the possibility to complete this special project.F irst of all,I would like to thanks the Department of Mathematics and Computer Science for giving the permission to commence this special project. I am deeply indebted to my advisor Assoc. Prof.D r.V eera Boonjing from the King Mongkut's Institute of Technology Ladkrabang who stimulates suggestions and encourages me in all the time of research and writing of this special project. My former colleagues and junior colleagues from the Department of Mathematics and Computer Science who supported me in my research work and encourage me when I am lost.I want to thank them for all their help,s upport,i nterest and valuable hints.

Chagardpon Chimpong

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Table of Contents

Page

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

List of Tables

Page

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

List of Figures

Page

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter1

Introduction

1.1 Problem Statement

In software development,s oftware testing is one of most important part which plays the role to ensure that software working properly. Software testing is very expensive and time consuming in order to perform combination testing despite to the large number of test cases. To eliminate this problem, computer scientists come up with many methods and techniques. One of those is pair-wise testing.P air-wise testing states that fault would occur only when 2 parameters interacted. Although pair-wise testing principle is very simple. To generate test cases that satisfy with the principle is not simply since most software has huge number of parameters. Computer scientists developed methods and techniques to generate test cases. Those methods and techniques can be classified into 2 approaches. One is computational approaches and the other is algebraic approaches.

Computational approaches iteratively search the combinations space to generate the required test cases until all pairs have been covered. The computational approaches can be applied to arbitrary system configurations.N evertheless,i n the case where the number of pairs to be considered is significantly large, adopting computational approaches can be expensive due to the need to consider explicit enumeration from all the combination space.

Algebraic approaches construct test sets using pre-defined rules.M ost algebraic approaches compute test sets directly by a mathematical function. Thus,t he computations involved in algebraic approaches are typically lightweight,a nd in some cases,a lgebraic approaches can produce the most optimal test sets. However, algebraic approaches often impose restrictions on the system configurations to which they can be applied. The main restriction that cause inflexible of algebraic approaches to generate a set of test is the input parameters and possible values of parameters have to match with the approaches. The computer scientists solve the restriction by inventing the method with more complexity and even give a low coverage when compare to exhaustive testing.I n this paper,w e propose a new approach to eliminate this restriction by transforming the input parameters into binary parameters to be suitable with

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

the specific algebraic approach. This new approach would promise to give a high coverage and optimal number of test cases when compare to others approaches or the exhaustive testing.

1.2 Objective

The objective of this special project is to propose an approach that eliminates the restriction of the input for algebraic approaches. The new approach would promise to give a high coverage over other approach and the less number of test cases than the exhaustive testing.

1.3 Scope of Study

The scope of this special project is to propose a new approach that transforms the non-binary parameters into binary parameters and generate a set of test cases using binary parameters approach to pair-wise test cases generation. The proposed approach can be applied to any problem.I n order to evaluate the used of the approach,w e did the experiment and use the result to compare with others approaches and the exhaustive testing. The result of the experiment will illustrate the difference between those approaches and the proposed approach.

1.4 Organizations

The rest of this special project consists of 4 chapters as follow:

Chapter 2 provides background of pair-wise testing and the current approaches.

Chapter 3 provides the algorithm of the proposed method.

Chapter 4 gives details of experiments.

Chapter 5 gives conclusion and recommendation.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter 2

Background

2.1 Pair-wise testing

A set of possible inputs for any nontrivial piece of software is too large to be tested exhaustively. Techniques like equivalence partitioning and boundary-value analysis help convert even a large number of test levels into a much smaller set with comparable defect-detection power. Still,i f software under test (SUT) can be influenced by a number of such factors,e xhaustive testing again becomes impractical.

Over the years,a number of combinatorial strategies have been devised to help testers choose subsets of input combinations that would maximize the probability of detecting defects: random testing, each-choice and base-choice,a nti-random and finally t-wise testing strategies with pair-wise testing being the most prominent among these.

