INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

AN APPROACH FOR SOFTWARE TEST CASE SELECTION USING HYBRID PSO

[1]Preeti Bala Thakur, [2]Prof. Toran Verma

[1]Dept. of Software Engineering, Rungta College of Engineering & Technology. Bhilai, India
E-Mail: preety3004@gmail.com
[2]Dept. of Computer science & Engineering, Rungta College of Engineering & Technology. Bhilai, India
E-Mail: toran.verma@rungta.ac.in

Abstract: - Software Testing is a continuous process of development and maintenance. In maintenance phase, regression testing gets exercised with additional resources for performance. The prioritization of test cases helps to reduce the cost-time of regression testing. Hence, completed Regression testing effectively. During software testing process many test suites can be generated in order to evaluate. In some cases, the execution of all suites cannot fit the available resources. Hence, automatic Test Case (TC) selection process could be used to reduce the suites based on some selection criterion. This process can be treated as an optimization technique problem, aiming to find out a subset of TCs which optimizes one or more objective functions (i.e., selection criteria). This Project Objective is Test Case Selection in Software Testing using Combination of Optimization Techniques Based on Particle Swarm Optimization and Ant Colony Optimization.

Keywords—Test Case Selection, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Software Testing.

I. INTRODUCTION

During maintenance phase, modifications or defects in software are corrected in SDLC. It is very difficult to decide which test case should be executed, or which test case should be mark as effective. Regression Testing is done within maintenance phase in software testing. Regression Testing assures that modification is done in software has not over-ruled user specification and expectations. To make Regression Testing more scheduled many criteria‘s can be followed reduction of test case, selection of test case, prioritization. During the software development process, the software testing activities has grown in importance due to the need of high quality products. Software products should be reliable, correct, and scalable. it is necessary to test the software at various conditions and software testing is an important component of the software development process.

There are three main activities associated with software testing: (1) test data generation, (2) test execution involving the use of test data and the software under test (SUT) and (3) evaluation of test results. In software testing is how to select test cases with the aim of uncovering as many defects or error as possible. Since exhaustive testing is impossible in terms of cost, and no realistic amount of systematic testing can guarantee the absence of errors, the key question is when and how do we determine whether testing has been conducted adequately or not. In order to reduce cost and time as well as to improve the quality or performance of the software, any extensive testing would require the automation of testing process. The three activities mentioned test data generation and evaluation of test results are the most labor intensive and thus would benefit most from automation.

i) Ant Colony Optimization –

ACO simulates the behavior of real ants. The first ACO technique is known as Ant System and it was applied to the travelling salesman problem. Many variants of this type of technique have been produced. ACO is a probabilistic optimization technique that can be applied to generate solutions form combinatorial optimizations problems. The artificial ants in the algorithm shows the stochastic solution construction procedures which make use of the dynamic evolution of the pheromone trails that reflect the ants' acquired search experience. The heuristic information related to the problem, in order to construct probabilistic solutions. In order to apply ACO to test case generation & selection process, a number of issues need to be addressed, namely, transformation of the testing issue or problem into a graph a heuristic measure for measuring the "goodness" of paths through the graph; a mechanism for creating possible solutions efficiently and a suitable criterion to stop solution generation; a suitable method for updating the pheromone; and a transition rule for determining the probability of an ant traversing from one node to other node in the graph to the next. In the next section, we will present an ACO approach to automatically generate test data from UML State chart diagrams for state-based software testing.

ii) Particle Swarm Optimization –

Many researchers have explored area of PSO and it has been used to solve routing optimization problems where the optimized path for traffic routing, packet routing is search. The scheduling problems such as: job-scheduling task scheduling problems in distributed systems where task or job is divided into subsets to make performance cost efficient. The PSO has been applied in the field of cryptography and crypt-analysis for military applications such as code encryption and security. The traveling salesman problems is to find optimal path to reach from source to destination with less time and cost, that make into-hard combinatorial problem has been solved using PSO technique and other combinatorial problems such as: packing. The applications of detection of fault and recovering from them includes field: test pat tern generation for circuits, software faults detection had been applied by PSO technique.

