Automated Generation of Independent Paths and Test Suite optimization Using Artificial Bee Colony
Gurcharan Kaur1 Bhupender Yadav2
1,2Department of Computer Science
Gurgaon Institute of Technology and Management, Gurgaon, Haryana

Abstract— Software test suite optimization is one of the most important problems in software engineering research. This paper deals with the concept to find the Automatic Generation of Feasible Independent Paths and Software Test Suite Optimization using Artificial Bee Colony (ABC) based novel search technique and Adaptive Neuro Fuzzy Inference system (ANFIS). In this approach, ABC combines both global search methods done by scout bees and local search method done by employed bees and onlooker bees. The parallel behaviour of these three bees makes generation of feasible independent paths and software test suite optimization faster. Finally, this paper compares the efficiency of ABC based approach with various approaches.

Keywords: ANFIS, ABC, SDLC

I. INTRODUCTION
Software Engineering Process, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. The concept generally refers to computer or information systems. These methodologies form the framework for planning and controlling the creation of an information system [1]. The main aim of Software Engineering Life Cycle is to achieve a high quality, high reliable software and always follows a software development life cycle process. Systems Development Life Cycle (SDLC) is a process used by a systems analyst to develop an information system, including requirements, validation, training, and user (stakeholder) ownership. Any SDLC should result in a high quality system that meets or exceeds customer expectations, reaches completion within time and cost estimates, works effectively and efficiently in the current and planned Information Technology infrastructure, and is inexpensive to maintain and cost-effective to enhance. One of the major activity in every software development life cycle is the software testing. Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs [2]. So, this paper mainly gives foundation for generating automated testing by the automated generation of independent test paths [3] and test suite optimization. Software testing [2] mainly includes two major methods i.e., black box testing and white box testing. White box testing (or structural testing or clear box testing or glass box testing or transparent box testing) is to test thoroughly the internals of the particular program module. Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The testers are only aware of what the software is supposed to do, not how it does it.

Test data generation is an un-decidable problem and can be non-deterministic that is NP-hard [4,5] or a solution exist which is not practical. The highly non-linear structure of software presents a challenge to search algorithms for finding optimal and efficient test data from a complex, discontinuous, nonlinear input search space. The basic approach for evaluating input value sets in dynamic structural test data generation methods [6] can be summarized as:
- Represent a set of input values as an initial solution,
- Apply these input values to the code and observe the generated path, and
- Compare the generated path with the desired path and calculate fitness values.

This paper presents an ABC based search algorithm to generate test data. In this research work, the functionality of the bee is extended to do the testing and monitoring activity so that it reduces the manual work and improves the confidence on the software by testing it with the coverage of the given software. Bee colony consists of three types of bees, namely scout bees, which randomly searches for the food sources, onlooker bee 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 [7, 8].

II. BACKGROUND WORK
The research on software testing problems has centered mainly on the Independent Path Generation and Software Test Suite Optimization. Many intelligent swarm computing software test techniques are extensively proposed and every algorithm has its own strengths and weaknesses. The major observations derived as part of the background study are:
- Neural Networks has difficulties in programming and algorithmic complexity and is a black-box data processing structure.
- Tabu search has difficulties in the amount of memory required to avoid stuck up at local optima and short term memory to remember all the test cases in the current search [9].
- Genetic Algorithms has difficulties in giving stable results (stuck up at local optima); the convergence is slow and has non-explicit memorization of best individuals [10].
- Ant Colony Optimization has drawbacks of higher length test sequences and repetition of nodes within the same sequence [3]. In initial stages, it starts searching slowly, lacks pheromone and easily tends to local optimum, has premature convergence when quantity is too high [11, 12].
In ABC (Artificial Bee Colony) optimization the software development time estimation is very important factor in case of quality improvement but there is no such method has been discussed for this.

III. PROPOSED STRATEGY

We Propose an ABC based search algorithm to generate test data. Artificial Bee Colony Algorithm (ABC) is nature-inspired metaheuristic, which imitates the foraging behaviour of bees. ABC as a stochastic technique is easy to implement, has fewer control parameters, and could easily be modify and hybridized with other metaheuristic algorithms. Due to its successful implementation, several researchers in the optimization and artificial intelligence domains have adopted it to be the main focus of their research work. In this research work, the functionality of the bee is extended to do the testing and monitoring activity so that it reduces the manual work and improves the confidence on the software by testing it with the coverage of the given software. Bee colony consists of three types of bees,[13] namely

- Scout bees, If the profitability of food source cannot be improved and the times of unchanged greater than the predetermined number of trials, which called “limit” and specified by the user of the ABC algorithm, the solutions will be abandoned by scout bees. Then, the scouts start to randomly search the new solutions.
- Onlooker bees observe the waggle dance in the dance area and calculate the profitability of food sources, then randomly select a higher food source. After that onlooker bees carry out random search in the neighbourhood of food source.
- Employed bee flies to a food source and finds a new food source within the neighbourhood of the food source. The higher quantity food source will be selected. The food source information stored by employed bee will be shared with onlooker bees.

The main steps of the algorithm are given below:
1. Initial food sources are produced for all employed bees
2. REPEAT
   - Each employed bee goes to a food source in her memory and determines a neighbour source, then evaluates its nectar amount and dances in the hive
   - Each onlooker watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source. After choosing a neighbour around that, she evaluates its nectar amount.
   - Abandoned food sources are determined and are replaced with the new food sources discovered by scouts.
   - The best food source found so far is registered.
3. UNTIL (requirements are met)

Fig. 1: Types of Bees

Moreover in proposed strategy there are some steps 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 (Adaptive Neuro Fuzzy Inference System). An adaptive neuro-fuzzy inference system or adaptive network-based fuzzy inference system is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. The technique was developed in the early 1990s. Since 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 corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions. Hence, ANFIS is considered to be a universal estimator.

IV. CONCLUSION

In this paper, The Proposed Strategy deals with improvement in quality of software by software Test Suite Optimization using Artificial Bee Colony (ABC) based novel search technique and method estimates the software development time accurately by proposed Adaptive Neuro Fuzzy Inference System (ANFIS).

REFERENCES

Ant and Particle Swarm Optimization, December, 2007, pp. 532-564.