Pair-wise testing can be defined as follows:

Given a set of N independent test factors: f_1, f_2, ..., f_N, with each factor f_i having L_i possible levels: $f_i = \{l_i,1, ..., l_i,L_i\}$,a set of tests R is produced.E ach test in R contains N test levels,on e for each test factor f_i,a nd collectively all tests in R cover all possible pairs of test factor levels (belonging to different parameters) i.e.f or each pair of factor levels $l_{i,p}$ and $l_{j,q}$, where $1 \leq p \leq L_i$,$1 \leq q \leq L_j$, and $i \neq j$ there exists at least one test in R that contains both $l_{i,p}$ and $l_{j,q}$.

2.2 Approaches to pair-wise testing

2.2.1 Computational Approaches

Computational approaches often rely on the generation of the all pair combinations. Based on all pair combinations,t he computational approaches iteratively search the combinations space to generate the required test case until all pairs have been covered.

2.2.1.1 AETG

A test tool called Automatic Efficient Test Generator (AETG) was created at Bellcore (now Telcordia). Burroughs,J ain, and Erickson [11] and Cohen, Dalal,K ajla, and Patton [1] report on the use of a tool called Automatic Efficient Test Generator

This material is used for educational use only, not allow for commercial use.
Forbidden to modify the content, and cite the document when use.

(AETG), which contains an algorithm for generating all pair-wise combinations. Cohen later described a heuristic greedy algorithm for achieving t-wise coverage, where t is an arbitrary number. Figure 1 shows an instance of the algorithm that will generate a test suite to satisfy pair-wise coverage.

Assume test cases t_1-t_{i-1} already selected

Let UC be a set of all pairs of values of any two parameters that are not yet covered by the test cases t_1- t_{i-1}

A) Select candidates for t_i by

1) Selecting the variable and the value included in most pairs in UC.

2) Making a random order of the rest of the variables.

3) For each variable,i n the sequence determined by step two, select the value included in most pairs in UC.

B) Repeat steps 1-3 *k* times and let t_i be the test case that covers the most pairs in UC.R emove those pairs from UC.

Repeat until UC is empty.

Figure 1: AETG heuristic algorithm for achieving pair-wise coverage

2.2.1.2 In Parameter Order (IPO)

For a system with two or more parameters,t he in-parameter-order (IPO) combination strategy [3,4] generates a test suite that satisfies pair-wise coverage for the values of the first two parameters. The test suite is then extended to satisfy pair-wise coverage for the values of the first three parameters,a nd continues to do so for the values of each additional parameter until all parameters are included in the test suite.

To extend the test suite with the values of the next parameter,t he IPO strategy uses two algorithms. The first algorithm, horizontal growth,a s shown in figure 2, extends the existing test cases in the test suite with values of the next parameter The second

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

algorithm,v ertical growth,a s shown in figure 3,c reates additional test cases such that the test suite satisfies pair-wise coverage for the values of the new parameter.

Algorithm IPO_H (τ, P_i)

{

Let τ be a test suite that satisfies pair-wise coverage for the values of parameters p_1 to p_{i-1}.

Assume that parameter p_i contains the values $v_1, v_2, \ldots, v_q$

π = { pairs between values of p_i and values of p_1 to p_{i-1}}

if ($|\tau| \leq q$)

{

for $1 \leq j \leq |\tau|$,e xtend the j^{th} test in τ by adding value v_j and remove from π pairs covered by the extended test.

}

else

{

for $1 \leq j \leq q$,e xtend the j^{th} test in τ by adding value v_j and remove from π pairs covered by the extended test

for $q < j \leq |\tau|$, extend the j^{th} test in τ by adding one value of p_i such that the resulting test covers the most number of pairs in π,a nd remove from π pairs covered by the extended test

}

}

Figure 2: IPO_H An algorithm for horizontal growth of a test suite by adding values for new parameters