II. LITERATURE REVIEW

Chengying Mao(2012) have reformed the basic Ant Colony Optimization to generate better quality test data for the purpose of earlier fault- revealing .Four rules have been re-defined to achieve the objective namely: the local transfer rule, the global transfer rule and pheromone update rule. The criterion used is the branch coverage. The result of the reformed Ant Colony Optimization outperforms the algorithms namely: Simulated Annealing and genetic algorithm.

Nirmal Kumar Gupta(2013) have proposed a strategy that uses genetic algorithm to reduce the number of unfeasible test cases and establishing genetic algorithm to generate suitable test cases and hence improving the genetic algorithm .The genetic algorithm is used to generate the test cases for Object Oriented Software that is for java programs. But practically to use genetic algorithm there is a problem as there is local convergence problem.

Osman Gokalp (2012) reformed the original Ant Colony Algorithm by employing crossover mechanism in genetic algorithm. Four ant colonies have been employed rather than one colony and it also employs pheromone tables .Each colony will have their own pheromone tables and the best pheromone tables with best solution is chosen. The reformed algorithm is tested using Traveling Salesman Problem. The result of the reformed algorithm outperforms the original Ant Colony Algorithm.

Deng M. & Chen et al (2009) has worked on an automatic test data generation model that combines dataflow analysis with GA. It is a code-based testing technique, which supports class testing at three levels. The DTGA presented in this paper is guided by dataflow information to generate test data to cover all def-use pairs. Experimental results show this model is characteristic of sufficient code coverage and fast test data generation and it can be used in software testing for both desktop and embedded systems. The Model proposed has been validated on classes. But we still need to validate it on a number of other bigger examples. Our future work is to make extensions to the model prototype in order to meet the requirements of large-scale desktop/embedded software.

Geniana Ioana Laiu (2012) used the evolutionary algorithms to generate path test data .The evolutionary algorithms are namely Genetic Algorithm (GA), Simulated Annealing (SA) and Particle Swarm Optimization (PSO).The test data generated from those algorithms are compared to find out which algorithms produces better quality of test data. The result shows that Simulated Annealing produces better quality of test data than Particle Swarm Optimization and Genetic Algorithm.

Sharma C. & Sabharwal (2013) has worked on applications of GA in different types of software testing are discussed. The GA is also used with fuzzy as well as in the neural networks in different types of testing. It is found that by using GA, the results and the performance of testing can be improved.

Windisch A. & Joachim et al (2007) has reported on the application of particle swarm optimization to structural software testing. Both particle swarm optimization and genetic algorithms were used to automatically generate test cases for the same set of test objects.

Kaur A. & Bhatt (2011) presents a combined analytic view of evolutionary computation techniques namely Genetic Algorithm and Particle Swarm Optimization. The PSO is an optimization technique, where global solution is constructed by analysis of the local optimal solution.

Rini D. & Siti et al (2011) have made review of the different methods of PSO algorithm. The process of PSO algorithm in finding optimal values follows the work of an animal society which has no leader. Particle swarm consists of a swarm of particles, where the particle represents a potential solution. Particle will move through a multidimensional search space to find the best position.

Yang X. (2014) has worked on Optimization algorithms based on swarm intelligence can have some distinct advantages over traditional methods. It has been found that most SI-based algorithms use mutation and selection to achieve exploration and exploitation.

III. PROBLEM OF EXITING SYSTEM

The Existing system, test case Selection using Genetic Algorithm, Simulated Annealing Algorithms and Particle Swarm Optimization have been widely used but when it comes to Ant Colony Optimization it is not still deeply studied in generating test case and prioritization. Moreover in regression testing when the code changes we have to retest even those successfully executed test suites to check whether the changes have introduced new bugs or faults .As a result the problem here is that there is time consuming and extra efforts and resources needed.

