

ENHANCEMENT AND COMPARISON OF ANT COLONY OPTIMIZATION FOR SOFTWARE RELIABILITY MODELS

¹Latha Shanmugam and ²Lilly Florence

¹Department of MCA, MVJ College of Engineering, Bangalore, Karnataka, India

²Department of MCA, Adiyamaan College of Engineering, Hosur, Tamil Nadu, India

Received 2013-07-08, Revised 2013-07-12; Accepted 2013-08-01

ABSTRACT

In Common parlance, the traditional software reliability estimation methods often rely on assumptions like statistical distributions that are often dubious and unrealistic. The ability to predict the number of faults during development phase and a proper testing process helps in specifying timely release of software and efficient management of project resources. In the Present Study Enhancement and Comparison of Ant Colony Optimization Methods for Software Reliability Models are studied and the estimation accuracy was calculated. The Enhanced method shows significant advantages in finding the goodness of fit for software reliability model such as finite and infinite failure Poisson model and binomial models.

Keywords: Ant Colony Optimization, Parameter Estimation Method, Software Reliability Model

1. INTRODUCTION

The fundamental idea of software reliability modeling is mainly to predict the reliability of the software with its failure data. Until, the late 60s, the attention of scientists and engineers were directed towards the hardware reliability (mechanical and electronic systems). From the 70s, with the continuous growth of technology software applications became the center of many studies. The possibility to create complex dependencies and better cost/price ratios compared to hardware led to a wide range of software applications. Today, computers and mobiles are used in everyday life, in industry, banks, large systems like power distribution, traffic, water supply. Computers are used even in life critical applications in hospitals; they control air traffic and airplane flight, where failures could lead to catastrophes and loss of many lives. Today, the software needs to support the recent technologies like cloud computing (Saravanan *et al.*, 2013), where the reliability is more challenging.

Due to increasing dependence and demand for Software, it is necessary to develop and maintain its reliability. Software functionality is becoming crucial from the aspects of reliability, safety of human lives and

security issues as well. Specification, evaluation and verification of these quality characteristics are important issues for both developers and users of the system.

Two commonly used traditional methods in parameter estimation are maximum likelihood and least squares method. As most Software Reliability Growth Models are non linear functions, various other Parameter Estimation methods have been proposed by several researchers in the literature. They introduced a method based on Expectation and Maximization principle, applied it for Hybrid Software Reliability Model, Discrete Software Reliability Models, Markov Modulated Reliability Models. The Parameters of Reliability include terminal reliability, broadcast reliability and network reliability. SSMC method was used for parameter estimation. This method was useful when complexity of system makes it impossible to formulate exact models. SSMC method was demonstrated as a good estimator for all the three types of the reliability and reliability bounds. Zeljkovic *et al.* (2011) given a Technical Review on Software Reliability Models and Parameter Estimation, in that Parameter Estimation was done with failure rate and reliability data. All these methods did not receive much attention in recent research, due to non linear function of Software Reliability Models.

Corresponding Author: Latha Shanmugam, Department of MCA, MVJ College of Engineering, Bangalore, Karnataka, India

This problem was overcome by a very effective Parameter Estimation Method, which is Ant Colony Optimization method. It is a brand-new bionic simulated evolutionary algorithm, which has been applied to many applications such as Scheduling, Set, Routing of Vehicles and Assignment application.

2. SOFTWARE RELIABILITY

Software Reliability is considered as the probability of operation which is failure-free over a specified period of time in a given environment to achieve the required purpose. The outcome of this is different for various users based on the system used. It cannot be defined objectively because of reliability measurements which are quoted out of context. It requires operational profile for its definition as the expected software usage pattern is defined by operational profile. It considers fault consequences and the reliability of the system depends on the severity of the fault.

2.1. Software Reliability Model (SRM)

Software reliability models are statistical models which can be used to make predictions about a software system's failure rate, when the failure history of the system is given. The models make assumptions about the fault discovery and the removal process. These assumptions determine the form of the model and the meaning of the model's parameters.

