Automated Testing Approach for Generation and Optimization of Test Cases using Hybrid Bat Algorithm

Rajesh Ku. Sahoo
Department of CSE
ABIT, Cuttack

Durga Pr. Mohapatra
Department of CSE
NIT, Rourkela

Manas Rj. Patra
Department of CSE
Berhampur University

ABSTRACT
Software testing is used to identify error or bugs. Manual testing is a time-consuming process to generate errors. Automated testing is also used for generating the test cases or test data in less time. Generation of test cases identifies the test cases with requirements. Automated generation of test cases has predefined test data which bring into the specified condition through the system under test (SUT). In this paper, the role of hybrid (BCBA) search technique is analyzed for generating and optimizing random automated test cases or test data. This paper also discusses the automatic generation and optimization of test cases from a hybrid bat algorithm which is a combination of a bee colony and bat algorithms applied in withdrawal operation of an ATM.

Keywords
Bee colony algorithm, Bat algorithm, BCBA algorithm, meta-heuristics, test case generation and optimization

1. INTRODUCTION
Software testing is a method which validates the customers' requirements and satisfactions. Testing approach is used to design the test cases through inputs and after execution of test cases, the results are verified [12]. Testing is carried out to find the inconsistencies and ambiguity available in the program specification. Automatic generation and optimization of test cases reduce the time and effort of tester [14]. Test cases may be defined to collect the required data input, performing actions and producing the desired output. Test case development accumulates with requirement specification in a particular path of a program. It is important for the design of test cases, reduce cost and improve the quality of software [16].In optimization technique, various recourses are utilized and every iteration with fitness functional value leads to the target value. Test case optimization has the ability to generate the test case from the system under test (SUT) with minimum errors [13]. Search based optimization methods are used for generating and optimizing the test cases. Various optimization techniques are implemented in different sets of data and take more time to optimize. So there is a scope to improve the result of optimization.

D.D.Karaboga[7] introduced Bee colony algorithm in 2005 and by this technique, the honey bees are searching different food source position through their foraging behavior. The aims of honey bees are to establish places of food source with highest nectar amount. This algorithm is very popular in the computational field. The bees will search for the best position of food source in the hope to get the better result through this algorithm. The food source position represents a possible set of solutions and the amount of nectar represent corresponding fitness values or quality of all solutions or the food source [20].

X.S Yang [9, 10] introduced the bat algorithm in the year 2010 which is having the echolocation with loudness and varying pulse emission behavior of microbats. They emit sounds and bouncing off objects in their path. From this, the bats can determine distance and size of objects. Echo of the microbats is capable of to determine how fast they are moving to find the object. Nearer they get closer to their prey, the better is the solution. This approach is used to replace the old solution with a new improved solution. This proposed work emphasizes on the appropriate hybrid optimization technique which gives a better result which is optimal.

The proposed approach uses the hybrid search technique that combines bat and bee colony algorithm where generated and optimized test cases are improved the design quality of the software. This paper represents the automated generation and optimization of test cases by using bee colony bat algorithm (BCBA).

The rest of the paper is organized as follows. Section 2 discusses basics of software testing, an overview of bee colony algorithm, bat algorithm, and BCBA hybrid algorithm.Section 3 is for related work on automated test case generation and optimization. Section 4 represents the proposed systems, methodology and working principle of proposed approach. Section 5 focuses on the simulation results. Section 6 focuses on the discussion and future scope and Section 7 concludes the paper.

2. BASIC CONCEPTS
Software testing emphasizes on the test case design with inputs, after executing the test cases, the results are examined. It is very important that test case generation is based on the software specification and system implementation algorithms. A test case describes how an accurate result comes from the set of inputted data. Software testing is a method by processing the well-defined input data and having the capability to verify the failures. At present software testing takes 55% of the total development cost. Automatic generation of test cases has a predefined test data which bring into the specified condition through the system under test(SUT).This approach is based on the specific criteria of test coverage where the proper test data is determined from the generation of test cases.

2.1 Overview of Bee colony Algorithm
Bee Colony algorithm (BCA) is an evolutionary based method which derived from the bees' behavior. It is developed by Dervis Karaboga[11] for optimization purpose in 2005. Bee colony method states that the bees are found their food source through their foraging behavior. The aims of honey bees are to establish places of food source with highest nectar amount. This algorithm is very popular in the computational field. According to this method, the bees will search for the best position of food source which gives a better result. The positions of food source represent a possible set of solutions and the nectar amount emphasizes on corresponding fitness values or quality of all solutions of the food source.

