International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH14) 28 & 29 November 2014

A Comparative Analysis of Ant Colony Optimization for its Applications into Software Testing

Prashant Vats,
AIMACT, Banasthali University,
Rajasthan, India.
(prashantvats12345@gmail.com)

Manju Mandot,
J.R.N. Rajasthan Vidyapith,
Rajasthan, India.
(mandot_manju@yahoo.co.in)

Anjana Gosain,
USICT, GGSIPU,
New Delhi, India.
(anjana_gosain@hotmail.com)

Abstract: **Ant Colony Optimization are meta-heuristic algorithms that uses the search based algorithms as their base. It applies the natural phenomenon of finding the best possible path by the Ants that is covering the minimum distance from the food source to the ant colony, which will be followed by the rest of the ants, resulting into the optimized path. This phenomenon can be applied to provide optimized solutions to solve some complex computational problems. In this paper, we have carried out a review for the applications of the Ant colony Optimization algorithms in context to various level of the Software Testing, thus proving their worth in providing solutions to the various aspects of the Software Testing.**

Keywords: Optimized, Metaheuristic, Code coverage, Pheromone, Swarm Intelligence.

I. INTRODUCTION

Ant colony optimization (ACO) uses the probabilistic approach for providing the optimized solutions for some complex problems into the field of computational science by the use of graphs for finding the best paths and by using the code coverage criteria for a given problem. The ACO technique involves finding an optimal path into the graphs, which is based on the behaviors of ants for seeking a path between their colony and a source of food.

Into our Mother Nature, initially ants moves randomly into their surroundings and upon finding their food source and then they locate back to their colonies. While moving into this process the ants lay down their pheromone trails onto the paths covered while moving into the search of food sources. In order to avoid their continuous travelling in a random order, the other ants find such a path by following the trail, returning to their base colonies and reinforcing it if they find food eventually. As the time passes the trails of the pheromones laid down by the ants onto the path visited, starts fading away thus reducing its attractive strength. During frequent traveling onto the same path by numerous ants the left out pheromone further decays. While moving onto the path covered the shortest path is visited by rest of the ants at number of times results into laying of more pheromones onto it and thus resulting into strongly connected shortest path route for the coverage of path routing across the food source and the ant colony. The fading of the pheromone sprayed by the ants also provides the avoidance for the converging to a locally optimal solution. The failure of decaying of the pheromone onto the path covered by the previous ants will be followed by the rest of the ants and will result into the constraining of the exploration of the solution space for the computational problem. So, when one ant has find the best possible path that is covering the minimum distance from the food source to the ant colony, will be followed by the rest of the ants, thus will be resulting into the optimized path. So, the logic behind the Ant colony algorithm is to find the best possible optimized solution for the computational problems that are using the code coverage path by mimicking the behaviors of the simulated ants which moves onto the path around a graph depicting a computational problem.

On the other hand, Software testing is performed by covering each line of code (LOC) exactly once, which ensures that no path into the code has been left untested. It is also a matter of concern that complete testing or exhaustive testing of a software code is not possible. While performing structural testing of the Software under Test (SUT) it has to be ensured that each node of code has been traversed exactly once so that no LOC of the SUT has remained untested. Further also, to avoid redundant test cases, best possible set of test cases are chosen in such a form which not only provides the optimized set of test cases using which the testing of the entire code can be performed. The selection of best test cases among the available set of test cases that has been produced during a Software testing process is also known as the Test case prioritization technique, which consists of optimal set of test cases using which the whole given SUT can be test by avoiding the redundant test cases and also thus results into reduced space complexity and time complexity as well [29]. So, while performing testing using prioritized test cases, our motive should be the selection & prioritization of a set of good test cases that ensures the complete code coverage of a software program.

So, in this paper, we have specified the various levels of the software testing into which applying of the ACO algorithms has proven their worth for providing optimized solutions to various issues related to the software testing. In the second part of this paper, we have carried out a comparative review of the applications of ACO into software testing at various levels of the testing and the type of testing. In the third section of this paper, we have carried out a comparative survey of the work performed by the various researchers in the area of Software Testing with the application of the ACO algorithms. In the fourth section of this paper, we have carried out our conclusion based onto our review and the comparative study. In the last section of this paper, we have discussed the future work related to the Applications of the ACO algorithms into the area of Software Testing.

© 2014 CIPECH14.

