PREDICTING SOFTWARE RELIABILITY AND QUALITY USING ANT COLONY OPTIMIZATION TECHNIQUE WITH TRAVELLING SALESMAN PROBLEM FOR SOFTWARE PROCESS

D. Hema Latha¹, Prof. P. Premchand²

Abstract-In the context of software engineering, software quality measures how good software is designed (quality of design), and how good the software conforms to that design (quality of conformance). Software quality is described as the ‘fitness for purpose’ of a piece of software. Software reliability is a failure free operation of software for a specific period of time under specified environment. Software reliability is defined as the probability with which the software will operate without any failure for a specific period of time in a specified environment. Software reliability and quality Software reliability and quality prediction is very challenging in starting phases of life cycle model. Software reliability and quality, when estimated in early phases of software development life cycle, saves lot of money and time as it prevents spending huge amount of money on fixing of defects in the software after it has been deployed to the client. Prediction of Software reliability and quality has thus become an important research area as every organization works towards to produce reliable and good quality software and error or defect free software. There are many software quality and reliability growth models that are used to estimate or predict the reliability and quality of the software. These models help in developing robust and fault tolerant systems.

In the past few years many software reliability models and quality models have been proposed for assessing reliability and quality for software, but developing accurate reliability and quality prediction models is difficult due to the recurrent or frequent changes in data in the domain of software engineering. As a result, the software reliability prediction models built on one dataset show a significant decrease in their accuracy when they are used with new data. The main objective of this paper is to introduce a model for predicting software reliability and quality with a new approach that optimizes the accuracy of software reliability and quality predictive models when used with real data. In this research paper, Ant Colony Optimization Technique (ACOT) with Travelling Salesman Problem (TSP) is proposed to predict software reliability and quality based on the data collected from literature. An ant colony system by combining with Travelling Sales Problem (TSP) algorithm has been used, which has been changed by implementing different algorithms and extra functionality, in an attempt to achieve better software reliability and quality results with new data for software process.

Keywords: Software Reliability, Software Quality, Bio-inspired Computing, Ant Colony Optimization technique (ACO), Travelling Salesman Problem (TSP).

1. INTRODUCTION

There is a prerequisite of the enormous scope of effort, time and currency to arrange and build up any feasible software apart from the human resource and their organization. For outstanding increasing competition, today’s profitable conditions have become very dynamic. Corporate industries need extremely fast to unstable needs of the market. Hence, software engineering which emphasizes with all these domains has become an important study in research area. The software crisis is defined as a mismatch between what the software can deliver and the capacities of computer systems, as well as the expectations of their users and where software problems cause the system tasks to be delayed, expensive, and/or not amenable to the user’s assumptions. At each stage of software development, the software is checked for reliability, quality, security, portability, usability, effective cost and response time.

Developing trustworthy software from the user’s perspective is a demanding profession for all software engineers. However, Software Reliability [1], [2], [3] is a significant aspect influencing system reliability. The following four practical aspects which are related to achieve reliable software systems and these aspects furthermore are treated as four fault Lifecycle techniques:

1) Fault avoidance: to avoid, by building, error existence. 2) Fault elimination: to identify, by confirmation and proof, the presence of faults and eliminate them. 3) Fault tolerance: to specify, redundancy, facility conforming to the requirement in spite of faults having rising. 4) Fault/failure Predicting to estimate: by the assessment, the occurrence of faults and consequences of failures. Quality [4], [5] is an important feature of reliability.

¹ Research Scholar, Dept of Computer Science, Rayalaseema University, Kurnool, Andhra Pradesh, India
² Professor, Dean, Faculty of Informatics, Dept of Computer Science and Engineering, UCE, Osmania University, Hyderabad, TS, India
Software reliability is the probability that software will not cause failure of a system for a particular point in time under particular circumstances. The probability is a function of the inputs to and use of the system as well as a function of the existence of faults in the software. According to ANSI, Software Reliability [6], [7] is defined as: “the probability of failure free software operation for a particular period of time in a particular environment”. Software reliability evaluation is significant to evaluate and forecast the trustworthiness and performance of software systems. Reliability representation is a crucial parameter of the reliability evaluation procedure and it also validate whether a product reaches the milestone of its reliability objective and is ready for deployment. The fundamental intention of most of software reliability models is making them to understand distinctiveness and reasons to fail software, and try to enumerate software reliability.

The current article lay emphasis on about a bio inspired computing technique Swarm Intelligence known as the Ant Colony Optimization Technique [8], [9] to predict software reliability. The anticipated method is employed into a TSP problem to predict software reliability and quality. And, thus, the focus of the discussion to be presented here is an ACO for discrete optimization that has been used to predict software reliability and quality using the Travelling Sales Person Problem where failure data is given as input and the result is calculated through Mean Time to Failure (MTTF) and Mean Time Between Failure (MTBF) to predict the reliability.

The rest of the paper is organized as follows. Proposed methodology with an algorithm and flow chart is explained in section II. Concluding remarks are given in section III.