2.2 Overview of Bat Algorithm

Bat algorithm was introduced by X.S Yang in 2010[9, 10].This method is emphasized on the echolocation, loudness and varying pulse emission behavior of microbats [12]. Frequency is transformed from sound pulses which used by microbats. Microbats use the sound pulses which are converted to frequency and the frequency is echoed. The duration between the sound emitted by microbats and the echo may be delayed. According to this algorithm, the position of all bats represents a possible set of solutions and their distance between the bats and prays represent fitness values or quality of all solutions.

2.3 Overview of BCBA Hybrid Algorithm

The proposed hybrid Algorithm is created or developed by merging the Bee Colony Optimization Algorithm with the approach used in bat Algorithm. Here total population of the candidate solution is subdivided into two parts. One part of the solution undergoes BCA and another part undergoes Bat optimization algorithm. The advantages of this algorithm are for its implementations in complex functions with mixed, random and discrete values.

3. RELATED WORK

Sharma et al. [1] focused on a technique that generates test cases from sequence and uses case diagram. Then use case and sequence diagram converted into use case and sequence diagram graph or system testing graph. Tripathy et al.[2] presented how the test cases are generated from activity diagram and sequence diagram. Then activity diagram and sequence diagrams are converted into activity diagram and sequence diagram graph. Lastly, system testing graph is traversed by DFS to generate the test cases by taking an example i.e. validation of ATM card. Swain et al. [3] proposed a technique called SATEC which uses both state chart diagram and activity diagram for generating the test class to achieve state-activity coverage. After generating the test cases it evaluates according to mutation analysis with mutation operations. According to Swain et al. [4] generation of test cases are done through activity control flow graph which derives from activity diagram. Activity flow graph (AFG) is traversed by DFS technique which generates all possible paths with path coverage. Dalai et al. [5] presented an approach that is sequence activity graph which traversed and generates the test cases. Debasish Kund and Debasish Samanta [6] described how the activity diagram is converted into activity graph to generate the test cases. From the activity graph, the generation of test cases is applied through DFS and BFS algorithms. After applying this algorithm it gives simple path coverage and activity path coverage. Pakinam N et al. [7] focused on the behavioral model which generates the test cases also describes basis path testing with cyclomatic complicity. It is used to generate test cases with less time and effort. Wang Lindhang et al. [8] proposed that generated test cases through MBT are Specification and code based. Model-based testing uses the requirement specification as input and generates the test cases which gives the architectural configuration of software. Sahoo et al. [15] described how the automated test cases are generated and optimized by firefly algorithm through the withdrawal operation of an ATM. According to swain, et al. [18] test data is generated from activity diagram. The activity diagram is converted to activity flow graph through all activities, executions, loops, start and end of the action. Sahoo et al. [17] explained how the automated test data are generated by harmony search algorithm by taking an example of ATM withdrawal operation. Suresh et al. [19] explained how test data are generated based on genetic algorithm through basis path testing. It converts the sequence diagram into sequence flow graph and generates automated test cases or test data through a genetic algorithm. Sahoo et al. [21] focused on the automated course timetable generation and optimization by a hybrid firefly approach which takes less iteration to optimize in comparison to bee colony algorithm and firefly algorithm. Arvinder Kaur et al. [22] described how the hybrid particle swarm optimization (HPSO) was used in regression testing efficiently with an appropriate feasible solution.

4. PROPOSED SYSTEM

This paper proposes a methodology for generating and optimizing the automated test cases by taking an example of withdrawal operation of an ATM machine. Automated test cases are generated and optimized by bee colony bat algorithm (BCBA). This method is used for evaluating its efficiency and effectiveness for generating the test cases and for maximizing to achieve the goal.

4.1. Necessity of Proposed System

The proposed system is intended to generate and optimize the automated test cases using bee colony bat algorithm (BCBA). In the case of BCBA, all the system may be initialized with the idea that honey bees will search for a better food source position which gives a better result. Bat Algorithm (BA) is conceptualized by using the behavior like echolocation of bats with a variation of pulse rate emission and loudness to get the best result. BCBA is a combination of a bee colony and bat algorithms which may generate the optimum solution. This paper also aims at finding out the effectiveness of the proposed approach through numbers of test cases or test data.