II. LITERATURE REVIEW

Authors K. Ayari, S. Bouktif [1] has proposed their work on automatic test input data generation for reducing the cost of coverage based test strategy during testing of SUT. It is includes Structural Testing which is performed at integrated system level testing. The proposed work incorporates the Fault-based testing which reduces the computational cost for automatic test input data generation in the context of mutation testing. Further the proposed work has addressed the customization of ACO to the problem of generating input test data to kill mutants' test cases. Its disadvantage is that the adequacy function involved during the generation of the test case does not have the provision of evaluating the costs of necessary sufficient conditions to kill a mutant in a test program. The proposed work has been implemented in Java.

Mattia V., Andre M. [2] has propose their work on a search based technique for the automatic generation of unit test cases based on a data-flow criteria & further to automate their work in EVOSUITE test generation tool which is based on ACO. It involves structural testing of SUT at Unit level by using Coverage Criteria based testing. Its advantage is that the use of EVOSUITE has produced the test cases which overcome the difficulty of manually covering def-use pairs for the SUT. At one end the proposed work represents a viable alternative for the simpler structural criteria like branch coverage but on the other part, it doesn't considers the aliasing or contextual def-use pairs so that the overall number of def-use pairs in the current analysis can be smaller than the existing test pairs. It has been implemented in Java.

Another author Ahmed S. [3] has presented their work on the generation of a set of optimal paths for covering all definition-use associations (du-pairs) in the SUT which is based on ACO. It involves structural testing of the SUT at the integrated system level during testing. It includes Control-Flow based Testing. Its advantage is that the proposed ACO based testing approach generates suite of test-data for satisfying the generated set of paths. Its disadvantage is that it doesn't address the construction of searching model for a program with input variable of Boolean & character type. It is implemented in C++.

Raka Z., Milan T. [4] has proposed an Object-Oriented framework for developing an Ant colony systems called GRAF-ANT which has been used for the simplicity of creating multi thread applications. It involves Structural Testing of the SUT at the system level by using Flow Path based Testing. Its advantage is that in this proposed framework Multi colony systems are connected with network calculations for the emulation of several standard network communication methods for testing purposes. Also further during testing of the SUT a powerful GUI in combination with a multi colony system allows for a large number of simultaneous colonies to run by simplifying the process of the retrieval of useful test results. Its disadvantage is that the GUI framework GRAF-ANT developed needs full application code to be compiled when it applied for testing purpose. It is implemented in C#.

Jose C., Francisco F. [5] has presented their work for proposing a methodology that works for the generation & optimization of the test data by employing the evolutionary search based techniques as a basis on which the information will be provided by analysis, interpretation & dynamic execution of an instrumented test object that is implemented in a Java byte code. It involves Structural Testing of the SUT at system level. It includes evolutionary testing of the system. Its advantage is that it focuses on the creation and optimization of a test set that is not only maximizes the code coverage but it also provides the optimization to occur at the test set level and at the test case level. Its disadvantage is that it has not been tested for Real time environment. It is implemented in Java.

Huaizhong L., C. Peng Lam [6] has proposed their work of n ACO based approach to test data generation for the state-based software testing. It is based on Structural Testing and involves integrated system level of testing. It uses State based testing. Its advantage is that it applies a transition rule for determining the probability of an ant traversing from one node in the graph to the next.

Shankarlingayya K., Kiran K. M. [7] has proposed a technique for generating a high priority test sequences for a given state machine by making use of a combination of "tree" from the state based system & applying ACO on it. It uses structural testing for integrated level system. It involves State based testing. Its advantage is that it generates high priority test sequences from the state based system with effectively controlled repetition of the state. It provides faster test sequence generation without any redundant or duplicate test sequences.

Madhumita P., Parth P. S. [8] has propose an optimum solution for path coverage based testing, by generating test data automatically using Differential Evolution, Genetic Algorithm and Artificial BCO algorithm. It involves Structural Testing to be performed at the system level. It involves path coverage based testing. Its advantage is that the proposed approach is able to cover conditions which contain Boolean value and multiple paths by using the weighted approach i.e. assigning weights to each path covered. It was implemented in MATLAB version R2008a.

Huaizhong L., Chiou P. L. [9] has proposed an Ant Colony Optimization approach for the automatic test sequence generation for state-based software testing by using use UML artifacts to automatically generate test sequences to achieve the required test coverage. It involves Structural Testing which is pursued at integrated system Level of the software under test. It includes State based Testing. Its advantage is that by using UML State chart diagrams, they can be directly used to generate test sequences. The whole generation process is also fully automated. In this proposed work the redundant exploration of the State chart diagrams is avoided due to the use of ants which results in the efficient generation of test sequences. But, its disadvantage is that using more ants cannot improve the results further for a test suite which contains the shortest test sequences. It is implemented by using a Dynamic Ant Simulator (DAS).