IV. PROPOSED METHODOLOGY

Steps in Methodology -
The main objective of this research work is to formulate various techniques for testing the Test Case Selection. In the proposed work well known algorithm namely Particle Swarm Optimization & Ant Colony Optimization hybrid technique are implemented over the Test Case Selection.
The main reason behind using the hybrid technique is to find the optimal path & select the test case for all independent paths. In this input data source code or program is taken. In which data is filtered with the help of the hybrid techniques that is Particle Swarm Optimization & Ant colony Optimization for selecting test case. In Fig.4.1 shows proposed methodology

- **Step 1. Source Code** - Taking input program or source code
- **Step 2. Generate CFG** - Generate CFG for each program
- **Step 3. Calculate Cyclomatic Complexity** - Finding Cyclomatic complexity for each node which shows the number of path available from the current node to reach the end node of control flow graph
- By using the formula :
- V (G) = E – N + 2
- where V(G) = cyclomatic complexity for graph
    - E= Number of flow graph edges
    - N= Number of flow graph nodes

- **Step 4. Initialize Particle –** Initialize ‘n’ number of particles {p1----pn}.Here, a particle represents a test case. In this step initialization of Test Cases done.

- **Step 5. Calculate Fitnesses** - Position & Velocity of particle helps in observing fitness of that particle. The best particles are selected from population.

Update Velocity by PSO – $\mathbf{V_{id} = V_{id} + c_1 r_1 (P_{best} - X_{id}) + c_2 r_2 (G_{best} - x_{id})}$

Where, V_{id} : Velocity of agent i at iteration
c_j : Weighting factor(uniformly distributed random number between 0 and 1),
X_{id} : current position of agent i at iteration d,
P_{best} : P_{best} of agent i ,
G_{best} : G_{best} of the Group.

- Update Velocity by ACO –

$$\mathbf{X_{id} = X_{id} + V_{id}}$$

In whole, the PSO is an optimization approach that updates and validates each step. The process of updation comes from solution space only. A particle updates itself depending on the global best or local best. The random variable chosen is not generated but formulated by analyzing the population's position & velocity and particle's position & velocity.

- **Step 6. Update Particle** - Each particle updates itself with best particle in population. The Position & Velocity get updated and fitness of each particle is either improved or retained. Updated population is crossed over by dividing in two parts. Evaluate new off-springs and select the best off-spring as prioritized solution. The whole process of hybrid prioritized algorithm carried out until feasible solution is achieved. The population keeps on updating to reach the destination (final solution).

- **Step 7. Possible Best Paths** - Each 'n' particle compare Position with respect to local best & then with global best. Updation position velocity and combine with either local or global best .If updated posit ion& velocity > old posit ion & velocity Keep the new position & velocity revert back to old position and velocity.
- **Step 8. Generate test data by traversing each path** - In this step generating path details for each iteration. Eliminating the redundant test cases selecting and prioritization test cases. Selecting the best path with minimum execution time.

- **Step 9. Pick the best path given by HPSO** - This step shows the hybrid approach of PSO approach. The proposed approach will attempt to get a better result. Selecting test cases from the test suites will reduce the test efforts and incorporating it with prioritization using appropriate optimization algorithm will lead to better and effective fault revealing.