4.1.1Proposed approach and working of proposed approach

The proposed hybrid Algorithm is created or developed by merging the Bee Colony Optimization Algorithm with the approach used in bat Algorithm. Here total population of the candidate solution is subdivided into two parts. One part of the solution undergoes BCA and another part undergoes Bat optimization algorithm. The advantages of this algorithm are for its implementations in complex functions with mixed, random and discrete values.

BCBA (Pseudo code test cases or test data generation by using BCBA Hybrid Approach)

Specify total the number of generation.
Specify population size.
Generate initial candidate solution
Evaluate its fitness function value 'fx'
$fx=1/(abs(net\_bal-wd\_amt)-min\_bal)+\varepsilon)^2$
Where ε varies from 0.1 to 0.9
Find the initial best solution
While generation(t)<500 do
Rank the solutions
Discard the bottom half solutions having worst fitness values
Top half best solutions undergoes operation in two phases separately
Make two copies of best solutions.
One copy undergoes Bee Colony Optimization i.e., Phase 1
Another copy undergoes Bat Algorithm optimization i.e., Phase 2

Phase 1
//Employed Bee Phase
Produce new candidate solution

Check the boundary conditions
Evaluate its fitness value
If(finess(new)>fitness(old))
then replace the older solution
//Probability Calculation Phase
Calculate the probability of occurrence of each solution P
//Onlooker Bee Phase
If P>rand()
Produce new candidate solution
Check the boundary conditions
Evaluate its fitness value
If(finess(new)>fitness(old))
then replace the older solution
End If
End If

Phase 2
Generate new solution by adjusting bat's frequency and updating their velocities and its position
if rand> pulse rate 'r'
Generate a local solution around the best solution
Check the boundary conditions
Evaluate its fitness value
If(finess(new)>fitness(old) and rand <A)
then replace the older solution
End if
End if
Memorize the best solution
Generation(k)=Generation(k)+1
Generate the current best solution
End While
Select the best solution having the best fitness value

Phase 1:- (Bee Colony Optimization)

The new solution can be calculated as

c=x(j)+ebf*x(j) (1)

where x(j)= candidate solution at j^{th} position
ebf = a random number in the range of [-1,+1]

The probability of occurrence for each candidate solution is calculated as follows :

prob(j)=fx(j)/tfx; (2)

where prob =probability factor
fx(j)=fitness function value
tfx =total fitness value of all candidate solution

In Onlooker Bee phase, the solution having probability greater than a random value in the range of [0, 1] are selected and there corresponding solutions are improved with the help of the following equation:

v(j)=x(j)+ebf*x(jj); (3)

Where ebf=a random number is in the range of [-0.1,+0.1]

Phase 2:- (Bat Algorithm)

The movement of bats can be calculated as

x1= x+v1; (4)

where x= candidate solution
x1=new bat position
v1=new velocity value

The movement of a Bat toward another is calculated as follows:

v1=v+(xbest-x)*f/vmin; (5)

Where v1=updated velocity of bats

xbest=current best solution

x=candidate solution

f=bat frequency

vmin=minimum bank balance

The bat frequency can be calculated as :

f=(fmin+(fmax-fmin)*rand)/vmin; (6)

Where fmin=minimum frequency value specified

fmax=maximum frequency value specified

rand=a random value in the range of [0,1]

Here fmin=0

fmax=vmin

The flow chart of hybrid BCBA is depicted in Figure 1.