Saurabh S., Sarvesh K. [10] has proposed their work for generating optimal paths and cyclomatric complexity for finding the number of feasible paths using control flow graph (CFG). It is based on structural testing. It is performed at integrated system Level testing. It involves path coverage based testing. Its advantage is that the proposed algorithm improves path testing by using ant colony Optimization algorithm. Further, in the proposed work the best path with the highest priority is selected first and then in continuous manner all the paths in the control flow graph can be tested.

Praveen R., Baby [11] has proposed an ACO technique for the automated and fully coverage state-transitions in the system by easily traversing all transitions at least once in the test-sequence. It performs structural testing which includes State transition testing. Its main advantage is that the State-transition based testing provides all state coverage. It also provides all event coverage and provides all transition coverage. In the proposed work UML state-transition diagram created, which shows the static behavior of software.

Zilong X., Fan M. [12] has proposed an ACO for the minimal test cost attribute reduction which is an important issue in cost-

sensitive learning. It involves structural testing which is performed at integrated system level. It involves Flow graph based testing. Its advantage is that the proposed ant colony optimization algorithm is significantly better to tackle the minimal test cost reduction problem. Its disadvantage is that in the proposed work emphasize should be given on improving the efficiency of the algorithm and improve it for large datasets. It was implemented using COSER and Java.

Rana F., Haidar M. H. [13] has proposed an automatic test pattern generation for combinational circuits using ACO. It involves Object oriented testing to be performed at integrated system level testing. It involves Fault based testing. Its main advantage is that it efficiently generates a set of test vectors that achieve high fault coverage in a short time. Its disadvantage is that the number of test patterns is small. It is implemented in Java.

Kewen L., Zilu Z. [14] has proposed a model for test data generation based on ACO and path coverage criteria for improving the efficiency of test data generation. It involves Structural testing which is performed the system level for the SUT. It includes path coverage based testing. Its advantage is that the proposed algorithm has a better performance and it improves the efficiency of test data generation. Its disadvantage is that it cannot be applied to object-oriented programs. It was implemented using C# .Net.

Xiang C., Qing G. [15] has proposed an algorithm based on ant colony optimization (ACO) for building prioritized pair wise interaction test suite (PITS) for providing a systematic approach for software testing. It involves Object oriented Testing which is performed at the system level of the SUT. It includes Prioritized interaction testing. Its advantage is that it provides a systematic approach for software testing. Its disadvantage is that in this proposed work its results cannot be generalized for all the possible inputs. It is implemented in java using the JDK 1.6 version.

Praveen R. S., K. Baby [16] has proposed an approach for the optimal path identification by using the basic property and behavior of the ant by using certain set of rules to find out all the effective optimal paths via using the ACO. It involves structural testing of the SUT which is performed at the system level. It includes Path coverage Testing. Its main advantage is that the proposed algorithm concentrates on the generation of paths, equal to cyclomatric complexity. It guarantees full path coverage by automatically selecting a best path sequence which covers maximum coverage at least once.

Cui D., Yin W. [17] has proposed an ACO for reducing the size & cost of the test-suite for achieving higher effectiveness of the test-suite minimization. It includes structural testing using Code coverage which is performed at the system level. Its advantage is that in it, the use of ACO considers the test coverage and test run costs for providing an optimal result which consist the subset with a minimal cost & improves the test efficiency.

Toufik B., Walid Z. [18] has focused their work on proposing an approach for describing the targeted test criteria using structural information of the system under test by using ACO for the automatic generation of the adequate test data. It involves structural testing to be performed at the integrated system level of testing for the SUT. It includes Search based code coverage testing. Its advantage is that the applicability of Search based software Testing performs better than random test data generation. It is implemented in Java.

Chengying M., Xinxin Y. [19] has proposed a framework for the ACO-based test data generation by ensuring the quality of test data set using branch coverage as a coverage criterion. It involves Structural Testing. It has been performed at the integrated system level of Testing. It includes Search-based software testing. Its advantage is that the proposed outperforms the traditional test data generation methods in several aspects such as cost, efficiency and fault-revealing ability. Its limitation is that the proposed algorithm may have been focused on reducing the time of global or local search and pheromone update. It has been implemented in C++ and MS Visual Studio .Net.

