Test Suite optimization Using Artificial Bee Colony and Adaptive Neural Fuzzy Inference System

Gurcharan Kaur^1, Bhupender Yadav^2

Department of Computer Science and Engineering, Gurgaon Institute of Technology and Management, Gurgaon, Haryana

Abstract: ***Software test suite optimization is one of the essential issue in software engineering analysis.This paper deals with the improvement in quality of software by software Test Suite Optimization using Artificial Bee Colony (ABC) based novel search technique and technique determine the software development time accurately by proposed Adaptive Neuro Fuzzy InferenceSystem (ANFIS).In this approach, ABC combines the equidistant behaviour of these three bees makes generation of feasible self-supporting paths and also makes software test suite optimization faster.Test Cases are generated using Test Path Sequence Comparison Method as the fitness value objective function. This research also presents an approach for the automated generation of feasible self-supporting test path based on the priority of all edge cover.***

Keywords: ABC, ANFIS, SDLC,SUT

1. Introduction

Software Engineering activity, covers those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of conflicting requirements of the various stakeholders, documenting, analysing, validating and managing the software or the requirements of system. Software Engineering activity sometimes become critical for the success of system or project [1].

The main goal of Software Engineering Life Cycle is to improve the quality, make system reliable and always follow the stages in Software Development Life Cycle (SDLC). It is a process which is used by system analyst to develop the system which is according to the user requirements.

The analyzation process must be done very carefully.Sometimes it becomes very tired and long process. The new introduced systems change the environment and relationships between people, so it is important to recognize all the stakeholders, take into account all their needs and ensure they understand the implications of the introduced systems. Analysts can employ several techniques to extract the requirements from the customer. Prototyping may be used to develop an example system that can be demonstrated to stakeholders.

Another major process in Software Engineering Life Cycle is testing. Software Testing is a process of assessing a system with the goal whether it satisfies the specified requirements of the user or not [2]. Testing is to put a system into an effect in order to identify any errors, or missing requirements in comparison to the actual requirements. Software testing targets in finding errors in code, weakness indesign, and misconception in requirements that appearas incorrect behaviour of the system.

Generally the softwaretesting can be divided into two categories based on the levelof knowledge on the system under test: white-box or black-boxtesting. Furthermore, the testing activity can be categorisedbased on the development phase when the testingtakes place and level of abstraction it investigates: module,integration, unit, or system testing, for instance.

A test suite is a set of several test cases for a component or system under test,where the post condition of one test is often used as the precondition for the next one [5]. The test suite optimization process involves generation of effective test cases in a test suite that can cover the given Software under Test (SUT) within less time.

Random test cases are generated by a random walk in the SUTstarting at any source node and terminating at a destination or sink node constrained by the edge probabilities. But this leads to an unlimited source of tests in which the selection of efficient test cases is very difficult [3]. So with the help of ABC and ANFIS we can easily find the best optimal path from source node to destination node.

2. Artificial Bee Colony Algorithm (ABC)

Artificial Bee Colony Algorithm (ABC) is a new swarm intelligence algorithm which is proposed by Karaboga in 2005 and inspired by the behaviour of honey bees. ABC is developed based on inspecting the behaviors of real bees on finding nectar and sharing the information of food sources to the bees in the hive.

The Algorithm uses the control limits such as colony size and maximum cycle number. The agents in ABC algorithm are Employed Bee, Onlooker Bee andScout Bee. The number of employed bees is equal to the number of food source. The Food source position is equal to the possible solution to a problem. The amount of nectar of food source is equal to the quality of solution.

The key steps for algorithms are:
Initialize.
Repeat
1) Place the employed bees on the food sources in the memory;
2) Place the onlooker bees on the food sources in the memory;

3) Send the scouts to the search area for discovering new food sources.

UNTIL (requirements are met).

