Combinatorial Test Case Generation Using Enhanced Bird Swarm Algorithm

[1]Prof.Swapnil Wani, [2]Mr.Prathamesh Shinde, [3]Mr.Venu Maduta

[1]Asst.Professor,[2,3]UG Student,[1,2,3]Computer Engg. Dept. Shivajirao S.Jondhle College of Engineering & Technology, Asangaon, Maharashtra, India.

[1]swapnilwani24@hotmail.com,[2]prathmeshshinde052@gmail.com,[3]venumad6677@gmail.com

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 mixturereporting of CT and the efficiency of errordiscovery.

(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

Sr. No	Paper Name	Author Name	Analysis	Advantages	Disadvantages
1	Swarm Intelligence Techniques: Comparative Study Of ACO And BCO	Aditi Chikhalikar, Avanti Darade, Patkar College Goregaon, Mumbai University	Techniques Based Feature Selection Methods Are Simple And Can Be Easily Combined With Other Statistical Feature Select Methods	TheyCan Execute Commands Like A Multicore Processor	Latency AmongLumps Makes Swiftly Useless To Spread A Calculation More Than Two Or Three Nodes In Chain.
2	A New Partical Swarm Optimization Algorithm With Adaptive Mutation Operater	Yuelin GAO, North National University, Yin Chuan, China	Is A Heuristic Global Optimization Method and Also anOptimization Algorithm	It Can Be Applied On Both Scientific Research And Engg Use.Is Has No Overlapping and Mutation Calculation.	The process can't solve the issue of dividing and expansion
3	Resource Scheduling For HF Reception Based On Improved Ant Colony Optimization Algorithm	Yang Liu, Lunwen Wang, Hefei, China	Communication Mechanism Of Real Ants Has Been Transfer To Many Optimization Problems Such As TSP.	Inherent Parallelism, Helpful response report for fast finding of ethical answers, Can Be Used In Dynamic Applications	Probability Distribution Changes By Iteration, Research Is Experimental Rather Than Theoretical
4	Integrating Greedy Based Approach With Genetic Algorithm To Generate Mixed Covering Arrays For Pair Wise Testing	Priti Bansal , Nitish Mittal, Netaji Subhash Institute Of Technology, New Delhi	Combinatorial Testing Technique That Generates A Pair Wise Interaction Test Set To Test All Possible Combinations Of Each Pair Input Parameter	The Random Mutation Guarantees To Some Extent That We See A Wide Range Of Solutions, Coding Them Is Really Easy Compared To Other Algorithms	It Might Not Find The Most Optimal Solution To The Defined Problems In All Cases

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 demand reduction is projected and makes EBSA a lot of appropriate into VS-CTDG (check information Generation) 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 $\tau$, parameters
Output: optimal test case (best)
For each bird $bi$ do     #test case is represented
    $bi$ 's positions initially
End for
t = 0, best = NULL;
While(t < M) #M maximum iteration time
    For each bird $bi$ do
        evaluate $bi$ fitness value and update local optimal and global optimal
        If $bi$ fitness value == maximum fitness value
            return $bi$
    End for
    computing mean distance, best individual, worst individual
    If(t % FQ $\neq$ 0)     #FQ frequency of flight
        For each bird $bi$ do
            If((rand(0,1) < P) #P forage probability
                update $bi$ position by formula(1) and post procedure the value of the location
            Else
                update $bi$ position by formula(2) and postprocedure the value of the location
        End for
    Else
        sort swarm by fitness value in descending order
        divide swarm into two group: top 50% as scrounger, last 50% as producer]
        For each bird $bi$ do
            If $bi$ in last 25% of swarm
                update $bi$ position by bird initialization
            Else if $bi$ in 25% and 50% of swarm
                update $bi$ position by formula(7) and postprocedure the value of location
            Else
                update $bi$ position by formula(6) and postprocedure the value of location
        End for
    t = t+1;
End while
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);
      }
 // Shuffle the list again to get different values for the next line
      Collections.shuffle(randomList);
      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 likeWeb server,Smart device,payment server,database,browser.