Kewen L., Zhixia Y. [20] has proposed a method based on ACO for the generation of pair-wise covering test data for the Optimized Test Sequence Generation. It involves Structural Testing performed at the integrated system level. It involves Branch coverage based Testing. Its advantage is that it generates fewer test cases that cover a more pair combination. It has been implemented in Java Applet.

D.J. Mala, M. Kamalapriya [21] have proposed and applied a non-pheromone based intelligent swarm optimization technique namely artificial bee colony optimization (ABC) for test suite optimization. It involves structural testing of the SUT which is performed at system level of testing. It has used path coverage based testing. Its advantage is that using the test adequacy criterion, the quality of the test cases is improved during each iteration to cover the paths in the software.

X. Chen, Q.Gu. [22] has proposed a novel approach for the generation of variable strength interaction test suites (VSITs) using ACO. It has used structural testing. It is performed at the integrated system level. It includes path coverage based testing. Its advantage is that it adopts a one-test-at-a-time strategy to build final test suites. It further generates a single test by adopting ant colony system (ACS) strategy by formulizing solution space, cost function and several heuristic settings in this ACS framework.

Praveen R. S., Km. Baby [23] has presented an algorithm by using an ant colony optimization technique, for the generating a series of optimal and minimal test sequences which is used for the behavior specification of any SUT. It involves Object Oriented Testing which has been performed at the integrated system level. It includes the State Transition Testing. Its advantage is that it generates a test sequence which is used in order to obtain the complete software coverage for an SUT. It also limits the repeated transitions that occurs in the test sequence and also provides the full code coverage for an SUT. It has been implemented using STTACO Tool and UML.

Praveen R. S., Vijay K. R. [24] has proposed an ACO approach for automatic test sequence generation for control flow based software testing by using use control flow graph. It involves Object Oriented Testing. It is performed at the integrated system level of testing. It involves Fault based coverage testing. Its advantage is that the test sequences are automatically generated for achieving the required test coverage by directly using the control flow graph.

Praveen R. S., Nitin J. [25] has proposed a technique for the generation of optimized test sequences from a markov chain based usage model using ACO by emphasizing on the cost and criticality of various states in the proposed model. It includes the Object Oriented Testing that has been performed at the system level of the SUT. It involves Model based testing. Its advantage is that the proposed algorithm is very effective in generating optimal set of test cases from a markov chain based usage model. The proposed algorithm is capable of deriving test cases with priority given to the most critical states and transitions. Its disadvantage is that it does not involve the presence of multiple transitional probabilities between the various states of the SUT.

Praveen R. S. [26] has proposed an efficient algorithm for the automatic generation of all possible paths in a Control Flow

Graph for the structural testing of a SUT, based on Pheromone releasing behavior of ants. It includes structural testing. It is performed at Integrated system level Testing. It involves the Control-Flow Testing of the SUT. Its advantage is that it generates paths equal to the cyclomatric complexity.

Praveen R. S., V. Ramachandran [27] has further proposed his work for generating the optimum set of test data using genetic algorithm and ant colony optimization for a given software while not compromise on exhaustive testing of the SUT. It involves structural testing which is performed at the integrated system level of Testing. It includes the Control flow graph based Testing. Its advantage is that the proposed algorithm minimizes the cost of testing by using the Control flow graph & path coverage based approach.

Yogesh S., Arvinder K. [28] has proposed their work onto the regression test case prioritization technique for the reordering of the test suites in a time constraint environment. It involves the Regression Testing which has been performed at the integrated system level. It includes the Fault coverage based testing. Its main advantage is that the proposed regression testing algorithm has prioritized the tests in order to cover maximum faults in minimum time.

III. A TABULAR COMPARISON OF THE ACO ONTO THE VARIOUS LEVELS OF SOFTWARE TESTING

Into this section of the paper we have carried out a tabular analysis of ACO algorithms for its application into Software Testing. We have represented the observations made during the analysis are given into Table. 1.

IV. CONCLUSION

In this paper we have reviewed the Ant Colony Optimization algorithms in context to their applications in providing effective and optimized results for the various aspects of Software Testing. A tabular comparison has been carried out after conducting a survey of the ACO algorithms. After review of which we conclude that by applying the ACO algorithms into the area of Software Testing has provided effective optimized results to the complex problems related to the code coverage of the SUT.

V. FUTURE WORK

Based on the above we can direct our research towards improving the results further for a test suite which contains the shortest test sequences. Further we can deviate our research towards automating the frameworks that are implementing these algorithms. We can also focus on reducing the time of global or local search and the pheromone update that is performed while traversing through the connected graphs.