3. Adaptive Neural Fuzzy Inference System

An adaptive neuro-fuzzy inference system (ANFIS) is a class of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. An adaptive neuro-fuzzy inference system (ANFIS) is also known as adaptive network-based fuzzy inference system.The technique was spread in the early 1990s[5][6].It integrates both neural networks and fuzzy logic principles. it has potential to capture the benefits of both in a single framework. Its inference system equivalent to a set of fuzzy IF–THEN rules that have learning capability to approximatenon-linear functions[7].

ANFIS in a more efficient and optimal way, one can use the best parameters obtained by genetic algorithm[8].Neural networks are good at training the dataset, but the clustering and feature input is somewhat weak in neural networks. Fuzzy logic based models are good at featuring and clustering but the training of datasets is not provided, hence ANFIS is combination of both.

4. Proposed Strategy

We Propose an ABC based search algorithm to generate test data. In this research work, theFunctionality of the bee is extended to do the testing and monitoring activity so that it minimize the manualwork and upgrade the confidence on the software by testing it with the coverage of the given software.Bee colonyconsists of three types of bees, namely scout bees, which randomly searches for the food sources, onlookerbee decides which food sources to be explored next from the list food sources given by scout bees, and lastly employee bees will search for new food source in neighbourhood of exhausted food source.

All steps in finding optimal weights for network are shown in proposed ABC algorithm framework in Figure 1. The figure shows how to find and select the best weights .The initialization of weights was compared with output and the best weight cycle was selected by scout bees' phase.

The bees (employed bees, onlooker bees) would continue searching until the last cycle to find the best weights for networks. The food source of which the nectar was neglected by the bees was replaced with a new food source by the scout bees.

Every bee (employed bees, onlooker bees) would produce new solution area for the network and the Greedy Selection would decide the best food source position. The proposed frameworks can easily train software defect data for prediction task by finding optimal network weights for ANFIS.

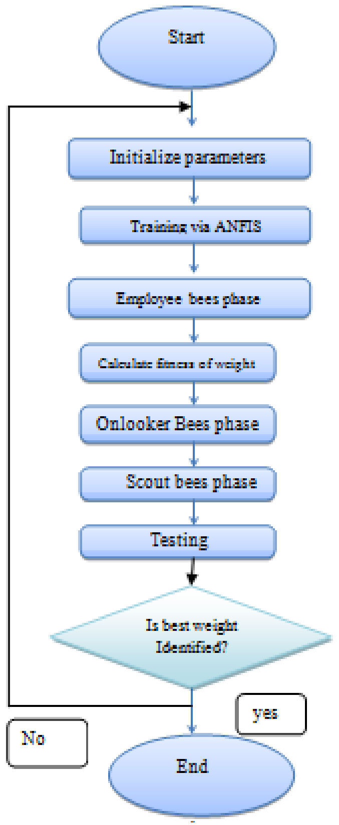

Figure 1: ABC Algorithm Framework

5. Enhancement Algorithm

The following is the detailed algorithm
1. Initialize the test cases which is performed by the search bee.
2. Search for an executable state and evaluate the test nodes
3. Initialize the current traversal path as cycle=1
4. Repeat
5. Produce new test cases v_{ij} in the neighbourhood of x_{ij} for the search agent
using the formula
$X_{ij}=min_j+rand\ (0,\ 1)*(max_j-minj)$
• Xij is the initial test case
• minj is the minimum no. of test cases
• maxj is the maximum no. of test cases
• rand () is a random number generation function which selects either 1 or 0 randomly.
6. Generate the test data using equivalence partitioning and boundary value analysis [10].
7. Apply greedy selection process [9] for the generated test data.
8. Traverse the SUT with the generated Test Data and calculate the fitness value.
9. The onlooker selects the test cases with the highest fitness value and abandons the rest.
10. The traversal process is carried out till a particular test data with 100% fitness value and 0% fitness value is produced
$Pi = (fit_i)/\Sigma\ (fit_i)$
Where Pi is the probability function which signifies the probability with which the i_{th} test data traversers a independent test path successfully.
11. Add the test case to the optimal repository.