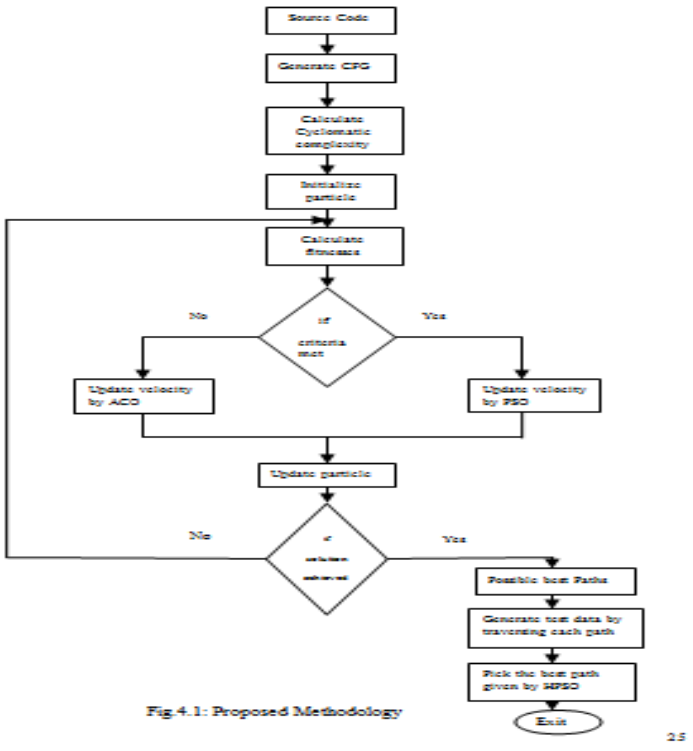

Fig 4.1: Proposed Methodology

Fig 1: Proposed Methodology

V. RESULT & DISCUSSIONS

Some of the screenshots for output screens. In order to evaluate the efficacy of the HPSO for test case selection within a time constrained environment. While searching for best solution in the search space, the particle can converge to one point between local and global best, hence, resulting in single point convergence, ignoring other aspects of search spaces. While searching for best solution in search space, particles communicate with each other and flow information about position and velocity.

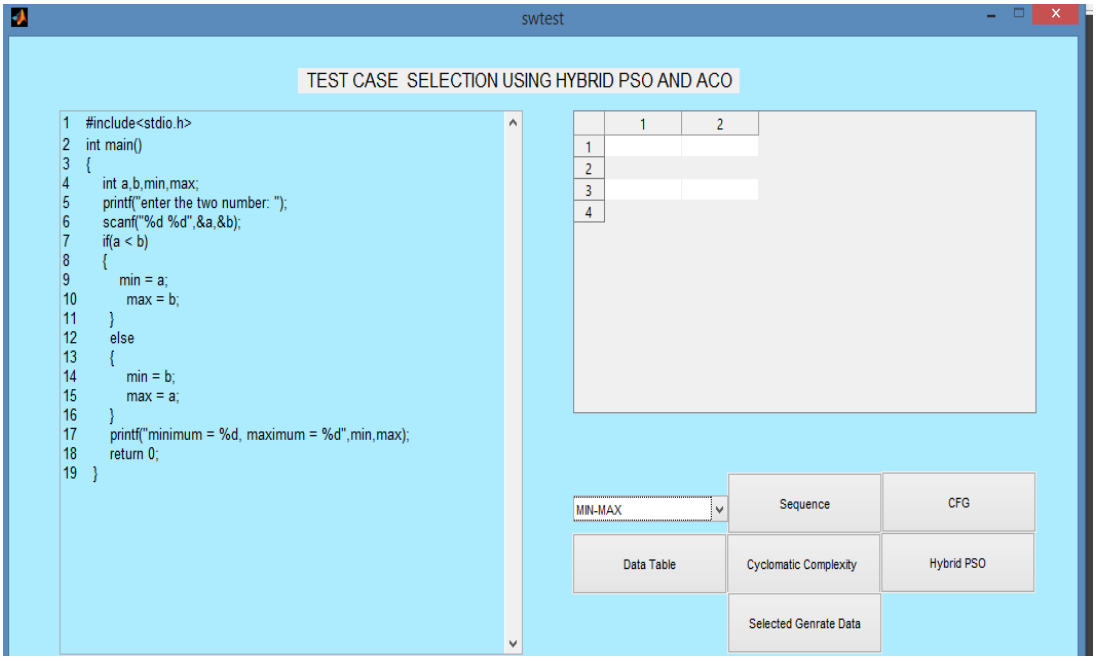

Fig 2: Sample Output Screen 1

For each run the best path and its execution time of all iterations are reported. Also the final weight on the edges and the path found in that run is shown. the optimal path was found by HPSO.

