

## International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IMPACT FACTOR: 5.258

*IJCSMC, Vol. 5, Issue. 5, May 2016, pg.278 – 280*

# Various Artificial Intelligence Approaches in Field of Software Testing

**Neha Bhateja**

Faculty, Department of Computer Science & Engineering, Amity University Gurgaon, Haryana, India  
[bhateja.neha@gmail.com](mailto:bhateja.neha@gmail.com)

---

*Abstract: Before releasing the software to the client end, the developers must know whether the designed software is working acceptably or not. In other terms, the software must be tested to check that all the requirements are integrated well. Software testing is one of the primary techniques to gain consumers confidence in the software. The development of techniques will also support the automation of software testing. Many artificial intelligence approaches are also used to optimize the testing resources. These approaches are used in various processes of the software testing, quality and reliability in terms of generation of test sequence, automation of testing etc. This paper presents the number of artificial intelligence approaches that are used by the researchers.*

*Keywords: Software Testing, Test Cases, Test data generation*

---

## I. INTRODUCTION

Software testing is a process to observe the behavior of software and recording results under given conditions. Software testing gives an observation to understand the risks involved in software implementation. Software Testing is a process of executing a program with intends to find software bugs so that they can be rectify.

Testing process only establish a confidence that product would operate well in the specific situation rather than ensure that software would work properly in all conditions. Software testing process includes generation of test data, test execution which include test data and software to test and evaluation of test results activities [1]. The test data generation involves activities for producing a set of test data that satisfied a criterion. A properly generated test suite not only locates the errors in software, but also helps in reducing the high cost associated with software testing. A test case is a data set, which includes input data, execution paths, conditions for execution, and requirements of testing [2]. Test cases are selected in such a way that all positive scenarios and negative scenarios of software are covered.

The test data generated automatically to achieve required test coverage. For this, many researchers are using the concept of artificial Intelligence in software testing field to achieve the better accuracy of automation testing [3]. The Artificial Intelligence approaches that are used in software testing are Ant Colony Optimization (ACO), Genetic Algorithm (GA), Tabu search (TB), Bee Colony, Data mining Concept etc. These approaches are known as meta heuristic approaches that are used in various software testing process, to achieve reliability and quality [2].

## II. AI APPROACH

### Ant Colony Optimization (ACO)

ACO is an algorithm that simulates the behavior of real ants. The first ACO technique was known as Ant System [4]. The approach uses the concept of the capability of ants to locate the shortest route path from their destination to the food source. The method used to communicate information between the individuals is related to paths, and used to decide which path to follow, consists of pheromone trails. For marking the path, the moving ant lays some pheromone on the path. While another ant moves around, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, and then reinforcing the trail with its own pheromone to marking the path for other one. ACO technique has been applied to solve many optimization problems like travelling salesman problem, knapsack problem, distributed networks, telecommunication networks, and test data generation [5, 6].

### Genetic algorithm

GA has emerged an optimization technique and searching method. GA is used to evaluate historical information to shortest the search into the region for better performance within the search space. Genetic algorithm based on the concept of genetic structure and behavior of the chromosomes of individuals within a population [7]. Three operators are used on its population: i) Selection ii) Crossover and iii) Mutation. Genetic Algorithms work efficiently when the search space is large in size, composite and not clearly understood, when domain knowledge is inadequate or expert knowledge is difficult to encode. Genetic algorithms are used in the field of software testing for the generation of test cases using graph theory and test data generation using a weighted control flow graph (CFG) [8,9].

### Tabu Search

Tabu search is a metaheuristic algorithm which is used to solve optimization problems. It uses the concept of memory structure that stores the information about the visited solutions and set of rules. Based on the problem, number of parameters are considered by Tabu search such as: objective function (used to calculate the cost of solution), a candidate list (for choosing good neighbor candidates) and memory structure (used to store the information about the recent moves and the frequently moves of the search). Tabu search is used for automatic generation of structural software testing and is a cost effective approach to optimize the cost of software process that will provide maximum code coverage (in terms of path, conditions and loop coverage) [10,11].

### Bee Colony

Bee Colony algorithm concept is extracted from the foraging behavior of honeybees. It is a combinatorial optimization and continuous optimization search algorithm used to perform neighborhood search combined with global search. This strategy is used to look for the best solution for an optimization problem. Artificial bee colony search technique is used for automatic generation of structural software test for the small input domains. This method generates optimal number of test cases for execution on software under test [12].