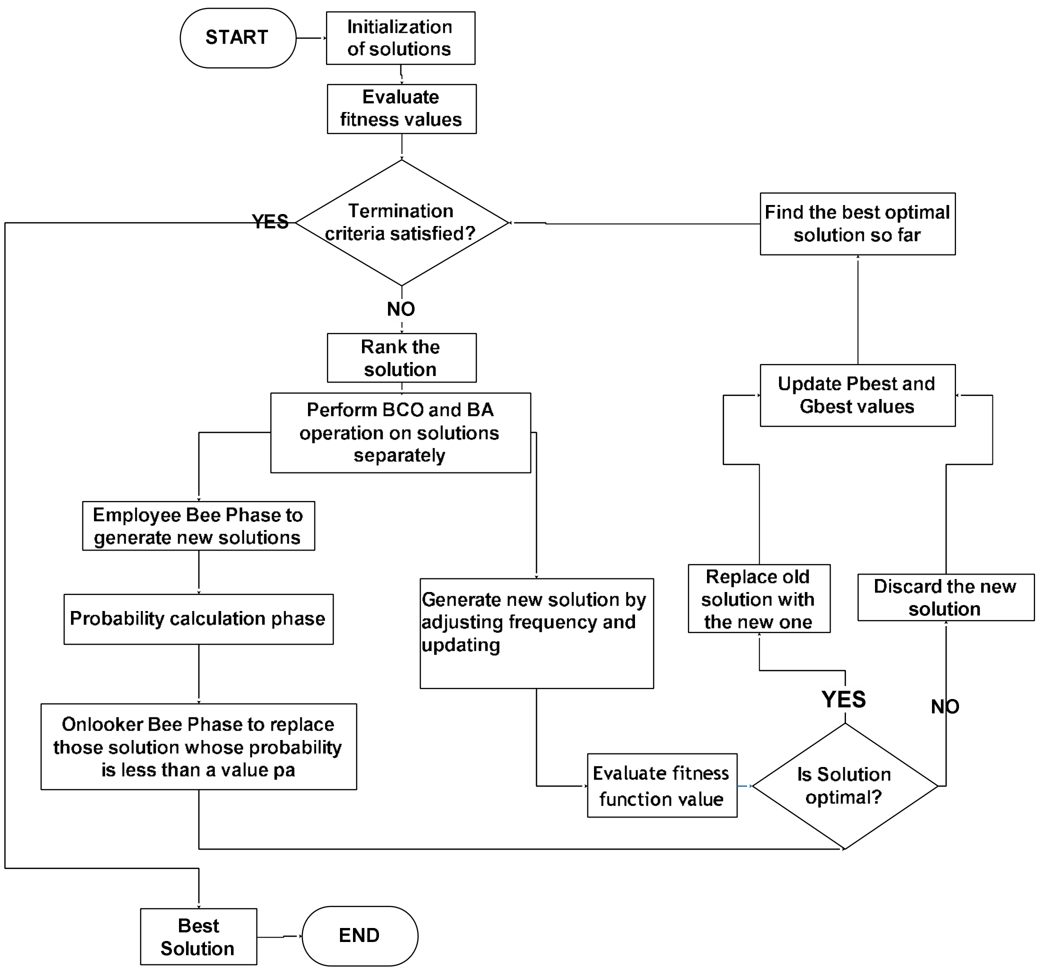

Fig. 1: Flowchart of test case generation using BCBA hybrid approach

4.2. Methodology

For Mathematical function

$f(x)=1/(abs(suc\_bal)+ \varepsilon)^2$ (7)

Where $0.1<= \varepsilon <0.9$ (taking ε -value because overflow condition due to infinity).

Here Successive Amount (suc_amt) is defined as:

suc_bal = net_bal - (wtd_amt - min_bal) (8)

Where net_bal = current account balance

min_bal= minimum bank balance limit

Initially, the number of solutions or population size and the maximum number of iterations or generations are provided by the user. After that, an initial population is generated randomly and their corresponding fitness values are calculated and stored. The initial best optimal solution is calculated. Then the candidate solutions are sorted in terms of their fitness functional values. Higher the fitness value more the solution tends toward optimality. After the sorting operation, the bottom half worst solution is discarded and are replaced with a copy of top half best solution found so far. Then both copies of top half best solution undergo two different phases of optimization techniques. In this case the first phase i.e., in Phase 1, the candidate solutions undergoes Bee Colony Optimization (BCO) and another copy of candidate solution undergo the second phase i.e., Phase 2 Bat Algorithm (BA).Phase 1 of BCO is subdivided into two more phases i.e., employed bee phase and onlooker bee phase. In employed bee phase a new solution is generated and checked if the fitness function values of the new candidate solution are better than the old existing solution or not. If the solution is having a better solution than the old solution, then it replaces by the new solution. After Employed Bee Phase, the relative fitness value of each candidate solution is calculated. In Onlooker Bee phase, the candidate solutions having a relative value less than a specific constant value ‘pa’ then that solution is discarded from the memory and is replaced with a newly generated random solution. In Phase 2, the candidate solutions are updated by adjusting its frequency, position, and velocity of the microbats. If the fitness function values of the new candidate solution are better than the old existing solution, then the old solution is replaced by the new solution. Then new better solutions are created around the current solution with the help of pulse rate value. After the completion of two phases of optimization, the current best solution is memorized. The results gained from both phases are merged. Again all the candidate solutions are sorted and the bottom half worst solution is discarded and is replaced with a copy of top half best solution. Then both copies of top half best solution undergo in two phases and the programs iterates until termination criteria are satisfied. The solution produced so far is the best optimal solution.