REFERENCES:

[1] Kamel A., Salah B., “Automatic mutation test input data generation via ant colony”, pub. in the Proceedings of the 9th annual conference on Genetic and evolutionary computation, pg. no.- 1074-1081, 2007.

[2] Mattia V., Andre M.,"Search-based data-flow test generation", pub. in proceedings of IEEE 24th International Symposium on Software Reliability Engineering (ISSRE), pg. no. 370-379, 2013.

[3] Ahmed S.,"A New Software Data-Flow Testing Approach via Ant Colony Algorithms", pub. in Universal Journal of Computer Science and Engineering Technology, Vol. 1, Iss. 1, pg. no. 64-72, 2010.

[4] Raka Z., Milan T., "An Object-Oriented Framework with Corresponding Graphical User Interface for Developing Ant Colony Optimization Based Algorithms", pub. in proceedings of WSEAS transaction on computers, Iss. 12, Vol. 7, pg. no.: 1948-1957, 2008.

[5] Jose C., Francisco F.,"Using Dynamic Analysis of Java Byte code for Evolutionary Object-Oriented Unit Testing", pub. in proceedings of 25th Brazilian Symposium on Computer Networks and Distributed Systems (SBRC), pg. no. 143-156, 2007.

[6] Huaizhong L., C. Peng Lam,"Software Test Data Generation using Ant Colony Optimization", pub. in International Journal of Computer, Information, Systems and Control Engineering Vol: 1, No: 1, pg. no. 126-129, 2007.

[7] Shankarlingayya K., Kiran K. M.," High Priority Test Sequence Generation for the State Based System", pub. in the conference proceeding of ISQT, STEP-AUTO, pg. no. 24-28, 2011.

[8] Madhumita P., Parth P. S., "Performance Analysis Of Test Data Generation For Path Coverage Based Testing using three metaheuristic algorithms", pub. in International Journal of Computer Science and Informatics, Vol. 3, Iss. 2, pg. no. 34-41, 2013.

[9] Huaizhong L., Chiou P. L., "An Ant Colony Optimization Approach to Test Sequence Generation for State-Based Software Testing", pub. in Proceedings of the Fifth IEEE International Conference on Quality Software (QSIC’05), 2005.

[10] Saurabh S., Sarvesh K., "Optimal path sequencing in basic path testing", pub. in International Journal of Advanced Computational Engineering and Networking, Volume – 1, Issue – 1, pg. no. 55-59, Mar-2013.

[11] Praveen R., Baby," Automatic Test Sequence Generation for State Transition Testing via Ant Colony Optimization", pub. in book Evolutionary Computation and Optimization Algorithms in Software Engineering: Applications and Techniques, Edition: IST, Pub. by IGIA GLOBAL, USA, 2005.

[12] Zilong X., Fan M., "Ant colony optimization to minimal test cost reduction", pub. in proceedings of IEEE International Conference on Granular Computing, 2012.

[13] Rana F., Haidar M. H.," An ant colony optimization approach for Test pattern generation", pub. in proceedings of IEEE Canadian Conference on Electrical and Computer Engineering, pg. no. 001397 - 001402, 2008.

[14] Kewen L., Zilu Z.,"Automatic Test Data Generation Based On Ant Colony Optimization", pub. in proceedings of Fifth IEEE International Conference on Natural Computation, pg. no. 216-220, 2009.

[15] Xiang C., Qing G.,"Building Prioritized Pairwise Interaction Test Suites with Ant Colony Optimization", pub. in proceedings of Ninth IEEE International Conference on Quality Software, pg. no. 347-352, 2009.

[16] Praveen R. S., K. Baby,"An Approach of Optimal Path Generation using Ant Colony Optimization", pub. in proceedings of IEEE Region 10 Conference TENCON, pg. no. 1-6, 2009.

[17] Cui D., Yin W.,"The Research of Test-Suite Reduction Technique", pub. in proceedings of IEEE International Conference on Consumer Electronics, Communications and Networks (CECNet), pg. no. 4552 - 4554, 2011.

[18] Toufik B., Walid Z.,"Targeted adequacy criteria for search-based test data generation", pub. in proceedings of IEEE International Conference on Information Technology and e-Services (ICITeS),pg. no. 1 - 6, 2012.

[19] Chengying M., Xinxin Y.,"Generating Test Data for Structural Testing Based on Ant Colony Optimization", pub. in proceedings of 12th IEEE International Conference on Quality Software, pg. no. 98-101, 2012.

