Combinatorial Test Case Generation Using Enhanced Bird Swarm Algorithm

Prof. Swapnil Wani, Mr. Prathamesh Shinde, Mr. Venu Maduta


Abstract- Combinatorial examining is an efficient black box checking approach for the machine with many numbers of parameters and their values. But, for significantly mixed and key element, combinatorial trying out still owns high complexity. Testing completely the center a part of those elements generally is a form of solution and Variable strength combinatorial test data generation (VS-CTDG) emerges. In this task, enhanced bird swarm algorithm (EBSA), a version of bird Swarm algorithm (BSA), is hired into the problem furthermore, a checking out requirement reduction is proposed and makes EBSA extra appropriate into VS-CTDG than ever. via benchmarks, EBSA is proved an effective method.[3].

Keywords- VS-CTDG, EBSA, BSA, Swarm.

I. INTRODUCTION

Software testing is associate degree approach that verifies the consistent between customers expect and code physical object and evaluates the inner correctness of software package. recorder testing that's inclined to code outer feature and white box testing that accustomed judge software inner correctness are presently 2 thought technologies within the fields of software testing. Combinatorial checking may be a reasonably recorder testing that uses sampling mechanism to extract partial test cases from complete test suites to sight code failures caused by parameters and their interactions from System underneath check (SUT).[1]. the idea of combinatorial testing is that system.

Combinatorial testing knowledge generation that constructs optimum covering array may be a analysis stew in combinatorial testing, though effectively reducing the amount of check cases, combinatorial testing usually encounters downside on combination explosion, the big variety of combination of parameters and their price, once applied into real industry’s software[4]. In most situations, t-way combinatorial testing is effective and promising, but some key elements bestowed systems want be tested higher strength, that is additionally known as VS-CTDG[6].

II. AIMS AND OBJECTIVES

a) Aim

Combinatorial testing is an efficient recorder testing technique for the system with massive numbers of parameters and their values. Combinatorial checking could be a reasonably recorder testing that uses sampling mechanism to extract partial test cases from complete test suites to sight package failures caused by parameters and their interactions from system beneath test.

b) Objective

Combinatorial testing data generation, is interested in how to generate the optimal covering array, which not only covers t-way parameter combinations, but also owns minimum coverage number, number of test case in optimal covering array. Therefore, these existing strategies often generate approximate optimal covering array for system under test (SUT) with multiple parameters with large value domain.

III. LITERATURE SURVEY

Combinatorial Testing (CT) can sense failures caused by connections of limits in the Software Under Test (SUT) with a covering array test suite generated by some sample tools.

To ensure successful testing, system should apply CT wisely. This requires professional skill and good judgment in its application. The full strengths and weaknesses of CT need to be better understood. In this system, survey the state of the research of CT. In this, have collected over 90 key papers related to CT. We classify these into eight categories[9].

1. Modeling (Model): Studies on finding the limits, values, and the interdealings of limits of SUT.

2. Testcasegeneration (Gen): Works on processing a tiny test suite efficiently.
(3) Constraints (Constr.): Works on circumvent unwell test cases in test suite generation.

(4) Failure characterization and diagnosis (Fault): Studies on fixing the detected faults.

(5) Betterment of testing methods and the application of CT (App.): Works on experiments testing method for CT and generating the results of the CT application.

(6) Arranging of test cases (Prior.): Works on series of test implementation to get faults as early as possible in the most low-cost way.

(7) Metric (Metric): Studies on computing the mixture reporting of CT and the efficiency of error discovery.

(8) Evaluation (Eval.): Studies on the unit to which CT adds to the perfection of software quality[8].

IV. EXISTING SYSTEM

The System is based on Bird Swarm Algorithm[5]. Bird swarm algorithm (BSA) is a kind of swarm intelligent evolutionary algorithm, which simulates the foraging behavior of bird swarm in nature to solve optimization problems. It employs a group of birds as a candidate solution set, and each bird represents a position in the solution space. Updating the position is used to search the optimal solution through one of the three behaviors: foraging behavior, vigilance behavior and flight behavior. After that, there is a fitness function to estimate bird quality. When the algorithm is initialized, all birds are randomly distributed throughout the solution space. Each bird in bird swarm flies in the solution space according to one of the three behaviors, and gradually converges to the approximate or optimal solution of the problem.

Assuming that the solution space of problem is a dimension real number space, each bird represents a position in the solution space, Note that BSA is originally used for optimization problems of real number space, but combinatorial test data generation is a discrete combinatorial optimization problem that each dimension of bird’s position is an integer. Therefore, when updating the position of birds, the algorithm needs make round operations on computing results[5].