Use **Hash code** to controlCheck 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

$$x_{i,j}^{t+1} = x_{i,j}^{t} + (p_{i,j} - x_{i,j}^{t}) * C * rand(0,1) + (g_j - x_{i,j}^{t}) * S * rand(0,1) \qquad (1)$$

For vigilance behavior

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \mathrm{A1} * (mean_j - x_{i,j}^{t}) * \mathrm{rand(0,1)} + \mathrm{A2} * (p_{k,j} - x_{i,j}^{t}) * \mathrm{rand}\ (-1,1) \quad (2)$$

$$\mathrm{A1} = \mathrm{a1} * \exp\left(-\frac{pFit_i}{sumFit+\varepsilon} * N\right) \ (3)$$

$$\mathrm{A2} = \mathrm{a2} * \exp\left(\left(-\frac{pFit_i - pFit_k}{|pFit_i - pFit_k| + \varepsilon}\right) * \left(\frac{N * pFit_k}{sumFit+\varepsilon}\right)\right) \qquad (4)$$

For flight behavior

$$x_{i,j}^{t+1} = x_{i,j}^{t} + random(0,1) * x_{i,j}^{t} \quad (5)$$

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

iii. SYSTEM ARCHITECTURE

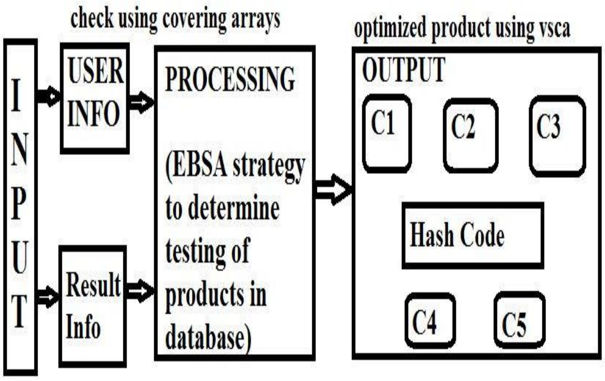

check using covering arrays
optimized product using vsca
INPUT
USER INFO
Result Info
PROCESSING
(EBSA strategy to determine testing of products in database)
OUTPUT
C1
C2
C3
Hash Code
C4
C5

Fig.1: 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.

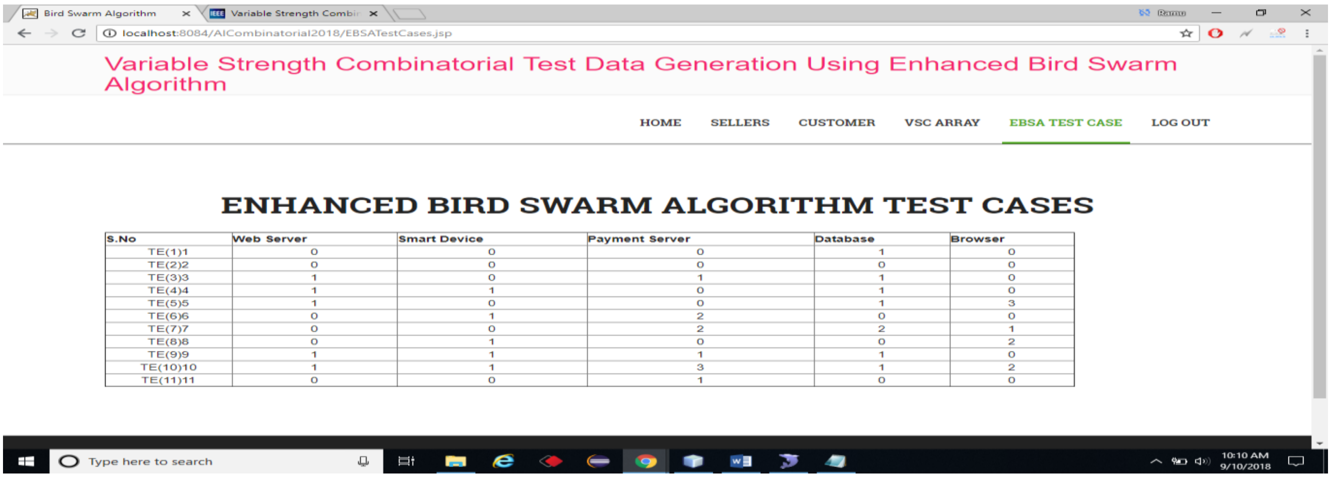
Later the hash codes are produced.