2. METHODOLOGY

2.1. Ant Colony Optimization Technique with travelling salesman problem

Ant Colony [10-13] is one of the techniques of bio inspired computing. The main concept of this technique is that the self-organizing rules which allow the highly synchronized behavior of real ants can be utilized to manage populations of artificial agents that cooperate to solve computational problems. Various distinctive attributes of the behavior of ant colonies have inspired different kinds of ant algorithms. Examples are foraging, distribution of labor, issue sorting, and cooperative transport. Ants coordinate their activities via stigmergy, a form of implicit interaction mediated by changes in the environment. For example, a foraging ant deposit a chemical on the ground which raises the probability that other ant will follow the same path. Biologists have presented that many colony-level behaviors witnessed in social insects can be described through relatively simple models in which only stigmeric communication is present. In other words, biologists have shown that it is often sufficient to consider stigmeric, indirect communication to explain how social insects can attain self-organization. The notion behind ant algorithms is to use a form of artificial stigmergy to coordinate societies of artificial agents. One of the most effective examples of ant algorithms is known as “ant colony optimization”, or ACO. ACO is motivated by the foraging behavior of ant colonies, and targets discrete optimization problems. The ants may deposit a pheromone on the ground while returning back to their nests. The ants follow with high probability pheromone trails their sense on the ground.

Each Ant evaluates the next move to another vertex based on Gambardella et al., [14 - 16],

\[
p^i_k = \begin{cases} 
\frac{(\tau_{ij})^\alpha \eta_{ij}^\beta}{\sum_j (\tau_{ij})^\alpha \eta_{ij}^\beta} & \text{if } j \text{ is allowed } \text{ for } k \\
0 & \text{otherwise}
\end{cases} \tag{1}
\]

\(p^i_k\) is the probability for a worker K to move to vertex “ij”

\(\tau_{ij}\) is the amount of pheromone deposited on edge to “ij”

\(\eta\) is the inverted distance, describes how fast ants select their path.

The tour cost of each ant is given by \(d_{ij}\) the tour cost from the city i to city j (edge weight) is calculated and hence the shortest path is found. This is applied to the Travelling Sales Person Problem and optimized solutions are obtained using

\[
d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2}
\]

The amount of pheromone deposited by each ant is given by
The ACO algorithm combined with travelling salesman problem, which has been proposed based on the study that real ants are skilled in finding the shortest path from a food source to the nest without using visual signals. From the originating point the ants start the tour selecting randomly any path. While the ant travelling from source to destination, each ant leaves pheromones, if the pheromones are strong that is taken as good quality. The ACO algorithm is as follows:

Steps for implementation

1. Input the Quality Test details with software data.
2. Initialize all the parameters with input values.
3. For each ant randomly select the node to start the tour.
4. For each ant keep selecting the next best route until all paths are covered.
5. Selection of the best path based on the pheromone strength depends on:
   a. Maximum Unique pheromone evaporation found.
   b. Maximum pheromone strength
6. Terminate the search/iteration process when all paths are covered.
7. For each ant ‘k’ formula to update the pheromone for each state transition ‘xy’ covered is as follows:
   \[
   \tau_{xy}^{k} = 1 - \rho \tau_k + \frac{\beta}{\alpha \times \text{Paths covered}} \]

Where,
- \( \tau_{xy}^{k} \) is the amount of pheromone deposit on a state transition \( xy \)
- \( \rho \) is the pheromone decrease rate
- \( \alpha \) is the total test cases covered by ant \( k \).
- \( \beta \) is the pheromone deposition rate.

8. Pheromone Updation
9. Deposit Pheromone on the each path covered by ants in their solution
10. Reduce Pheromone on each edge by the rate of decrease factor. (Evaporate)
11. Determine the final path.
12. At each iteration, selection of best path depends on:
   a. Minimum Execution time
   b. Maximize the pheromone strength
13. At each node the weight of the pheromone deposited at the path is stored and quality test is based on the pheromone strength.

Probability to find best path = \( n \)

\[
\text{execution time} \times (\text{Phermonone trail } i, j)
\]

Paths covered \( k \beta \alpha \times \text{Phermonontrail } i, j \)
2.3 ACO Flow Chart to predict software reliability and quality
Flow chart for Ant Colony Optimization (ACO) with Travelling Salesman Problem (TSP) to predict software reliability and quality is shown in figure 1.

Figure 1. Flow chart of ACO algorithm
3. CONCLUSION
In this research paper, the authors proposed a methodology to predict reliability and quality for a software. ACO is a comparatively new metaheuristic concept for resolving tough combinatorial optimization problems. Simulated or Artificial ants realize a random construction heuristic approach which compose a probabilistic judgment. The cumulated search practice is taken into consideration by the adapting the pheromone trail. ACO exhibits great performance when used for “ill-structured” problems like network congestion and routing. When ACO local search is implemented is mandatory to obtain optimistic results.

Ant algorithms fit into a group of Meta heuristics, which are known for range of applications to realistic problems encountered in scientific, business applications and industrial scenarios. A variety of applications depicted in this study focuses on ant algorithms that can be applied to plenty of sensible situations. The algorithms employed in this work are inspired by an observation emphasizing on real ants nature i.e. foraging and searching abilities that can provide good answers to genuine and real time optimization and software reliability solutions. The indirect interaction and the co-operative communication of the simulated ant agents is enthused from their actual living counterpart, exhibiting great elasticity and sensitivity to vibrant problems. The function of these programs and investigational validations are enormously studied owing to their potential to offer most favorable generic solutions to specific complex problems such as imaging problems, local search, compression theory, image mapping and searching databases.

The exploration is still in progress as many of the facets of ACO algorithm are still to be unraveled. It is expected that this study stimulates further discussion for better reliability solutions.

4. FUTURE WORK
The prospects of Ant algorithm based applications with reference to the geometric tolerance amalgamation and distribution of the potential and probable regions of exploration.

The investigation of more efficient pheromone models might reduce the need of comprehensive intensification phases and the future must evolve theoretical development of models for experimentation and for creating effective models. 33

5. REFERENCES