The Bee Colony Algorithm Employed and Onlooker Bee will search for an optimal solution. It will keep track of the population based solution and upgrade its position or location. According to Bat Algorithm, the position of all bats represents a possible set of solutions and their distance between the bats and prays which represent fitness values or quality of all solutions. In BCBA hybrid approach combined the Bee Colony and Bat algorithm which gives the optimal solution to maximize the mathematical function f(x).It may be implemented using MATLAB-7.0 as shown in “table 1”. This table primarily focuses to generate the best solution in the search space.

Table 1.Fitness Function Value for each sample space or test case

	Bee Colony Algorithm		**Bat Algorithm**		**Bee Colony Bat Algorithm (BCBA)**	
Iteration Number	**Test cases/Test data**	**Fitness Function Value**	**Test cases/Test data**	**Fitness Function Value**	**Test cases/Test data**	**Fitness Function Value**
1	4200	6.0073e-010	4500	6.0966e-010	4800	6.1879e-010
10	6700	6.8171e-010	8500	7.506e-010	6400	6.7115e-010
20	6900	6.8888e-010	9700	8.025e-010	8400	7.4651e-010
30	9600	7.9798e-010	14000	1.0406e-009	12400	9.4094e-010
40	15000	1.1111e-009	21300	1.7803e-009	20000	1.6e-009
50	19500	1.5379e-009	24400	2.3565e-009	24400	2.3564e-009
60	20400	1.6524e-009	30400	4.6912e-009	28400	3.6288e-009
70	23000	2.0661e-009	32900	6.83e-009	32800	6.7184e -009
80	26400	2.8905e-009	34500	9.07e-009	36400	1.352e-008
90	30400	4.6912e-009	35800	1.1814e-008	38700	2.5194e-008
100	35300	1.0628e-008	37600	1.8261e-008	39300	3.0777e-008
110	37800	1.9289e-008	40400	4.7256e-008	41000	6.2495e-008
120	39700	3.5598e-008	40900	5.9484e-008	42300	1.3716e-007
130	40800	5.6685e-008	41500	8.1626e-008	44000	9.9975e-007
140	42100	1.1889e-007	41800	9.765e-008	44000	9.998e-007
150	43800	6.9427e-007	42600	1.7359e-007	44000	9.998e-007
160	44000	9.997e-007	43500	4.4438e-007	44000	9.998e-007
170	44000	9.998e-007	43600	5.1013e-007	44000	9.998e-007
180	44000	9.998e-007	44000	9.996e-007	44000	9.998e-007
200	44000	9.998e-007	44000	9.997e-007	44000	9.998e-007

5. SIMULATION RESULTS

The proposed approach generates the automated test cases through test cases or test data for Bank ATM by using BCBA hybrid algorithms. The “figure 2” shows the relation between two variable quantities which are iteration numbers and test cases or test data values measured along one of a pair of axis represented in “table 1”.

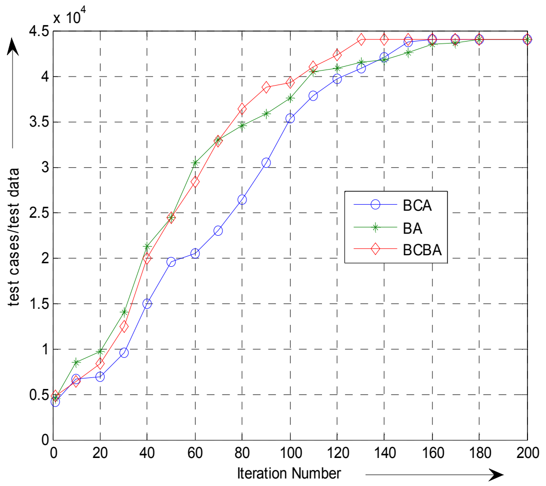

Fig 2: Graphical representation of iteration numbers and test cases or test data and values for table 1

The proposed approach generates the test cases or test data for ATM’s withdrawal operation using hybrid bat algorithm (BCBA). Table 2 shows the value range of fitness functional value with test cases/test data. The functional value of fitness function is represented in terms of percentage. After evaluation, it was found that using bee colony algorithm the optimal solution is achieved after 160 iterations whereas by using bat Algorithm the optimal solution is achieved after 180 iterations. But by implementing the hybrid bat approach (BCBA) it was observed that the optimal result is achieved around 130 iterations. The proposed approach generates the test case or test data for Bank ATM’s withdrawal operation using bee colony, bat, and BCBA algorithm. “Table 2” represents the range of fitness functional value with different test cases or test data and also it gives the individual candidate solution according to the fitness functional value range in terms of percentage.