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
Algorithm IPO_V (τ, π)
{
    Let τ' be an empty set
    for each pair in π
    {
        assume that the pair contains value ω of p_k,1 ≤ k < i, and value υ of p_i
        if (τ' contains a test case with "-" as the value of p_k and u and the value of p_i)
            modify this test case by replacing the "-" with ω
        else
            add a new test case to τ' that has ω as the value of p_k, υ as the value
            of p_i,a nd "-" as the value of every other parameter;
    }
    τ = τ ∪ τ'
}
```

Figure 3: IPO_V An algorithm for vertical growth of a test suite by adding values for new parameters

2.2.2 Algebraic Approaches

Algebraic approaches construct test sets using pre-defined rules.M ost algebraic approaches compute test sets directly by a mathematical function.

2.2.2.1 Two-valued Parameter

The method to generate a set of test cases for two-valued parameter problems was introduced by Maity, Nayak, Zaman, Bansal,a nd Srivastava [12].

Suppose one parameter has strings consisting of zeros and ones. And number of parameter is 2k - 1.L et us define the weight of a string to be the number of ones in it.

Let S_{2k-1} be the collection of all binary strings of length 2k -1 and weight k. And

$$|S_{2k-1}| = \binom{2k-1}{k}$$

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

For example,t he 10 strings of S_5:

0	0	0	0	1	1	1	1	1	1
0	1	1	1	0	0	0	1	1	1
1	0	1	1	0	1	1	0	0	1
1	1	0	1	1	0	1	0	1	0
1	1	1	0	1	1	0	1	0	0

Figure 4: the 10 strings of S_5

If one appends a 0 at the bottom of each column of $S2_{k-1}$,t hen it is possible to conclude that each of the four possible combinations (0 0),(0 1),(1 0) and (1 1) appears at least once in each pair of column and satisfies pair-wise.

Algorithm:

Input: Number of parameters n.

Output: A test set.

1.C ompute smallest k such that $n \leq \binom{2k-1}{k}$

2.C hoose any n strings from S_{2k-1}.

3. Append one zero at the bottom of each chosen string to get a test set of size 2k.

End Algorithm.

Example: Suppose the problem has 5 Parameters and each of them has 2 values as follow:

Table 1: List of parameters for the example

Parameters	Name	Values
1	Operation	{Create, Show}
2	Name	{Empty, Non-Empty}
3	Symbol	{Empty, Non-Empty}
4	Atomic Number	{Invalid, Valid}
5	Properties	{Empty, Non-Empty}

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Step 1: Compute smallest k such that $n \leq \binom{2k-1}{k}$

For k=3, $S_5 = 10$ and for k=2, $S_3 = 3$. Hence the desired integer ***k=3***.

Step 2: Choose any n strings from S_{2k-1}.

0 0 0 0 1

0 1 1 1 0

1 0 1 1 0

1 1 0 1 1

1 1 1 0 1

Figure 5: Show the selected sub strings

Step 3: Append one zero at the bottom of each chosen string to get a test set of size 2k.

0 0 0 0 1

0 1 1 1 0

1 0 1 1 0

1 1 0 1 1

1 1 1 0 1

0 0 0 0 0

Figure 6: Show the selected sub strings after appended the last

Table 2: The test cases after replaced ones and zero with values

	1	2	3	4	5
1	Create	Empty	Empty	Invalid	Non-Empty
2	Create	Non-Empty	Non-Empty	Valid	Empty
3	Show	Empty	Non-Empty	Valid	Empty
4	Show	Non-Empty	Empty	Valid	Non-Empty
5	Show	Non-Empty	Non-Empty	Invalid	Non-Empty
6	Create	Empty	Empty	Invalid	Empty

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

The set of test cases is the set of all rows. For the example the set of test cases consists of 6 cases which are all rows of table 2.

2.2.2.2 Orthogonal Arrays (OA)

Orthogonal Arrays is a mathematical concept that has been known for quite some time. The application of orthogonal arrays to testing was first introduced by Mandl [8] and later more thoroughly described by Williams and Probert [10].

The foundation of OA is Latin Squares. A Latin Square is a V x V matrix completely filled with symbols from a set that has cardinality V. The matrix has the property that the same symbol occurs exactly once in each row and column.F igure 7 contains an example of a 3 x 3 Latin Square with the symbols {1,2 ,3 }.

1	2	3
3	1	2
2	3	1

Figure 7: A 3 x 3 Latin Square