It is an important tool in quality management and release planning. There is a large number of different models that often exhibit strengths in different areas. For all the Models there are some assumptions we can state based on the software failure process. Specified assumptions should be checked before applying the model (Goseva-Popstojanova and Trivedi, 2000). For Several assumptions such as, program failures should occur independently at the time of testing, the software should be operated in a similar manner as the expected operational usage, The set of inputs per test run is to be selected randomly, all the failures are observed, a detected Fault is Immediately to be corrected, the total number of faults in the program is finite, no Fault Occurs at the start of the test, no new Faults are introduced during the fault removal Process, failure rate to be decreased with test time, failure rate is should be Proportional to the number of remaining faults, reliability is a Function to calculate the number of remaining faults, test effort should be used as a basis for failure rate are some of the assumptions.

3. PARAMETER ESTIMATION METHODS

Parameter Estimation Methods are used to predict the software reliability models. Earlier various Traditional Methods were used to predict the software Reliability Models such as curve fitting, Maximum Likelihood Estimation and Least Squares Estimation methods. The decision making process often delayed due to very late availability of testing results generated by curve fitting analysis which is a major key problem in industries. This problem was overcome by other Parameter Estimation Methods.

There are many approaches that can be used to address the problem with software reliability model that need a lot of data to make stable predictions to be estimated as well as parameters. In Parameter Estimation, Parameters are estimated using the information from the current project or from the past or combined approach. Aljahdali (2011) studied fuzzy model and its members to estimate the expected software faults during testing process which provided high performance modeling capability.

3.1. Maximum Likelihood Estimation (MLE)

The principle of Maximum Likelihood Estimation (MLE), originally developed by R.A. Fisher in 1920s, states that the desired probability distribution is the one that makes the observed data "most likely," which means that one must seek the value of the parameter vector that maximizes the likelihood function. The resulting parameter vector, which is sought by searching the multi-dimensional parameter space, is called the MLE estimate (Liang *et al.*, 2010).

MLE requires usually no distributional assumptions or only minimal assumptions and it is useful for obtaining a descriptive measure for the purpose of summarizing observed data. But it doesn't give basis for testing hypotheses or constructing models. The Popular method to estimate the parameters is the Maximum Likelihood method. In this method, parameters can be estimated by solving the Maximum Likelihood equations (Chen and Beaulieu, 2010), which cannot deliver a solution perfectly. The problem can be resolved by deriving a satisfied condition for each model which assures a unique solution of Maximum Likelihood equations. If the condition is unstable and not satisfied, only the known model parameters can be estimated. Recursive Estimation Algorithm can be used for unknown model parameters; we can have a recursive application of the Maximum Likelihood Method. Shaik *et al.* (2011)

used Maximum Likelihood Estimation Method for test effort functions in Exponential/Power Models and the result was compared with noise.

3.2. Least Squares Estimation (LSE)

Least Square Estimation methods are available to evaluate the set of parameters with the highest probability of being correct for a given set of experimental data. In this method, least squares are taken to estimate parameters by minimizing the squared discrepancies between observed data and their expected values. Compared to ordinary sampling plans, the first failure-censored sampling plan has an advantage of saving both test-time and resources.

Non linear least-square analysis actually comprises a group of numerical procedures that can be used to evaluate optimal values of the parameters in vectors for the experimental data. Several assumptions are implied. In general, the non linear Least-Square procedures consist of an algorithm that used an initial approximation vector (g) of the parameters to generate a better approximation. These better values are then used as initial approximation in the next iteration to yield even better approximation. This process is continued until the approximations converged to a stable set of values. Ishii *et al.* (2012) examined four LSE Methods with applications to software Reliability Prediction. It was shown that LSE methods are still attractive in terms of goodness of fit performance and predictive performance in many cases. LSE methods that are used are Basic weighted least squares method, Modified Weighted Least Squares Method and Root Least Squares Method hence, the Performance was compared with MLE and LSE methods.

Maximum Likelihood Estimation and Least Squares Estimation methods received much attention in recent research analysis.

4. ANT COLONY OPTIMIZATION (ACO)

Ant Colony Optimization (Li *et al.*, 2010) is a technique which uses probability to solve problems where the computations are reduced with the help of graphs to get efficient paths. Observing the behavior of ants when they search for a path from their colony to the food source, this algorithm was used to find the optimal path. This concept has been expanded in solving a wide variety of numerical problems. Many solutions have been suggested based on different aspects of ant behavior. ACO is a bionic simulated evolutionary algorithm. ACO was applied to many optimization

problems like quadratic assignment, protein folding methods and many derived methods were adapted for problems where real variables are used, problems of stochastic nature, multi-targets and other implementations. It has also been used to produce bare-optimal solutions to the travelling salesman problem. The ant colony algorithm can run continuously and adapt to changes in real time.