Table 2: %of test cases/test data in terms of maximum fitness value

FITNESS VALUE RANGE	**% OF TEST CASES/TEST DATA(BCA)**	**% OF TEST CASES/TEST DATA(BA)**	**% OF TEST CASES/TEST DATA(BCBA)**
0 $\leq$ f(x) < 0.3	40	30	35

$0.3 \leq f(x) < 0.7$	35	35	25
$0.7 \leq f(x) < 1.0$	25	35	45

The above table shows that around 45% test cases or test data are having the higher fitness function f(x) value and lies in between 0.7 and 1.0 by using hybrid bat algorithm (BCBA) but in case of bee colony algorithm and bat algorithm, only 25% and 35% of test cases or test data are available within the higher range of fitness function.

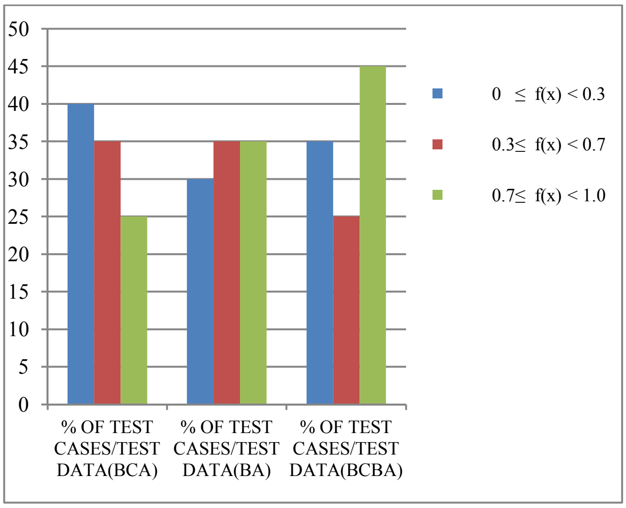

Fig 2: Graphical representation of % test case or test data and fitness value range for table 2

By considering all the functional value of fitness function from "table 2", hybrid bat algorithm is having higher fitness value range as compared to bee colony and bat algorithm. "Figure 2" shows a pictorial representation of the relation of two variable quantities like percentage of test cases or test data and fitness value range.

6. DISCUSSION AND FUTURE SCOPE

By considering the mathematical function $fx=1/(abs(net\_bal\text{-}wd\_amt)+\varepsilon)^2$, where ε varies from 0.1 to 0.9, this proposed paper generates and optimized the test cases as well as test data automatically through bee colony algorithm, bat algorithm and combinations of a bee colony and bat algorithm(BCBA). By considering some sample test cases it has been observed that the functional value depends upon the parametric values of the input variables and food source position. Bat algorithm generates the optimized test cases with test data which would also have the coverage of path. The input test data is given through SUT along a path. Microbats find their position and compute the functional value of the fitness function. This proposed BCBA algorithm is optimized the automated test cases or test data. For any algorithm implementation first, the algorithm is converted into pseudo code before the application developed. The optimum value is obtained from Bat algorithm comes from Bee Colony Algorithm (BCA).In future, the different hybrid technique will be used which handle various errors or faults with better path coverage by using UML diagram graph.

7. CONCLUSION

Software testing meets the user requirements and conditions. This paper also describes the fundamental notions of BCA, BA, BCBA and how the test cases are automatically generated and optimized. This proposed approach optimized the test cases which are maximized with minimum iterations and time. According to result, it is analyzed that BCBA technique gives better result in comparisons with Bee Colony algorithm and Bat algorithm. The design quality can be improved and multiple generated test cases can be used for execution. By using BCBA hybrid technique test cases are optimized and get the optimum result.

8. REFERENCES

[1] Monalisha Sharma, Debasish Kundu, Rajib mall.2007, "Automatic Test case Generation from UML sequence Diagrams" the proceeding if IEEE conference on software maintenance.