Two Latin Squares are orthogonal if, when they are combined entry by entry, each pair of elements occurs precisely once in the combined square.F igure 8 shows an example of two orthogonal 3 x 3 Latin Squares and the resulting combined square.

1	2	3
3	1	2
2	3	1

1	2	3
2	3	1
3	1	2

1,1	2,2	3,3
3,2	1,3	2,1
2,3	3,1	1,2

Figure 8: Two orthogonal 3 x 3 Latin Squares and the resulting combined square

If indexes are added to the rows and the columns of the matrix,e ach position in the matrix can be described as a tuple $< x,y , z_i >$, where z_i represents the values of the $< x,y >$ position. Figure 4 contains the indexed Latin Square from Figure 9 and Figure 10

This m... allowed For comm... use.
Forbidden to modify the content, and cite the document when use.

contains the resulting set of tuples. The set of all tuples constructed by a Latin Square satisfies pair-wise coverage.

Coordinates	1	2	3
1	1	2	3
2	3	1	2
3	2	3	1

Figure 9: A 3 x 3 Latin Square augmented with coordinates

tuple	Tuple < *xyz* >
1	111
2	123
3	132
4	212
5	221
6	233
7	313
8	322
9	331

Figure 10: Tuples from the 3 x 3 Latin Square that satisfies pair-wise coverage

To illustrate how orthogonal arrays are used to create test cases, consider a test problem with four parameters a,b , c,a nd d, where three values have been chosen as interesting for all parameters.

The orthogonal arrays that match with problem consist of nine runs as it shown as follow.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

	a	b	c	d
1	1	1	1	1
2	1	2	2	3
3	1	3	3	2
4	2	1	2	2
5	2	2	3	1
6	2	3	1	3
7	3	1	3	3
8	3	2	1	2
9	3	3	2	1

Figure 11: The set of test case for the example problem

A test suite based on orthogonal arrays satisfies pair-wise coverage, even after undefined values have been replaced and possibly some duplicate tuples have been removed. This means that the approximate number of test cases generated by the orthogonal arrays combination strategy is v^2_{max}, where $v_{max} = Max^N_{i=1} v_i$, N is the number of parameters, and v_i is the number of values of parameter i.W illiams and Probert [10] give further details on how test cases are created from orthogonal arrays.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter 3

A Binary Transformation Approach

In this chapter, we propose an approach which transforms possible values of all parameters in the original problem to be a binary parameter that consist of zero and one where zero represent that the possible is not used for the test case and one represent that the possible values is used for the test case. After the transformation of parameters,w e can apply binary parameters test cases generation but we need to adapt the technique to match our problem. All of the step would show as follow.

Step 1: Prepare Input Parameters

Let N be a set of parameters for any problem and consists of parameter $X_1, X_2, \ldots, X_n$. For each parameter X_i contains L_i possible values.S uch that L_i consist of $l_1, l_2, \ldots, l_{Li}$.

A new set of parameters is created by transforming every value of L_i for each X_i into binary parameter. That give a number of new parameter set is number of all possible values in N.

For example,

Given a set of test parameter contains 3 parameters X_1, X_2,a nd X_3.

X_1 contains 2 possible values A and B.

X_2 contains 3 possible values C, D, E.

X_3 contains 4 possible values F, G, H, I.

Transforming all possible values for each parameter into binary parameters consist of zero and one. We get a new set of test parameters which consists of 9 parameters {A, B, C, D, E, F, G, H, I}.

Step 2: Construct Binary Strings

Construct a binary strings which $|S_{2k-1}| \geq$ number of parameters with respect to the new set of input parameters.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

From the example in Step 1,

The number of input parameters = 9. Then we use S_5 = 10 and k = 3.W e can use the binary strings in Figure 11.

0	0	0	0	1	1	1	1	1	1
0	1	1	1	0	0	0	1	1	1
1	0	1	1	0	1	1	0	0	1
1	1	0	1	1	0	1	0	1	0
1	1	1	0	1	1	0	1	0	0

Figure 12: The constructed binary strings

Step 3: Select sub binary strings and index each column with input parameters