[20] Kewen L., Zhixia Y.,"Extracting Test Sequences from a Markov Software Usage Model by ACO", pub. in proceeding of Springer Genetic and Evolutionary Computation — GECCO, pg no. Vol. 2724, pg. no. 2465-2476, 2003.

[21] D.J. Mala, M. Kamalapriya,"IntelligenTester –Test Sequence Optimization framework using Multi-Agents", pub. in the Journal Of Computers, VOL. 3, NO. 6, pg. no. 39-46, June- 2008.

[22] X. Chen, Q. Gu,"Variable Strength Interaction Testing with an Ant Colony System Approach,” pub. in the Proceedings of Software Engineering Conference, APSEC '09, Penang, Asia-Pacific,pg. no.160-167, Dec 2009.

[23] Praveen R. S., Km. Baby, “Automated Software Testing Using Metahurestic Technique Based on An Ant Colony Optimization", pub. in proceedings of International Symposium on Electronic System Design (ISED), pg. no. 235-240, 2010.

[24] Praveen R. S., Vijay K. R., "An ant colony optimization approach to test sequence generation for control flow based software testing,” pub. in the proceedings of 3rd International Conference on Information Systems, Technology and Management (ICISTM‘09), pg. no. 345-346, 2009.

[25] Praveen R. S., Nitin J.," Optimized Test Sequence Generation from Usage Models using Ant Colony Optimization,” pub. in International Journal of Software Engineering & Applications (IJSEA), Vol. 2, No. 2, pg. no. 14-28, 2010.

[26] Praveen R. S., "Structured Testing Using Ant Colony Optimization", pub. In proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia (IITM’10), pg. no. 203-207, Dec 2010.

[27] Praveen R. S., V. Ramachandran, "Generation of test data using Meta heuristic approach," pub. in the Proceedings of IEEE TENCON, Nov 2008.

[28] Yogesh S., Arvinder K.,"Test Case Prioritization Using Ant Colony optimization”, pub. in ACM Newsletter SIGSOFT(ACM) Software Engineering Notes, New York, USA, pg. no. 1-7, 2010.

[29] Bharati S., Shweta S.,"Literature Survey of Ant Colony Optimization in Software Testing", pub. in proceedings of Sixth IEEE CSI International Conference on Software Engineering (CONSEG), pg. no. 1-7, 2012.