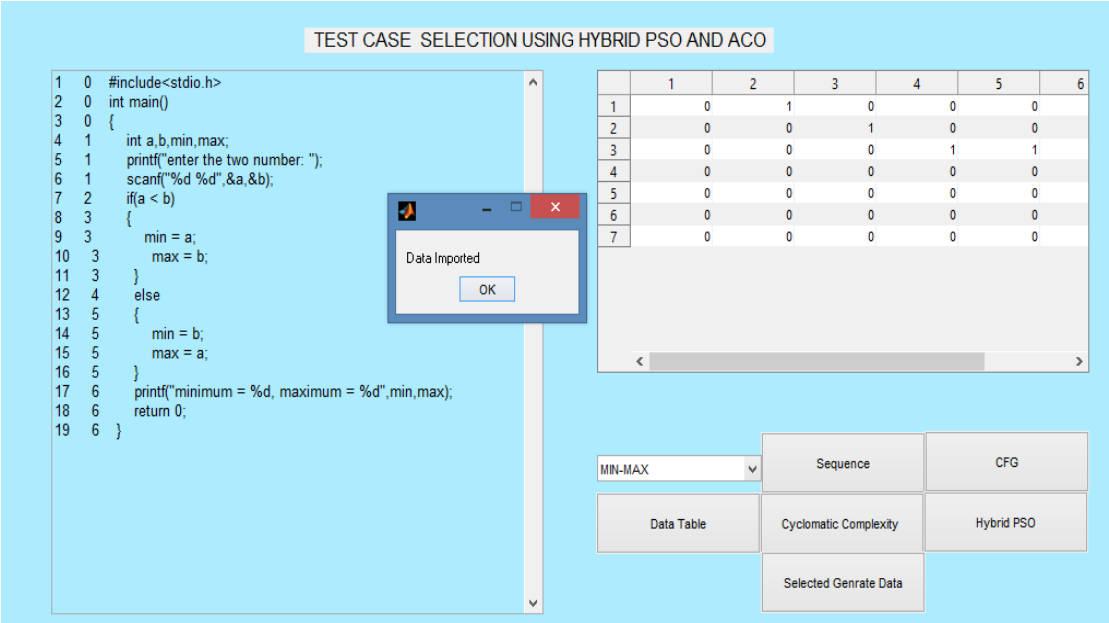

Fig 3: Sample Output Screen 2

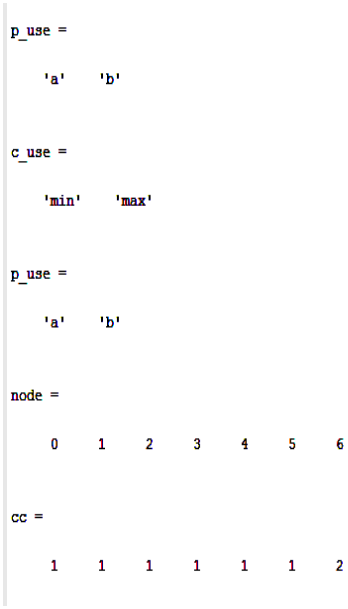

Fig 4: Sample Output Screen 3

The initiation of HPSO algorithm starts with generation of population. The HPSO is an artificial intelligence based technique, where we consider some population. The population plays major role in deciding which way Solution will approach. The population consists of particles or group of particles forming population. On selecting the population randomly from a given problem, we assign them position and velocities on the basis of original PSO technique.