V. COMPARATIVE ANALYSIS

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Paper Name</th>
<th>Author Name</th>
<th>Analysis</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Swarm Intelligence Techniques: Comparative Study Of ACO And BCO</td>
<td>Aditi Chikhalikar, Avanti Darade, Parkar College Goregaon, Mumbai University</td>
<td>Techniques Based Feature Selection Methods Are Simple And Can Be Easily Combined With Other Statistical Feature Select Methods</td>
<td>They Can Execute Commands Like A Multicore Processor</td>
<td>Latency Among Lumps Makes Swiftly Useless To Spread A Calculation More Than Two Or Three Nodes In Chain.</td>
</tr>
<tr>
<td>2</td>
<td>A New Partial Swarm Optimization Algorithm With Adaptive Mutation Operator</td>
<td>Yuelin GAO, North National University, Yin Chuan, China</td>
<td>Is A Heuristic Global Optimization Method and Also an Optimization Algorithm</td>
<td>It Can Be Applied On Both Scientific Research And Engg Use. Is Has No Overlapping and Mutation Calculation.</td>
<td>The process can't solve the issue of dividing and expansion</td>
</tr>
<tr>
<td>3</td>
<td>Resource Scheduling For HF Reception Based On Improved Ant Colony Optimization Algorithm</td>
<td>Yang Liu, Lunwen Wang, Hefei, China</td>
<td>Communication Mechanism Of Real Ants Has Been Transfer To Many Optimization Problems Such As TSP.</td>
<td>Inherent Parallelism, Helpful response report for fast finding of ethical answers, Can Be Used In Dynamic Applications</td>
<td>Probability Distribution Changes By Iteration, Research Is Experimental Rather Than Theoretical</td>
</tr>
<tr>
<td>4</td>
<td>Integrating Greedy Based Approach With Genetic Algorithm To Generate Mixed Covering Arrays For Pair Wise Testing</td>
<td>Prith Bansal, Nitesh Mittal, Netaji Subhash Institute Of Technology, New Delhi</td>
<td>Combinatorial Testing Technique That Generates A Pair Wise Interaction Test Set To Test All Possible Combinations Of Each Pair Input Parameter</td>
<td>The Random Mutation Guarantees To Some Extent That We See A Wide Range Of Solutions, Coding Them Is Really Easy Compared To Other Algorithms</td>
<td>It Might Not Find The Most Optimal Solution To The Defined Problems In All Cases</td>
</tr>
</tbody>
</table>
VI. PROBLEM STATEMENT
Problem is being solved using the Enhanced Bird Swarm Algorithm over Bird Swarm Algorithm for the combinatorial testing. Moreover, a testing requirement reduction is proposed and makes EBSA more suitable into VS-CTDG than ever.

VII. PROPOSED SYSTEM
Enhanced bird swarm algorithm (EBSA), a variant of Bird Swarm algorithm (BSA), is utilized into the matter. Moreover, a testing requirement reduction is projected and makes EBSA a lot more appropriate into VS-CTDG than ever. Combinatorial strategy roughly is classified into 2 categories containing algebraical technique and computing approach, which incorporates one-test-at-a-time (OTAT), input parameter order (IPO), and search-based approach. The project in the main introduces search-based approach that's typically integrated with OTAT framework. The OTAT generates a test suit at a time consistent with greedy strategy to hide uncovered t-tuples as a lot of as doable till uncovered t-tuples are lined. Its input is System below check (SUT) and combinatorial strength and its output is covering array (CA).

i. ALGORITHM
Test case generation
Input: system under test, strength τ, parameters
Output: optimal test case (best)
1: For each bird bi do  #test case is represented
2: bi’s positions initially
3: End for
4: t = 0, best = NULL;
5: While (t < M) #M maximum iteration time
6: For each bird bi do
7: evaluate fitness value and update local optimal and global optimal
8: If fitness value == maximum fitness value
9: return bi
10: End if
11: computing mean distance, best individual, worst individual
12: If (1 % FQ ≠ 0) #FQ frequency of flight
13: For each bird bi do
14: If (rand(0,1) < P) #P forage probability
15: update position by formula(1) and post procedure the value of the location
16: Else
17: update position by formula(2) and post procedure the value of the location
18: End if
19: Else
20: sort swarm by fitness value in descending order
21: divide swarm into two group: top 50% as scrounger, last 50% as producer
22: For each bird bi do
23: If bi in last 25% of swarm
24: update bi position by bird initialization
25: Else if bi in 25% and 50% of swarm
26: update bi position by formula(7) and post procedure the value of location
27: Else
28: update bi position by formula(6) and post procedure the value of location
29: End for
30: t = t+1;
31: End while
32: return best

