A COMPARISON OF PARAMETER BEST ESTIMATION METHOD FOR SOFTWARE RELIABILITY MODELS

Latha Shanmugam[1] and Dr. Lilly Florence[2]

[1]Research Scholar, Anna University, Tamil Nadu
lathashanmugam4@gmail.com
[2]Professor, Adiyamaan College of Engineering, Hosur.
lilly_swamy@yahoo.com

ABSTRACT

During the past few Decades, many software reliability growth models have been suggested for estimating reliability of software as software reliability growth models. The Functions suggested were non-linear in nature, so it was difficult to estimate the proper parameters. An Estimation method based on Ant Colony Algorithm in which parameters are estimated is discussed in this paper. In this paper, Numerical examples which have been based on five sets of real failure data have been discussed Using existing methods viable solutions for some of the models and data sets cannot be obtained, where as in the proposed method, at least one solution can be obtained. The accuracy of the results using proposed method when compared with PSO algorithm has higher accuracy for at least 10 times for majority of the models

KEYWORDS

Software Reliability Growth Model, Estimation, Particle Swam Optimization, Ant Colony Algorithm

1. INTRODUCTION

Software Development organizations have a challenging task of meeting two requirements simultaneously. The first one being able to predict and meet Business Growth opportunities and the second one is being providing software with minimum fault percentage. For Quantitative control of software testing process and also to measure the reliability of the software, SRGMs are used. Many models have been developed in the past by estimating initial fault number and their effect on software operations and also to predict software reliability.

Software Reliability Model is categorized into two, one is static model and the other one is dynamic model. Dynamic models observe the temporary behavior of debugging process during testing phase. In Static Models, modeling and analysis of program logic is done on the same code. A Model which describes about error detection in software Reliability is called Software Reliability Growth Model.

We assume that the software system is subject to failure randomly due to software errors. Whenever, there is a software failure, it is removed and assumed, that new errors are not introduced. A growth curve can be made between time span of software testing and cumulative number of errors detected. Two types of curves can be made. 1) Exponential curve 2) S-Shaped curve. Failure is defined as an unacceptable departure of program operation caused by a software error remaining in the system [9]

DOI : 10.5121/ijsea.2012.3508

Modified Gompertz model had given an approximation, in a complex software system based on failure data which have been collected during maintenance phase that matched the real data. In this way, it is possible to supply valuable system to both the customer and managing the software team[2].

2. EXISTING ESTIMATION METHODS

Parameter Estimation is a method by Conversing the Model Base Prediction; it takes sample data as given and estimate the most likely model fitting that data set. It constructs the model and constraints based on domain knowledge also it resolves to find most likely values for parameters of the model.

2.1 Maximum Likelihood Estimation (MLE):

In statistics, **MLE** is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model's parameters. The method of maximum likelihood corresponds to many well-known estimation methods in statistics. MLE would accomplish the estimates by taking the mean and variance as parameters and finding particular parametric values that make the observed results the most probable

Drawbacks of MLE

- With small numbers of failures MLE's can be heavily biased and the large sample optimality properties do not apply
- Calculating MLE's often requires specialized software for solving complex non-linear equations. This is less of a problem as time goes by, as more statistical packages are upgrading to contain MLE analysis capability every year.

2.2 Least Square Estimation Method (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 [3].

3. RELATED WORK

As most of the Software Reliability Growth Models are nonlinear in nature, the basic two methods (Maximum Likelihood & Least Square) are not suitable. Researchers found many new methods for parameter estimations. Okamura and Dohi [4] introduced a method by using Expectation-Maximization (EM) principal and he applied the same in Hybrid, Discrete time and Markov Modulated Software Reliability Models. Minohara and Tohma [5] introduced a Genetic Algorithm for Hyper-Geometric Distribution Software Reliability Growth Models and it was observed that length of coding string has been reduced by proportion coefficient; hence the parameter's searching range was decreased. Zhang [7] applied Particle Swarm Optimization (PSO) algorithm, for the estimation, but it was observed that the searching range is too large and the convergence speed is slow and accuracy is not high. Ritika Wason [1] proposed new paradigm for estimation by novel Finite Automata based reliability Model that implicitly scores

over the traditional models on many factors, most importantly due to the fact that is based on the realistic assumption that a software system in execution is a Finite State Machine.

4. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions (particles) and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position and is also guided towards the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm towards the best solutions.

4.1 Drawbacks of PSO

It is a method that easily suffers from the partial optimism, which causes less extract at the reputation of its speed and direction. Then the method cannot work out the problem of scattering and optimization. Also PSO cannot work without a problem such as the solution to the energy field and the moving rules of the particles in energy field in a non coordinate system [12]

4.2 Research issues in PSO

Matching algorithms to Problems (or algorithmic components), Application to different kind of problems. (Dynamic, stochastic, combinational), Parameter Selection (How many particles, which topology), Identification of "state of the art" PSO algorithms (Comparisons), New Variants (Modifications hybridizations) Theoretical aspects (Particle behavior, stagnation), an estimation of distribution particle swam optimization algorithm [13]

5. SOFTWARE RELIABILITY GROWTH MODEL - REQUIRED CHARACTERISTICS

Suppose that a software system is tested and a software failure that is caused by a software fault is still remaining in the system. It is assumed that a software failure occurs at random time and the software fault caused by the failure is immediately removed without introducing new faults. This is the basic concept of Software Reliability Growth Models. The Required Software Reliability Growth Model's Characteristics are as follows.

- SRGM can be viewed as a product of a cumulative density function and a positive constant.
  H (t) =a [1-exp (-$\int_0^1 d(x)dx$)] = $aG(t)$
  Here H (t) is mean value function.
  G (t) is Cumulative Density Function.
- The fault detection rate must be finite; G (t) must meet the condition that the corresponding failure rate function is finite.
  h (t) =d (t) [a-H (t)]==ag(t)
  Where g (t) =dG (t)/dt is the probability density function (pdf) associated with G (t).
  d (t) =ag (t) $\frac{ag(t)}{a-aG(t)} = \frac{g(t)}{1-G(t)}$ = r (t) -------- (1)
  Where r (t) is the failure rate function associated with G (t). h (t) is intensity function. Since the fault detection rate must be finite, G (t) must meet the condition that the corresponding failure rate function is finite.

- We need to represent appropriately the right tail of g (t) behavior to achieve a good reliability prediction. The right tail of SRGM associated with g (t) should be heavy. All these Characteristics should be satisfied for the Software Reliability Model [14]

5.1 Factors affecting the performance of Software Reliability Growth Models [8]

Unsolved problematic issues: The software behavior cannot be predicted as it is not straight forward; we come across Reliability Models which hamper the performance to a great extent.

Unfounded types of assumptions: The Assumptions which have been proposed are not justified either theoretically or practically.

Complexity of the software: The software systems have complex structure and very difficult to identify. The Existing reliability formulates a model to predict the future reliability.

Complexity of the reliability models: These are many models existing but cannot satisfy the requirements of Reliability Prediction. The models are complex in nature and cannot be simplified, so it is difficult to implement them in software domain for improving reliability of software.

Weakness of the reliability model: In all the existing reliability models, the functionality is based on Probabilities. This cannot be recommended for software reliability modeling methodologies. All the models are also based on black box testing which do not consider internal structure of the software and also the models are not able to eliminate all the errors in software.

The Misconception of fault and failure phenomena: The assumption that failure rate is directly proportional to number of faults in program is considered unrealistic. As software has different types of faults and each one should be treated differently to remove the errors.

Inaccurate Modeling Parameters Most of the models use parameters which are not even justified. In reality, there are many uncertainties surrounding those parameters and they can rarely be estimated accurately. Lack of enough experimental data has been considered as a stumbling for the success of the reliability models

Difficulty in selecting the reliability models: The parameters used can never be estimated correctly therefore the experiment data is not sufficient to design a successful model selection. As a result, there are no universally accepted methodologies to select the reliability models that correspond correctly to the software environment which is undergoing reliability measurements.

6. ABOUT ANT COLONY ALGORITHM

The basic idea of software reliability modeling is to predict the reliability of the software with its failure data. During the past 40 years, nearly one hundred software reliability growth models (SRGM) have been proposed. Two traditional methods in parameter estimation are maximum likelihood and least squares method. Ant colony algorithm [10] is a brand-new bionic simulated evolutionary algorithm, which has been applied to many fields. It has a lot of virtues. First, it uses a positive feedback, parallel and self-catalyze mechanism. Thus it has an excellent robustness and is easy to collaborate with other methods. Second, it has a well performance to the optimization problem, and can get rid of traditional optimization methods' weakness. Third, coding and operation with ant colony algorithm is easy, and it has a good convergence rate. Therefore, this

algorithm is suitable for the parameters estimation of software reliability models. Ant Colony Optimization method works in the Least Square's principle.