4.1. Basic Principles of ACO

In search of food, ants set about randomly, after finding, they will come back to the colony leaving a trail of pheromones (**Fig. 1**). The remaining ants, when they find this path will not travel randomly, but follow this trail and return back, if they find food at the end of the trail. Over the period of time the pheromone in the trail evaporates which reduces its attraction strength. If no ants are travelling down the path and back, the pheromones will evaporate faster. On shorter path, ants will travel more frequently as the density of the pheromone is higher than on longer paths. If pheromone evaporates faster there is a chance of avoiding the decision on local optimal solution. Without evaporation, the initial path chosen by ants will become attractive to the ones which are following. If this is the case, the chances of finding a better solution are limited. Finally, the ants will choose a good single path to the food source from their colony. The ACO algorithm is based on adopting this behavior of ants. It is done with the help of simulation in which ants are allowed to walk around in the graph to represent the problem scenario.

4.2. Parameter Estimation Method Based on Ant Colony Optimization

ACO was originally used in the discrete best network path searching; some changes are to be made while adjusting it to parameter estimation problem. ACO uses a positive feedback, parallel and self catalyzed mechanism. It has an excellent robustness and is easy to collaborate with other methods. It shows good performance to any optimization problem and can get rid of traditional optimization method's weaknesses. Coding and operation with ACO is easy. It has a good convergence rate. Therefore, the ACO algorithm is suitable in parameter estimation for software reliability models. Changes should be made to the ant colony algorithm which has been used originally in search of discrete best networks, to suit to the parameter estimation problem. In the current research, the ACO was improved based on the characteristics of SRGMs (Jiang, 2009).

Failure Data was made as a Solution Space which is taken from Musa real time application data Set to predict Parameter Estimation Accuracy. The ACO algorithm (Shanmugam and Florence, 2012) is explained below:

Step 1: Parameter Estimation problem can be translated into functional optimization problem using Least Squares Principle. The objective of the Function (Min J) is to predict estimation accuracy that should have lower value compared to original ACO.

Fitness (J) is calculated Euclidean Distance between the actual failure and predicted failure number:

$$\min J = \sqrt{\sum_{t=0}^T [(m(t) - \mu(t))]^2}$$

Where:

$m(t)$ = The actual number of failure,

$\mu(t)$ = The predicted number of failure by the software reliability model

T = The Total time the software runs and Time 't' gives the failure occurrences.

Step 2: The initial position of each ant is set and solution space has been divided and the other parameters 'a' and 'b' need to be initialized. For example, In G-O Model, if there are two parameters 'a' and 'b', values ranging from $0 < a < a_{max}$, $0 < b < b_{max}$ respectively.

This solution space is divided into $n1$ and $n2$. Let the number of ants living in these areas taken as m randomly.

Step 3: As long as search time is less than N_{max} , each ant automatically can search for best paths. P_0 is Travel Threshold.

The Probability choosing local searching or global searching by the ant k is P_k :

$$P_k = r_k / \tau_k$$

where, $r_k = \tau_k - \tau_{best}$, τ_k is the current path of ant k , τ_{best} is the current best path of ant k .

If $P_k < P_0$ the ant chooses local search. Else global search is chosen. New fitness K is calculated and compared with current fitness J , if it is better than, then the ant's position is updated else not:

$$\text{Update } \tau_k = (1 - \rho) \tau_k + J_k$$

where, ρ is the rate of volatilization rate of pheromone.

5. DESIGN OF ENHANCED ANT COLONY OPTIMIZATION

The Enhanced Ant Colony Optimization algorithm is designed by considering divide and conquers technique. ' α ' represents the number of equal spaces to be split and ' β ' is the number of solution spaces to discard at each stage. Later the Solution Space is determined by dividing it into ' α ' equal intervals. A counter is initialized to keep track of ant's movement. As long as the total number of ants are greater than or equal to 1, the selection of ant search process is to be started and the fitness of an ant is calculated using the Equation 1 and the result is represented by J . By keeping the minimum fitness value of an ant, the remaining ants will be discarded at each stage. The expected outcome is achieved with the optimum time complexity Equation 2 and space complexity Equation 3. The Algorithm is as mentioned as follows:

Begin

Set the Initial Value of α -the number of equal interval space to split.