Select any sub binary strings which number of column equal to number of parameters. After selected sub binary string, we do not need to append the row of zero at the bottom since zeros is used to represent that the parameter is not used in the test case. Then we index each column with each parameter.

From the example in Step 1,

There are 10 difference sub binary strings that can be selected.I n this case,w e select the first ninth column and index each column with the parameters.

	A	B	C	D	E	F	G	H	I
R_1	0	0	0	0	1	1	1	1	1
R_2	0	1	1	1	0	0	0	1	1
R_3	1	0	1	1	0	1	1	0	0
R_4	1	1	0	1	1	0	1	0	1
R_5	1	1	1	0	1	1	0	1	0

Figure 13: table of sub binary after index each row and column

Step 4: Generate test cases

To generate a set of test case, first we generate candidate test cases for every row. In order to generate candidate test cases for each row, we need to consider with the original problem.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Since the possible values of each parameter in the original problem can occur only one. (Two or more possible cannot be occur at the same time) Then we generate the candidate test for each row using permutation and consider with the original problem.

After we obtain all candidate test cases, we eliminate the duplicate test cases.(If we consider it as set,w e union the set of all rows) Finally,w e get the set of test cases.

From the example Step 1,

We can generate candidate test cases as follow.

R_1 cannot generate any test case because there is no valid number with respect to A and B which from the first parameter X_1 in the problem.

R_2 = {BCH, BCI, BDH, BDI}

R_3 = {ACF,A CG, ADF,A DG}

R_4 = {ADG, ADI,A EG, AEI, BDG, BDI, BEG, BEI}

R_5 = {ACF,A CH, AEF, AEH, BCF,B CH, BEF, BEH}

The result of the example in Step 1,

A set of test case = {BCH, BCI, BDH, BDI, ACF,A CG, ADF, ADG, ADI, AEG, AEI, BDG, BEG, BEI,A CH, AEF, AEH, BCF, BEF, BEH}

Number of cases = 20 cases.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter 4

Evaluation

4.1 Problems for experiment

Given 3 problems such that problem 1 has 3 parameters where 2 parameters with 2 possible values and 1 parameter with 4 possible values. Problem 2 has 3 parameters where 2 parameters with 3 possible values and 1 parameter with 4 possible values.P roblem 3 has 4 parameters where 2 parameters with 2 possible values,1 parameters with 3 possible values, and 1 parameters with 4 possible values.

Table 3: The details of experiment problems

	Problem 1	Problem 2	Problem 3
Number of 2 valued parameters	2	0	2
Number of 3 valued parameters	0	2	1
Number of 4 valued parameters	1	1	1
Total number of parameters	3	3	4

The given problems cannot be solved using two-valued parameter method or orthogonal arrays. Anyway,t he given problems can be used the computational approaches to generate a set of test cases.

4.2 Experiment result and evaluation

4.2.1 Coverage

To measure coverage of the set of test cases,w e measure by compare with the exhaustive testing.W e also compare with the computational approaches to illustrate the difference.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Table 4: The coverage of each method

	Problem 1	Problem 2	Problem 3
The Proposed Method	**87.5%**	**69.4%**	**64.6%**
Computational Approaches	**56.3%**	**36.1%**	**25.0%**
Exhaustive Testing	**100%**	**100%**	**100%**

From the experiment result,u sing the proposed method to generate a set of test cases and use the test cases to test the problem,w e can ensure that the faults would be discovered more than 60% of all possible combination cases.

4.2.2 Number of test cases

The coverage is calculated from the number of test cases and compare with the exhaustive testing. The numbers of test cases are shown as follow.

Table 5: The number of test cases for each method

	Problem 1	Problem 2	Problem 3
The Proposed Method	**14**	**25**	**31**
Computational Approaches	**9**	**13**	**12**
Exhaustive Testing	**16**	**36**	**48**

From table 4,t he proposed method cannot give the optimized pair-wise test cases. However the number of test cases that the proposed method gives is less than the exhaustive testing.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