SYSTEM ALGORITHM
Step 1: Input
1. Product Information (details of product)
2. UserInformation (details of user)
Step 2: Processing
System will determine admin & customer utility by using covering arrays. Apply EBSA strategy to determine testing of products present in the database. Perform product optimization using random list.
Optimization code:
Public static void generatePuzzle (int[][] array) {
    Array = new int[10][5]; //creates array of size N
    //generates random numbers 1 inclusive to # exclusive
    List<Integer> randomList = new ArrayList<Integer>();
    System.out.println("Lenght "+array.length);
    //For loop for 10 times
    for (int i = 0; i <9; i++)
        randomList.add(i + 1);
    //Create a shuffled list of products
    Collections.shuffle(randomList);
    for (int i = 0; i < array.length; i++)
        for (int j = 0; j < array[i].length; j++)
            array[i][j] = randomList.get(j);
    //Create a shuffled list of products
    Collections.shuffle(randomList);
    for (int i = 0; i < array.length; i++)
        for (int j = 0; j < array[i].length; j++)
            array[i][j] = randomList.get(j);
    System.out.println(randomList);
}
Create a payment webserver to integrate with the EBSA system.
Step 3: Desired Output
System will create a list of optimized product by using variable strength covering array, create Test cases depending upon attributes like Web server, Smart device, payment server, database, browser.
Use Hash code to control Check case value.
public int hashCode() {
int hash = 5;
    hash = 53 * hash + (this.webServer != null ?
        this.webServer.hashCode() : 0);
    hash = 53 * hash + (this.smartDevice != null ?
        this.smartDevice.hashCode() : 0);
    hash = 53 * hash + (this.paymentServer != null ?
        this.paymentServer.hashCode() : 0);
    hash = 53 * hash + (this.databaseName != null ?
        this.databaseName.hashCode() : 0);
    hash = 53 * hash + (this.browser != null ?
        this.browser.hashCode() : 0);
    return hash;
}

ii. MATHEMATICAL MODEL

It uses optimal solution by enhancing bird position with three behaviors. Position is updated by any one behavior and gets optimal solution for problem. In the following, the mathematical model of three behaviors are:

For foraging behavior
\[ x_{i,j}^{t+1} = x_{i,j}^t + (p_{i,j} - x_{i,j}^t) * C * \text{rand}(0,1) + (g_{j} - x_{i,j}^t) * S * \text{rand}(0,1) \]  
(1)

For vigilance behavior
\[ x_{i,j}^{t+1} = x_{i,j}^t + A1 * (mean_{i,j} - x_{i,j}^t) * \text{rand}(0,1) + A2 * (p_{k,j} - x_{i,j}^t) * \text{rand}(-1,1) \]  
(2)

\[ A1 = a1 * \exp \left(- \frac{pFl_{l-1} + pFl_{l+1}}{\text{sumFl} + \epsilon} \right) \]  
(3)

\[ A2 = a2 * \exp \left(- \frac{pFl_{l-1} - pFl_{l+1}}{\text{sumFl} + \epsilon} \right) \]  
(4)

For flight behavior
\[ x_{i,j}^{t+1} = x_{i,j}^t + \text{random}(0,1) * x_{i,j}^t \]  
(5)

\[ x_{i,j}^{t+1} = x_{i,j}^t + (x_{i,j}^t - x_{i,j}^t) * FL * \text{rand}(0,1) \]  
(6)

iii. SYSTEM ARCHITECTURE

The Input is being given as the user’s info and the product info. Before processing it is check using covering array. EBSA strategy determines testing of products in database. Later the hash codes are produced.

The system has five components namely Web Server with 2 configurations, Smart Phone with 2 configurations, Payment Server with 2, Database with 3, and Browser with 3. Here components are viewed as factors and configurations as levels.

VIII. ADVANTAGES

- It is simple to use.
- Variable strength combinatorial testing is more realistic technique.
- A new method called EBSA for adaptable for combinatorial test data generation has been presented.
- EBSA, an improved version of BSA, has been evaluated.
- In this EBSA is an effective approach, especially in VSCA containing MCA.

IX. DESIGN DETAILS

Fig.1: System Architecture

Fig.3: EBSA Test Case Result
X. CONCLUSION

We have tried to implement the Author “Lizhi cai, Yang Zhang and Weijia ji” of paper “Variable Strength Combinatorial Test Data Generation Using Enhanced Bird Swarm Algorithm” IEEE 2018. Variable strength combinatorial testing is a more realistic technique that uniform strength combinatorial testing. EBSA, an better variety of BSA, has been evaluated through benchmarks. Experimental results indicate EBSA is an effective approach, especially in VSCA containing MCA.

REFERENCES