Set the initial value of β -the number of solution space to discard at each stage.

Divide the solution space into α equal interval. [$N_{sum} = \text{Solution Space} / \alpha$].

Set count = 1.

While $N_{sum} \geq 1$

If $N_{sum} = 1$

$N_{sum} = (\text{count} / 2) + 1$

Count = 0

End if

Choose Ant search in each equal interval space

Calculate fitness J_k :

$$\min J = \sqrt{\sum_{t=0}^T [(m(t) - \mu(t))]^2} \tag{1}$$

Sort the solution space based on J_k

Discard the β higher solution space interval

$N_{sum} = N_{sum} - 1$;

Count = Count + 1;

End While

Output the position of the Ant in the final solution space which has the best fit result.

End

5.1. Comparative Performance of ACO and Enhanced ACO

For this Parameter Estimation Method, six classic Software Reliability Models were selected to calculate estimation accuracy (Table 1). G-O Model, Delay S-Shaped Model, Weibull Model, M-O Model, Jelinski-Moranda Model and Dunane Models were taken for experiment evaluation (Hirose, 2011).

Failure data have been taken from the Musa real time data set to estimate the accuracy of Software Reliability Models. Compared to existing Ant Colony Optimization, the Enhanced ACO is giving the better result of more than 10% estimation accuracy as it should be the lower value. In the following images, it is shown that the Parameter Estimation Accuracy of Enhanced ACO is lower than Existing Ant Colony Optimization algorithm. In the graph outer line shows the Existing Ant Colony Optimization estimation accuracy and the inner line shows the Enhanced Ant Colony Optimization Estimation Accuracy.

5.2. Limitations of Existing ACO

In the existing ACO, a different technique has been used to calculate the new fitness. If $P_k < P_0$, Ant K searches in its current area locally and calculate its new fitness J_k otherwise Ant k searches globally. The Time complexity is depended on max loop times N_{max} and the amount of ants m , so that Time complexity was written as $O(N_{max} gm)$. The Space Complexity is more as every time search will happen either locally or globally depending on the condition. Estimation Accuracy was also little higher compared to enhanced ACO.

5.3. Observations

In the above figures (Fig. 2-6), it is shown the estimation accuracy of Ant Colony Optimization and Enhanced Ant Colony Optimization. The Performance of the enhanced ACO is dependent on the size of the solution space and the parameter α . If the solution space involves into two variables whose values $0 < a < N_a$, $0 < b < N_b$, then the solution space can be written as $N = N_a * N_b$.

Time Complexity can be written as follows:

$$T = N (1 / \alpha) = N / \alpha \tag{2}$$

Space Complexity can be written as follows:

$$O (m + (m - \alpha)) \tag{3}$$

where, m is the number of ants, α is the number of intervals.

Table 1. Model form of software reliability growth models

Type	SRM	$\mu(t)$
Finite failure poisson	G-O Model	$a(1-e^{-bt})$
Finite failure poisson	Delay S Shaped model	$a(1-(1+bt)e^{-bt})$
Finite failure binomial	Weibull model	$a(1-(e^{-\beta t})^\alpha)$
Infinite failure poisson	M-O model	$a \ln(1+bt)$
Binomial	JM model	$N(1-\exp(-\varphi t))$
Infinite failure	Dunane model	$A t^\beta$