Algebraic approaches to generate a set of pair-wise testing often comes restrictions. The important restriction is algebraic approaches can solve only some specific pattern of problem.I n order to eliminate this problem, we proposed the method that applied the input problem to match the pattern restriction by transforming the input problem into binary problem and use two-valued pair-wise testing to generate a set of test cases. The proposed method overcomes the restriction since the proposed method can be applied to any problem. The advantage point of the proposed method is the coverage of a set of test cases which can used to guarantee the percentage of error that can be discovered. However, the proposed method give more test cases when compare to other methods or approaches which aim to give the optimize number of test cases with only satisfy pair-wise coverage. Other disadvantage point of the proposed method is non-deterministic.B ecause the number of test cases and coverage can be vary due to the constructed binary strings and/or the selected sub binary strings that used to generate the set of test cases.

5.2 Recommendation

This paper only proposed the method that transforms the input problem to the binary problem. The recommendation would be about the other method to apply the input problem for overcoming the restrictions such as applying the input problem to interval problem or fuzzy set. This recommendation might give an idea to apply some other test cases generation method that might give the better result either on coverage or number of test cases.

71946

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

References

[1] Cohen, D.M., Dalal, S.R., Fredman,M .L., Patton, G.C.T he AETG system: An Approach to Testing Based on Combinatorial Design. IEEE Trans.on Software Engineering 23(7),(1997)

[2] Shiba, T., Tsuchiya, T., Kikuno, T.: Using Artificial Life Techniques to Generate Test Cases for Combinatorial Testing.I n: 28th Annual Intl. Computer Software and Applications Conference (COMPSAC 2004), Hong Kong, China (2004)

[3] Y. Lei and K.C.T ai. In-parameter-order: A test generation strategy for pair-wise testing. In Proceedings of the third IEEE High Assurance Systems Engineering Symposium,p ages 254-261. IEEE, November 1998.

[4] Y. Lei and K.C.T ai. A Test Generation Strategy for Pairwise Testing. Technical Report TR-200I-03, Department of Computer Science, North Carolina State University,R aleigh,2001 .

[5] Yan,J ., Zhang, J.: Backtracking Algorithms and Search Heuristics to Generate Test Suites for Combinatorial Testing.I n: Proc. of the 30th Annual Intl.C omputer Software and Applications Conference (COMPSAC 2006),C hicago USA, September 2006, vol.1 ,p ages 385–394.I EEE CS Press, Los Alamitos (2006)

[6] Lei, Y., Kacker,R ., Kuhn, D.R., Okun, V., Lawrence, J.: IPOG: A General Strategy for TWay Software Testing.I n: 14th Annual IEEE Intl.C onf. and Workshops on the Engineering of Computer-Based Systems, Tucson, AZ, March 2007,pp . 549–556.I EEE CS Press, Los Alamitos (2007)

[7] Bush, K.A.: Orthogonal Arrays of Index Unity. Annals of Mathematical Statistics 23,426–434 (1952)

[8] Mandl,R .: Orthogonal Latin squares: an application of experiment design to compiler testing. Communications of the ACM 28(10),1054–1058 (1985)

[9] Hartman, A., Raskin, L.: Problems and Algorithms for Covering Arrays. Discrete Mathematics 284(1-3),149–156 (2004)

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

[10]Williams, A.W., Probert, R.L.: A Practical Strategy for Testing Pair-Wise Coverage of Network Interfaces.I n: Proc.o f the 7th Intl.S ymp.on Software Reliability Engineering (ISSRE),W hite Plains, New York (1996)

[11]K. Burroughs, A. Jain, and R.L. Erickson.I mproved quality of protocol testing through techniques of experimental design.I n Proceedings of the IEEE International Conference on Communications (Supercomm/ICC'94),M ay 1-5, New Orleans, Louisiana, USA, pages 745{752. IEEE, May 1994.

[12]Soumen Maity, Amiya Nayak,M arzia Zaman, Nita Bansal,a nd Alka Srivastava. An Improved Test Generation Algorithm for Pair-Wise Testing.F astA bstractI SSRE 2003

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.