[2] Abinash Tripathy and Anirban Mitra.2013, "Test case generation using activity Diagram and sequence Diagram" proceeding of ICADC, AISC 174, pp. 121-129, Springer.

[3] Santosh Kumar Swain and Durga Prasad Mohapatra.2010 ,"Test case Generation from Behavioral UML models",International Journal of computer applications (0975-8887) vol.6,No.8,pp.5-11.

[4] Ranjita Kumari Swain, Vikash Panthi, Prafulla Kumar Behera,2013,"Generation of test cases using activity Diagram" International journal of computer science and informatics, ISSN(PRINT); 2231-5292, vol.3, Issue 2,pp.1-10.

[5] Swagatika Dalai, Arup Acharya, and Durga Prasad Mohapatra.2012, "Test case generation for concurrent object-oriented systems using combinational UML models ",IJCA,Vol.3,No.5,pp.97-102 .

[6] Debasish Kundu, Debasish Samanta.2009,"A novel Approach to generating test cases from UML activity Diagrams", in the journal of object technology,vol-8,no-3,pp.65-83.

[7] Pakinam N.Boghdady, Nagwa L.Badr, Mohammed Hashem, and Mohamed F.tolba,2011,"A proposed test case generation technique based on activity Diagrams", International journal of Engineering & Technology IJET-IJENS Vol.11 No.03,pp.35-52.

[8] Wang Lindhang et al.2004, Generating Test cases from UML activity diagram based on Grey based method, Asia-Pacific software engineering conference, Nov-dec,2004,pp.284-291.

[9] X.S. Yang.2010, A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65–74.

[10] X.S. Yang.2011, Bat algorithm for multi-objective optimization. International Journal of Bio-Inspired Computation, 3(5):267– 274.

[11] D. Dervis Karaboga,2005, An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

[12] Ravi North and Kailash KP Chandrika.2008, Model-based automated Test case generation, SET labs Briefings, Vol.6, No 1, pp 39-46.

[13] Lionel Briand and Yvan Labiche,2001, A UML-Based approach to system Testing, lecture Notes in computer science; Springer-Verilog, LNCS 2185,pp.194-208.

[14] S. Singla, D.Kumar, H.M. Rai, P.Singla,2011, A hybrid PSO approach to automating Test data generation for data flow coverage with dominance concepts, International journal of advanced science and technology, Vol.37,pp.15-26.

[15] Rajesh Kumar Sahoo, Durga Prasad Mohapatra, Manas Ranjan Patra,2016," A firefly Algorithm Based Approach for Automated Generation and Optimization of Test cases", International Journal of Computer Sciences and Engineering, vol.4, Issue-8,pp.54-58,ISSN:2347-2693.

[16] S .S. Priya and P.D.Sheba.2013, "Test case Generation from UML models- A Survey," In Proc. International conference on information systems and computing(IC ISC-2013). India, January 2013, Vol.3, Special issue.1.

[17] Rajesh Kumar Sahoo, Deeptimanta Ojha, Durga Prasad Mohapatra, Manas Ranjan Patra,2016,"Automatic generation and optimization of test data using harmony search algorithm", ACITY,pp.23-32.

[18] R. Swain, V. Panthi and P. Behera.2013, "Generation Of Test Cases Using Activity Diagram", International Journal of Computer Science and Informatics, vol. 3, issue. 2, pp.1-10.

[19] Yeresime Suresh, Santanu Ku.Rath.2014,"Evolutionary Algorithms for object-Oriented Test Data Generation", ACM SIGSOFT Software Engineering Notes, Vol. 39, No. 4,pp.1-6.

[20]Rajesh Kumar Sahoo, Deeptimanta Ojha, Durga Prasad Mohapatra, Manas Ranjan Patra,2016,"Automated Test case Generation and optimization: A Comparative Review", International Journal of Computer Science & Information Technology, Vol.8, No.5,pp.19-32.

[21]Rajesh Kumar Sahoo, Deeptimanta Ojha, Durga Prasad Mohapatra, Manas Ranjan Patra,2017,"Automatic generation and optimization of course timetable using a hybrid approach", Journal of Theoretical and Applied Information Technology, Vol.95, No.1,pp.68-77.

[22]Arvinder Kaur, Divya Bhatt,2011,"Hybrid particle swarm optimization for regression testing", International Journal of Computer Science and Engineering(IJCSE), Vol.3, No.5,pp.1815-1824.