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

VIII. ADVANTAGES

- It is simple to use.
- Variable strength combinatorial testing is more realistic technique.
- A new methodcalled EBSA for adaptableforte 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



Variable Strength Combinatorial Test Data Generation Using Enhanced Bird Swarm Algorithm

ENHANCED BIRD SWARM ALGORITHM TEST CASES

S.No	Web Server	Smart Device	Payment Server	Database	Browser
TE(1)1	0	0	0	1	0
TE(2)2	0	0	0	0	0
TE(3)3	1	0	1	1	0
TE(4)4	1	1	0	1	0
TE(5)5	1	0	0	1	3
TE(6)6	0	1	2	0	0
TE(7)7	0	0	2	2	1
TE(8)8	0	1	0	0	2
TE(9)9	1	1	1	1	0
TE(10)10	1	1	3	1	2
TE(11)11	0	0	1	0	0

Fig.3: EBSA Test Case Result

X. CONCLUSION

We have tried to implement the Author “**Lizhi cai, Yang Zhang and Weijia ji”** ofpaper“**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 bettervariety of BSA, has been evaluated through benchmarks. Experimental results indicate EBSA is an effective approach, especially in VSCA containing MCA.

REFERENCES

[1] Kuhn, D. Richard, and M. J. Reilly. "An Investigation of the Applicability of Design of Experiments to Software Testing." Software Engineering Workshop, 2002.Proceedings.NASA Goddard/ieee IEEE, 2002:91-95

[2] Kuhn, D & Bryce, Renee & Duan, Feng & Sh Ghandehari, Laleh & Lei, yu & N. Kacker, Raghu. (2015). Combinatorial Testing. Theory and Practice. Advances in Computers. 99. 1-66. 10.1016/bs.adcom.2015.05.003.

[3] Cai, Lizhi & Zhang, Yang & Ji, Weijia. (2018). Variable Strength Combinatorial Test Data Generation Using Enhanced Bird Swarm Algorithm. 391-398. 10.1109/SNPD.2018.8441104.

[4] Cohen, M. B, et al. "A variable strength interaction testing of components." ComputerSoftware and Applications Conference, 2003. COMPSAC 2003. Proceedings. International IEEE, 2003:413-418.

[5] Meng X B, Gao X Z, Lu L, et al. A new bio-inspired optimisation algorithm: Bird Swarm Algorithm[J]. Journal of Experimental &Theoretical Artificial Intelligence, 2015:1-15.

[6]Chen, X., Gu, Q., Li, A., Chen, D., 2009. Variable strength interaction testing with an ant colony system approach. In: Proceedings of the 2009 16th Asia-Pacific Software Engineering Conference. IEEE Computer Society.

[7]Niu, X.-T & Nie, C.-H & Chan, A. (2014). Identifying failure-inducing combinations with tuple relationship tree. Jisuanji Xuebao/Chinese Journal of Computers. 37. 2505-2518. 10.3724/SP.J.1016.2014.02505.

[8]Nie, Changhai & Leung, Hareton. (2011). A Survey of Combinatorial Testing. ACM Comput. Surv.. 43. 11. 10.1145/1883612.1883618.

[9] Chen, Xiang, et al. "Applying Particle Swarm Optimization to Pairwise Testing." Computer Software and Applications Conference IEEE, 2010:107-116.

[10] Hartman, Alan, and Leonid Raskin. “Problems and algorithms for covering arrays.” Discrete Mathematics 284.1 (2004): 149-156.

[11] Bansal, Priti, et al. “Construction of variable strength covering array for Combinatorial testing Using Greedy Approach to Genetic Algorithms.” E-Informatica Software Engineering Journal9.1 (2015): 87-105