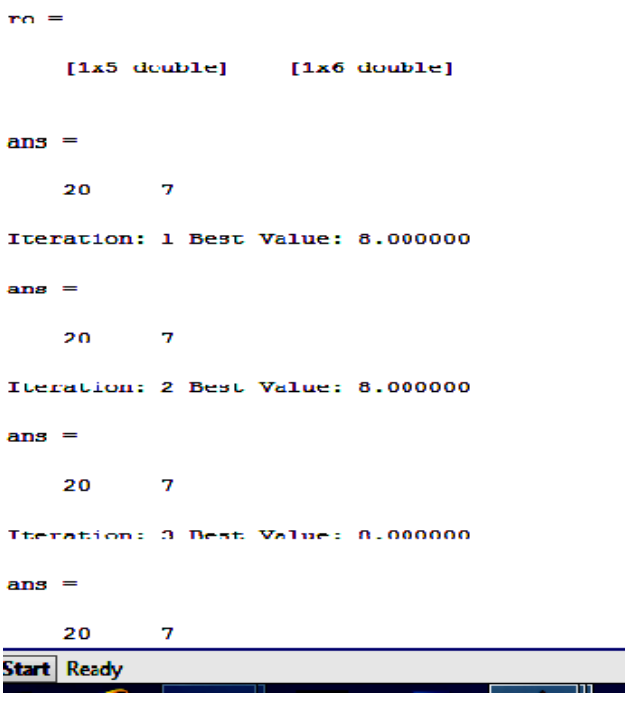

Fig 5: Sample Output Screen 4

VI. CONCLUSION & SCOPE OF FURTHER WORK

The main goal of research is to combine the power of two algorithms ACO and PSO. It proves its power and effectiveness towards solving the testing problems. Effectiveness and efficiency and is a much simpler algorithm The Observation shows that particle swarm Optimization is competitive with ACO and even outperforms them

for complex cases. Present work surveyed on various techniques of Software Test Case Selection and Prioritization in Ant Colony Optimization as well as brief comparisons of ACO algorithm with GA,SA and PSO algorithms.

Regression testing is very important for the software maintenance. To make it effective Test Case Selection technique is used. ACO is an algorithm used for implement test case Selection. ACO select the test cases based on the probability. We can summarize our proposed tool advantage in following few points:
In future more work can be done on following:

1) On other applications of Ant colony optimization like Travelling Salesman Problem.
2) Some other techniques of regression testing like test case prioritization can be modify to achieve better result

REFERENCES

[1] Chengying Mao, YuXinxin, Chen Jifu, Chen Jinfu (2012)"Generating Test Data for Structural Testing Based On Ant Colony Optimization "12th International Conference on Quality Software, Xi'an, Shaanxi, pp. 98 – 101.

[2] Gupta Nirmal Kumar and Rohil Mukesh Kumar (2013) "Improving GA based Automated Test Data Generation Technique for Object Oriented Software", IEEE International Advance Computing Conference (IACC), Ghaziabad, pp.249 - 253..

[3] Kaur Dr. Arvinder, Divya Bhatt (2011)" Hybrid Particle Swarm Optimization for Regression Testing" IJCSE, Vol. 3 No. 5.

[4] Yang Xin-She(2014)" Swarm Intelligence Based

[5] Algorithms: A Critical Analysis" Tandon Anisha, Poonam Malik (2012)"Breeding Software test cases with genetic alogorithm" IJAET

[6] Singh Naveen, Mrs. Kavita Agarwals (2013)" Software Testing using Evolutionary Approach IJSR.

[7] Gokalp Osman and Ugur Aybars (2012) "Improving Performance of ACO Algorithms Using Crossover Mechanism Based on Mean of Pheromone Tables". International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Trabzon, pp. 1 – 4.

[8] Latiu Geniana Ioana, Cret Octavian Augustin, Vcariu Lucia (2012) "Automatic Test Data Generation for Software Path Testing using Evolutionary Algorithms", International Conference on Emerging Intelligent Data and Web Technologies, Bucharest, pp.1-8.

[9] Windisch Andreas, Stefan Wappler et al (2007)" Applying Particle Swarm Optimization to Software Testing"ACM.

[10] Rini Dian palupi, siti mariyam shamsuddin et al, (2011)"Particle Swarm Optimization: Technique, System. And Challenges" IJCA,Volume 14– No

[11] Deng M, Chen R et al (2009), "Automatic Test Data Generation Model by Combining Dataflow Analysis with Genetic Algorithm",IEEE National Natural Science Foundation of China, pp. 429-433.

[12] Mouy P, Marre B et al ,(2008), "Generation of all- paths unit test with function calls",IEEE International Conference on Software Testing, Verification,and Validation,pp. 32-41.

[13] Tanwer S, Kumar D,(2010), "Automatic Test Case Generation of C Program Using CFG", International Journal of Computer Science Issues, Vol. 7, Issue 4, No 8.

[14] Lama S, Prasad M L, Srivastav P, (2011), "Automated Generation of Independent Paths and Test Suite Optimization using Artificial Bee Colony, Published by Elsevier Ltd. Selection.

[15] Ghiduk A. (2009), "Search –Based Testing Guidance Using Dominances vs. Control Dependencies", IEEE 16th Asia-Pacific Software Engineering Conference, pp. 145-151.