### Data Mining

Data mining is a process of extract of useful information and knowledge from a large database which is further applied for decision making. The parameters included in data mining are clustering, association, path analysis, classification and regression. Data mining algorithms can be efficiently used for automation of tested systems. Mined tools like Pr-Miner, MAPO, PARSEWeb, mined exception-handling rules [13,14,15] are used on software data which includes source code, traces of execution process, history of the code revision, bugs reports, documentation etc. mining models are also used to identifying the equivalence classes in system inputs.

## III. CONCLUSION

Software testing is the process of validation and verification of the software product. Effective software testing will contribute delivery the reliable and quality oriented software product, more satisfied users. To achieve this many metaheuristic approaches are used in software testing process for test sequence generation, automation of testing, measuring quality, checking reliability, etc.

## REFERENCES

- [1] Dorigo, M., Maniezzo, V., und Colorni, A.: The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*. 26(1):29–41. 1996.
- [2] Doerner, K., Gutjahr, W. J., “Extracting Test Sequences from a Markov Software Usage Model by ACO”, *LNCS*, Vol. 2724, pp. 2465-2476, Springer Verlag, 2003.
- [3] Howe, A. E., Mayrhauser A. V., and Mraz, R. T., “Test Case Generation as an AI Planning Problem”, *Automated Software Engineering*, Vol. 4, pp 77-106, 1997.
- [4] Dorigo M., Maniezzo, V., Colorni, A., “Positive Feedback as a Search Strategy”, Technical Report No. 91- 016, Politecnico di Milano, Italy, 1991.
- [5] H.Li, and C.Peng Lam, “Software Test Data Generation Using Ant Colony Optimization,” In *Transactions on Engineering, Computing and Technology*, 2005.
- [6] P.Zhao, P.Zhao, and X.Zhang, “New Ant Colony Optimization for the Knapsack Problem,” in the *Proceedings of the 7th International Conference on Computer-Aided Industrial Design and Conceptual Design*, Nov 2006, pp 1-3.
- [7] H.Li, and C.Peng Lam,” An Ant Colony Optimization Approach to Test Sequence Generation for State-Based Software Testing”, *Proceedings of the Fifth International Conference on Quality Software (QSIC’05)* 1550-6002/05, 2005.
- [8] Praveen Ranjan Srivastava, Tai-hoon Kim, “Application of Genetic Algorithm in Software Testing”, *International Journal of Software Engineering and Its Applications* Vol. 3, No.4, October 2009.
- [9] C.C. Michael, G.E. McGraw, M.A. Schatz and C.C. Walton, “ Genetic Algorithm for dynamic test data generations”, Technical report RSTR-003-97-11, May 1997.
- [10] E. Diaz, J. Tuya and R. blanco “Automatic Software Testing Using a Metaheuristic Technique Based on Tabu Search,” *Proceedings 18th IEEE International Conference on Automated Software Engineering*, Montreal, 2003, pp. 301-313.
- [11] E. Díaz, J. Tuya, R. Blanco and J. J. Dolado, “A Tabu Search Algorithms for Structural Software Testing,” *ACM proceeding*, Vol. 35, No. 10, October 2008, pp. 3052- 3072.
- [12] J. D. McCaffrey, "Generation of Pairwise Test Sets using using a simulated bee colony algorithm ", *IEEE International Conference on Information Reuse & Integration*,pp. 115-119, Aug 2009.
- [13] D. R. Engler, D. Y. Chen, and A. Chou. Bugs as inconsistent behavior: A general approach to inferring errors in systems code. In *Symposium on Operating Systems Principles*, pages 57–72, 2001.
- [14] S. Thummalapenta and T. Xie. Parseweb: a programmer assistant for reusing open source code on the web. In *ASE*, pages 204–213, 2007.
- [15] H. Zhong, T. Xie, L. Zhang, J. Pei, and H. Mei. Mapo: Mining and recommending api usage patterns. In *ECOOP 2009 - Object-Oriented Programming, 23rd European Conference*, pages 318–343, 2009.
- [16] S. Thummalapenta and T. Xie. Alattin: Mining alternative patterns for detecting neglected conditions. In *ASE*, pages 283–294, 2009.