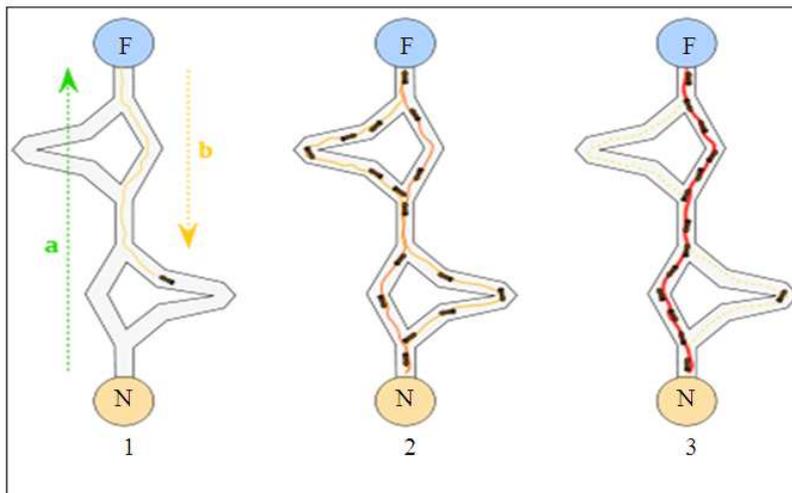


Fig. 1. Basic ant colony optimization

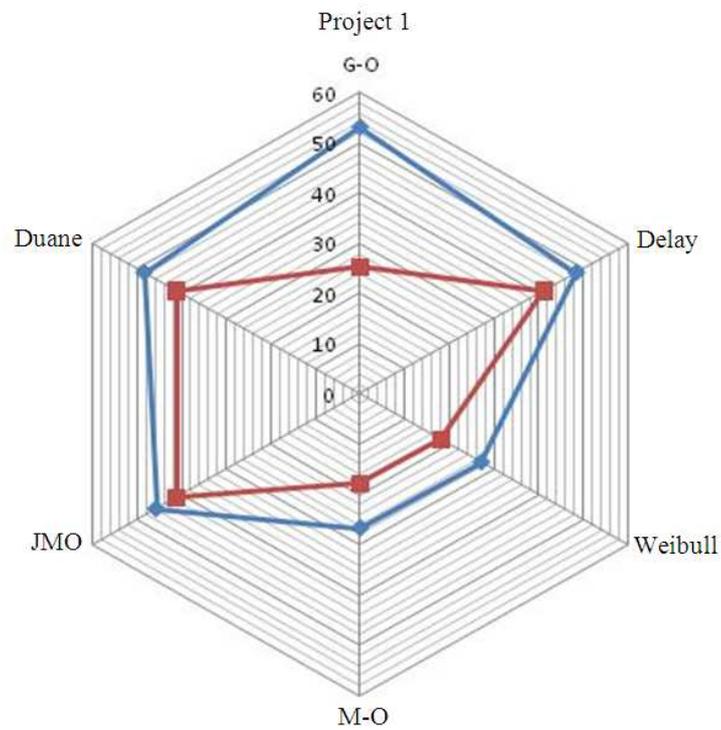


Fig. 2. Estimation accuracy of ACO and enhanced ACO for project 1

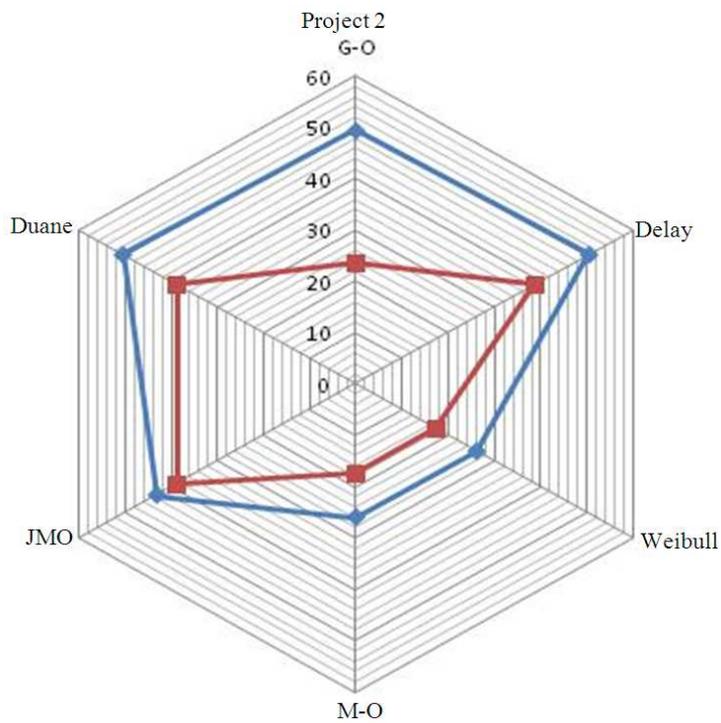


Fig. 3. Estimation Accuracy of ACO and Enhanced ACO for Project 2

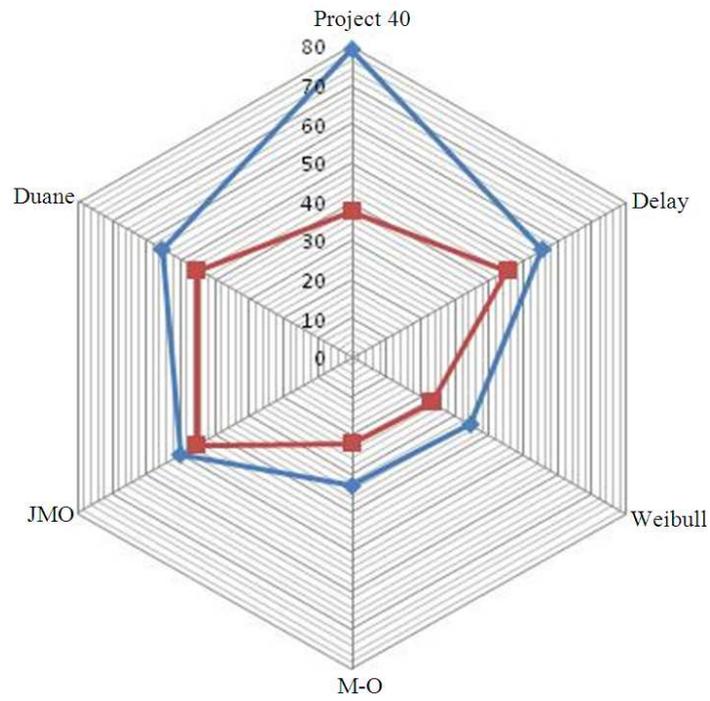


Fig. 4. Estimation Accuracy of AC0 and Enhanced AC0 for Project 40

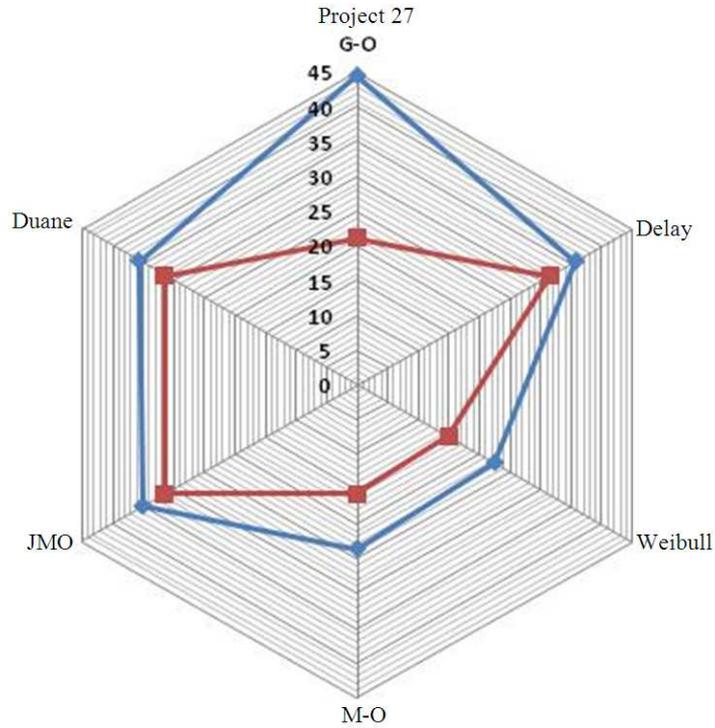


Fig. 5. Estimation accuracy of AC0 and enhanced AC0 for project 27

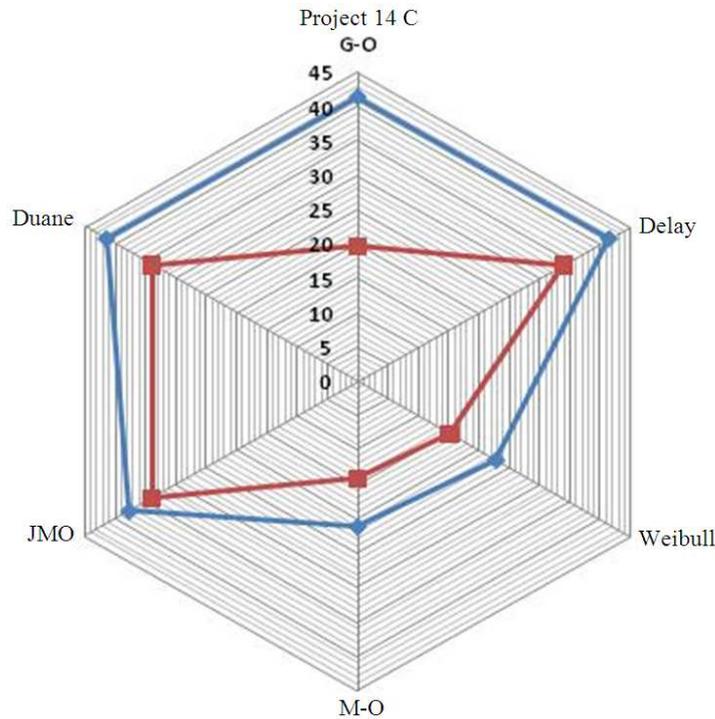


Fig. 6. Estimation accuracy of ACO and enhanced ACO for project 14C

5.4. Advantages of Proposed Method

The Enhanced ACO can be used for any kind of Software Reliability Models. It is comparatively giving better Estimation Accuracy (approximately 10%) than the existing Ant Colony Optimization. The results of Enhanced ACO are much lower than those by the existing Ant Colony Optimization. Getting new fitness is very easy as time interval is introduced and the procedures are based on the principle of divide and conquer method.

6. CONCLUSION

An important aspect in reliability engineering is estimating parameters of Software Reliability Models. Many of the traditional parameter estimation methods are complex to estimate the accuracy because of their nonlinear nature. In the current research