12. In the next iteration scouts generate the new test data and go to step 5.

6. Experimental Result

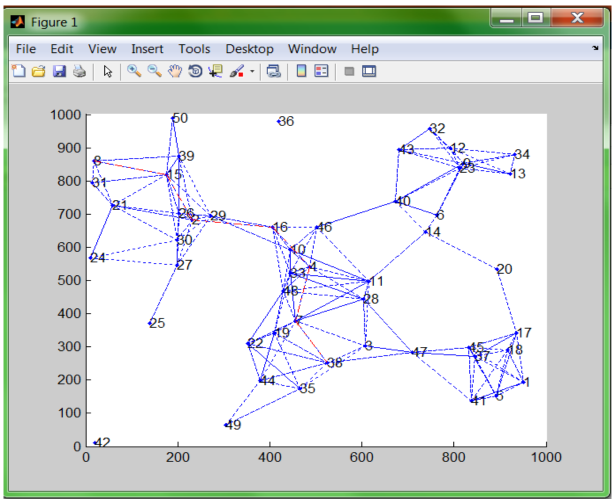

Figure 2: Result

Our approach uses the ABC Test Suite Optimization tool to optimize the Test Cases by employing the ABC algorithm. The ANFIS tool considers a program as an input to generate independent paths. Using the generated independent paths Test Cases are traversed along the paths with the help of ABC algorithm. By doing so the test cases with maximum coverage (High fitness Value) are recognized. Finally the optimal Test Suite is generated.

7. Conclusion

In this work, we proposed an approach which uses fewer iterations to complete the task. The results give the more scalable test suite. It also requires less time to complete the task and best in achieving the global optimal solution.

8. Scope for Future Research

This researchh work may be extended for further research in the following dimensions/environments: This research work concentrates on optimization based on test adequacy criteria such as coverage and fault revealing capability of the test cases. In the future research work, this can be done based on other test adequacy criteria such as data based and flow based measures.

References

[1] Kotonya, G. and Sommerville, I. 1998. *Requirements Engineering: Processes and Techniques* Chichester, UK: John Wiley and Sons.
[2] http://www.tutorialspoint.com/software_testing/
[3] Kaner, Cem (November 17, 2006). "Exploratory Testing"(PDF). Florida Institute of Technology, *Quality Assurance Institute Worldwide Annual Software Testing Conference*, Orlando, FL. Retrieved November 22, 2014.
[4] Naresh Chauhan "Software Testing Principles and Practises" (2012)
[5] Jang, Jyh-Shing R (1991). *Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm* (PDF). Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, July 14–19 2. pp. 762–767.
[6] Jang, J.-S.R. (1993). "ANFIS: adaptive-network-based fuzzy inference system". *IEEE Transactions on Systems, Manand Cybernetics* 23 (3).doi:10.1109/21.256541
[7] Abraham, A. (2005), "Adaptation of Fuzzy Inference System Using Neural Learning", in Nedjah, Nadia; de MacedoMourelle, Luiza, *Fuzzy Systems Engineering: Theory and Practice*, Studies in Fuzziness and Soft Computing 181, Germany: Springer Verlag, pp. 53–83,doi:10.1007/11339366_3
[8] Tahmasebi, P. (2012). "A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation" (PDF). *Computers & Geosciences* 42: 18–27.
[9] Adi Srikanth, Nandakishore J. Kulkarni, K. Venkat Naveen, Puneet Singh, and
[10] Praveen Ranjan Srivastava," Test Case Optimization Using Artificial Bee ColonyAlgorithm", ACC 2011, Part III, CCIS 192, pp. 570–579, 2011
[11] Dr. ArvinderKaur, ShivangiGoyal,"A Bee Colony Optimization Algorithm for Fault Coverage Based Regression Test Suite Prioritization",International Journal of Advanced Science and Technology Vol. 29, April, 2011
[12] Blackwell, A. H.; Manar, E., eds. (2015). "Prototype". *UXL Encyclopedia of Science* (3rd ed.). Retrieved 13 July 2015.