TABLE 1. A COMPARATIVE REVIEW OF THE ACO ONTO THE VARIOUS LEVELS OF SOFTWARE TESTING

Sr. No.	Authors	Issues Addressed	Type of Testing	Level of Testing	Testing Approach	Merits	Demerits	Tools used
1.	K. Ayari, S. Bouktif [1] (2007)	For automatic test input data generation to reduce the cost of coverage based test strategy	Structural Testing	Integrated System Level Testing	Fault-based testing	Reduces the computational cost for automatic test input data generation in the context of mutation testing + customization of ACO to the problem of generating input test data to kill mutants	Test case adequacy function is not having the provision to evaluate the costs of necessary sufficient conditions to kill a mutant in a test program.	Java.
2.	Mattia V., Andre M. [2] (2013)	To propose a search based technique to automatically generate unit tests for data-flow criteria & to automate it in EVOSUITE test generation tool based on ACO.	Structural Testing	Unit Level Testing.	Coverage Criteria based testing	Use EVOSUITE to produce the tests overcomes the difficulty of manually covering def-use pairs+ represents a viable alternative to simpler structural criteria like branch coverage.	Doesn't considers the aliasing or contextual def-use pairs so that the overall number of def-use pairs in the current analysis is smaller than the existing test pairs.	Java
3.	Ahmed S. [3] 2010.	For generation of a set of optimal paths to cover all definition-use associations (du-pairs) in the program under test based on ACO.	Structural Testing	Integrated system Level testing.	Control-Flow based Testing	Proposed ACO based testing approach generate suite of test-data for satisfying the generated set of paths.	Doesn't address the construction of searching model for a program with input variable of Boolean & character type.	C++
4.	Raka Z., Milan T. [4] 2008.	To propose an Object-Oriented framework for developing Ant colony systems called GRAF-ANT for the simplicity of creating multi thread applications	Structural Testing	System level Testing	Flow Path based Testing	In the proposed frame-work Multi colony systems are connected with network calculations for emulating several standard network communication methods for testing purposes. + During testing a power-ful GUI combined with a multi colony system allows a large number of simultaneous colonies to run, simplified the process of retrieving results	GRAF-ANT needs full application code to be compiled when it applied for testing purpose.	C#
5.	Jose C., Francisco F. [5] 2007.	For proposing a methodology for generating & optimizing test data by employing evolutionary search techniques basis on the information provided by analysis, interpretation & dynamic execution of a instrumented test object in Java byte code.	Structural Testing	System level Testing	Evolutionary Testing	Focus on the creation and optimization of a test set that maximizes code coverage + Optimization occurs at the test set level and at the test case level:	Has not been tested for Real time environment.	Java
6.	Huaizhong L., C. Peng Lam [6] 2007.	Proposal of n ACO based approach to test data generation for the state-based software testing	Structural Testing	Integrated system level testing	State based testing	It applies a transition rule For determining the probability of an ant traversing from one node in the graph to the next	NA	NA
7.	Shankarlingayya K., Kiran K. M. [7] 2011.	To propose a technique for generating a high priority test sequences for a given state Machine by making use of a combination of "tree" from the state based system & applying ACO on it.	Structural Testing	Integrated system Level testing.	State based testing	Generate high priority test sequences from the state based system with effectively controlled repetition of the state +faster test sequence generation without any redundant or duplicate test sequences	NA	NA
8.	Madhumita P., Parth P. S. [8] 2013.	To propose a n optimum solution for path coverage based testing, by generating test data automatically using Differential Evolution, Genetic Algorithm and Artificial BCO algorithm	Structural Testing	System Level testing.	Path coverage based testing	The proposed approach is able to cover conditions contains Boolean value and multiple paths by using the weighted approach i.e. Assigning weights to each path covered.	NA	MATLAB ver. R2008a
9.	Huaizhong L., Chiou P. L. [9] 2005.	To proposes an Ant Colony Optimization approach to automatic test sequence generation for state-based software testing by using use UML artifacts to automatically generate test sequences to achieve required test coverage.	Structural Testing	Integrated system Level testing.	State based Testing	UML Statechart diagrams e are directly used to generate test sequences + whole generation process is fully automated + red-undant exploration of the Statechart diagrams is avoided due to the use of ants, resulting in efficient generation of test sequences.	Using more ants cannot improve the results further for a test suite which contains the shortest test sequences	Dynamic Ant Simulator (DAS)
10.	Saurabh S., Sarvesh K. [10] 2013.	To generate optimal paths and cyclomatric complexity for finding the number of feasible paths using control flow graph (CFG)	Structural testing	Integrated system Level testing.	Path coverage based testing	Proposed algorithm improves path testing by using ant colony Optimization algorithm+ The best past with highest priority is sele-cted first and then in Continuous manner all the paths in the control flow graph can be tested.	NA	NA
11.	Praveen R., Baby [11] 2005.	To propose as ACO technique for automated and fully coverage state-transitions in the system by easily traversing all transitions at least once in the test-sequence.	Structural testing	State transition testing	State transition testing	State-transition based testing provides all state coverage + It provides all event coverage + provides all transition coverage + UML state-transition diagram created, which shows the static behavior of software.	NA	NA
12.	Zilong X., Fan M. [12] 2012.	To propose an ACO for minimal test cost attribute reduction is an important issue in cost-sensitive learning	Structural testing	Integrated system level testing	Flow graph based testing	The proposed ant colony optimization algorithm is significantly better to tackle the minimal test cost reduction problem.	Emphasize should be given on improving efficiency of the algorithm and improve it for large datasets.	COSER+ Java
13.	Rana F., Haidar M. H.[13] 2008.	To propose an automatic test pattern generation for combinational circuits using ACO.	Object oriented Testing	Integrated system level testing	Fault based testing	It efficiently generates a set of test vectors that achieve high fault coverage in a short time.	The number of test patterns are small.	Java,