an Enhanced Ant Colony Optimization Method was introduced to calculate Parameter Estimation accuracy. Based on the results, it was found that the Enhanced Ant Colony Optimization Method is giving better estimation accuracy than Existing ACO method. Time and Space Complexity is also reduced.

7. REFERENCES

- Aljahdali, S., 2011. Development of software reliability growth models for industrial applications using fuzzy logic. *J. Comput. Sci.*, 7: 1574-1580. DOI: 10.3844/jcssp.2011.1574.1580
- Chen, Y. and N.C. Beaulieu, 2010. A novel approximation of NDA ML estimation for UWB channels. *IEEE Trans. Commun.*, 58: 2795-2798. DOI: 10.1109/TCOMM.2010.082710.080097
- Goseva-Popstojanova, K. and K.S. Trivedi, 2000. Failure correlation in software reliability models. *IEEE Trans. Reliab.*, 49: 37-48. DOI: 10.1109/24.855535
- Hirose, H., 2011. Parameter estimation for the truncated weibull model using the ordinary differential equation. *Proceedings of the 1st ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, May 23-25, IEEE Xplore Press, Jeju Island, pp: 396-399. DOI: 10.1109/CNSI.2011.63
- Ishii, T., T. Dohi and H. Okamura, 2012. Software reliability prediction based on least squares estimation. *Q. Technol. Quantitative Manage.*, 9: 243-264.