14.	Kewen L., Zilu Z. [14] 2009	To propose a model for test data generation based on ACO and path coverage criteria for improving the efficiency of test data generation	Structural testing	System level testing	Path coverage based testing	The algorithm has a better performance and improve the efficiency of test data generation	It cannot be applied to object-oriented programs	C# .Net
15.	Xiang C., Qing G [15] 2009.	To propose an ant colony optimization (ACO) to build prioritized pair wise interaction test suite (PITS) for providing a systematic approach for software testing	Object oriented Testing	System level testing	Prioritized interaction testing	It provides a systematic approach for software testing.	In it results cannot be generalize for all possible inputs.	JDK 1.6
16.	Praveen R. S., K. Baby [16] 2009	To propose an approach for the optimal path identification by using the basic property and behavior of the ant by using certain set of rules to find out all effective optimal paths via ant colony optimization	Structural testing	System level testing	Path coverage Testing	Concentrates on generation of paths, equal to cyclomatric complexity + guarantees full path coverage + automatically selects a best path sequence which covers maximum coverage at least once	NA	NA
17.	Cui D., Yin W. [17] 2011.	To reduce the size & cost of the test-suite for achieving higher effectiveness of test-suite minimization	Structural testing	System level testing	Code coverage testing	Use of ACO considers the test coverage and test run costs for providing an optimal result consists the subset with minimal cost & improves Test efficiency	NA	NA
18.	Toufik B., Walid Z. [18] 2012.	To propose an approach for describing targeted test criteria using structural information of the system under test by using ACO for the automatically generating the adequate test data	Structural testing	Integrated system level Testing	Search based code coverage testing	Applicability of Search based software Testing performs better than random test data generation	NA	Java
19.	Chengying M., Xinxin Y. [19] 2012	To propose a framework of ACO-based test data generation while ensuring the quality of test data set using branch coverage as a coverage criterion.	Structural Testing	Integrated system level Testing	Search-based software testing	Proposed outperforms the traditional test data generation methods in several aspects such as cost, efficiency and fault-revealing ability	Proposed algorithm can be focused on reducing the time of global/local search and pheromone update.	C++ + MS Visual Studio .Net
20.	Kewen L., Zhixia Y. [20] 2003.	To propose a method based on ACO for the generation of pair-wise covering test data for Optimized Test Sequence Generation	Structural Testing	Integrated system level Testing	Branch coverage based Testing	Generates fewer test cases that cover more pair combination	NA	Java Applet.
21.	D.J.Mala , M. Kamalapriya [21] 2008.	To propose and apply a non-pheromone based intelligent swarm optimization technique namely artificial bee colony optimization (ABC) for test suite optimization	Structural testing	System level testing	Path coverage based testing	Using the test adequacy criterion quality of the test cases is improved during each iteration to cover the paths in the software.	NA	NA
22.	X. Chen, Q.Gu. [22] 2009.	Proposing a novel approach for the generation of variable strength interaction test suites (VSITs) using ACO.	Structural testing	Integrated system level Testing	Path coverage based testing	It adopt a one-test-at-a-time strategy to build final test suites + generate a single test by adopting ant colony system (ACS) strategy by formulizing solution space, cost function and several heuristic settings in this ACS framework	NA	NA
23.	Praveen R. S., Km. Baby [23] 2010.	To present an algorithm by applying an ant colony optimization technique, for generation of optimal and minimal test sequences for behavior specification of software	Object Oriented Testing	Integrated system level Testing	State Transition Testing	Generates test sequence in order to obtain the complete software coverage + limits the repeated number of transitions in the test sequence and also pro-vides the full coverage	NA	STTACO Tool +UML
24.	Praveen R. S., Vijay K. R. [24] 2009.	To propose ACO approach for automatic test sequence generation for control flow based software testing by using use control flow graph	Object Oriented Testing	Integrated system level Testing	Fault based coverage testing	Automatically generate test sequences for achieving required test coverage by directly using the control flow graph	NA	NA
25.	Praveen R. S., Nitin J. [25] 2010.	To propose a technique for the generation of optimized test sequences from a markov chain based usage model using ACO by emphasizing on the cost and criticality of various states in the proposed model.	Object Oriented Testing	System level testing	Model based testing	Effective in generating optimal set of test cases from a markov chain based usage model + capable of deriving test cases with priority given to the most critical states and transitions	Does not involve the presence of multiple transitional probabilities between states.	NA
26.	Praveen R. S. [26] 2010.	To propose an efficient algorithm for the automatic generation of all possible paths in a Control Flow Graph for the structural testing of a SUT based on Pheromone releasing behavior of ants.	Structural testing	Integrated system level Testing	Control-Flow Testing	Generates paths equal to the cyclomatric complexity	NA	NA
27.	Praveen R. S., V. Ramachandran [27]2008.	To generate the optimum set of test data using genetic algorithm and ant colony optimization for a given software while not compromise on exhaustive testing of the SUT.	Structural testing	Integrated system level Testing	Control flow graph based Testing	Minimize the cost of testing by using the Control flow graph & path coverage based approach.	NA	NA
28.	Yogesh S., Arvinder K. [28] 2010.	To propose a regression test case prioritization technique for the reordering of the test suites in a time constraint environment	Regrassion Testing	Integrated system level Testing	Fault coverage based testing	Prioritize the tests in order to cover maximum faults in minimum time	NA	NA