- Jiang, R., 2009. Required characteristics for software reliability growth models. Proceedings of the WRI World Congress on Software Engineering, May 19-21, IEEE Xplore Press, Xiamen, pp: 228-232. DOI: 10.1109/WCSE.2009.157
- Li, W., Q. Yin and X. Zhang, 2010. Continuous quantum ant colony optimization and its application to optimization and analysis of induction motor structure. Proceedings of the IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications, Sep. 23-26, IEEE Xplore Press, Changsha, pp: 313-317. DOI: 10.1109/BICTA.2010.5645311
- Liang M., Y. Zhuang and C. Cai, 2010. Face classification using sparse reconstruction embedding. Proceedings of International Conference on Computational Intelligence and Software Engineering, Dec. 10-12, IEEE Xplore Press, Wuhan, pp: 1-4. DOI: 10.1109/CISE.2010.5677037
- Saravanan, V., S. Thirukumaran, M. Anitha and S. Shanthana, 2013. Enabling self auditing for mobile clients in cloud computing. Int. J. Adv. Comput. Technol.
- Shaik, R. and S. Akthar, Rafi, S.K. and S. Akthar, 2011. Discrete Software Reliability Growth Models with discrete test effort functions. Int. J. Soft. Eng. Applic., 2: 11-21.
- Shanmugam, L. and L. Florence, 2012. A comparison of parameter best estimation method for software reliability models. IJSEA, 3: 91-102. DOI: 10.5121/ijsea.2012.3508
- Zeljko, V., N. Radovanovic and D. Ilic, 2011. Software reliability: Models and parameters estimation. Scientif. Techn. Rev., 61: 57-60.