6.1 Applications of ACO

Ant Colony Optimization is mainly used for goodness of fit. It can be used in many fields, like scheduling, Travelling Sales man's problem, Telecommunication Networks and Vehicle Routing Problem.

Scheduling: It is a widespread problem of practical importance. An online example can be found at the University of Leeds Computer Science department. This is a bus driver scheduling application written by Paul Forsyth & Anthony Wren based on ACO.

Telecommunication Networks: Network *routing* refers to the activity of creating, maintaining and using routing tables (one for each node in the network) to determine where to direct an incoming data stream so that it can continue its travel through the network. In telecommunications this is an extremely difficult problem because of the constant changes in network traffic load. This is where the adaptive advantages of the Ant Colony methodology can help. A good example is available as a technical report by G. DiCaro and Marco Dorigo from IRIDA, University Libre de Bruxelles.

Vehicle Routing Problem: This obviously bears some resemblance with the Travelling Salesman Problem discussed in the previous section. Here we have to take more considerations into account. For instance in the example introduced here, there is an issue of vehicle carrying capacity. Bernd Mullenheimer, Richard Hartl and Christine Strauss have created an Ant Algorithm for a vehicle routing problem.

7. PARAMETER ESTIMATION METHOD BASED ON ANT COLONY OPTIMIZATION ALGORITHM

It was discovered that while ants are searching for food, they secret pheromone which will be a guide for other ants to follow the route. This pheromone is volatile in nature, so the ants will follow the route as long as signal is strong. If no ant has chosen a path and a path has been selected long time back, then ants will find difficulty in searching for the food. The Ant Colony algorithm idea is used based on the above fact. It will allow us to find optimal solutions from a set of candidate solutions. Basic Ant Colony Algorithm discussion can be found in [10]. 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 this paper, we improve the algorithm based on the characteristics of SRGMs [6]. The algorithm is explained as follows:

Step 1: For solving the Parameter Estimation problem, it is converted into Optimization Problem with the help of Least Square's principal. The resulting function is as (1):

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

The Euclidean distance between the actual failure number and the predicted failure number is represented as J in the Function. J (Fitness).

The Time t gives the failure occurrences; the total time, the software runs is represented by T .

The total discovered failure number in time t, is m(t), the predicted failure number by SRGM are expected failure number by time its $\mu(t)$.

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

This value domain is divided into two areas n1 & n2. Then all the parameter's value domain is divided into n1 x n2 areas. Let number of ants living in these areas is taken as m randomly.

Step 3: As long as searched time is less than N_{max} (Maximum repetition times) each ant automatically can search for best path.

The Travel threshold = P0. The pheromone $\tau_k = J_k$ *for* any ant k

The probability of choosing local searching or global searching by the ant k is P_k

$$P_k = r_k / \tau_k \quad (2)$$

Where rk = $\tau_k - \tau_{best}$

τ_k - Current path of ant k

τ_{best} - Current best path of ant k

If $P_k < P0$ 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, then the ant's position is updated else not. To update τ_k, the following formula is used

$$\tau_k = (1 - \rho)\, \tau_k + J_k \quad (3)$$

ρ - Rate of volatilization rate of pheromone. The following algorithm gives the pseudo code of the algorithm.

Algorithm 1: Parameters estimation method based on ant colony algorithm

Input: $\mu(t)$, $m(t)$:
Output: fitting result
Comment: $\mu(t)$:software reliability growth model
$m(t)$: actual failure number
begin
Set the initial value of P0, ρ, *Nmax* and *m*;
Divide the solution space into n_1 x n_2 areas and Put the ants in these areas randomly;
Calculate initial fitness $\tau_k = J_k$ for each ant *k* as (1)
$N \leftarrow 0$;
while $N < N_{max}$
$k \leftarrow 0$;
while $k < m$
Calculate travel probability p_k for each ant *k* as (2);
if $p_k < P0$
else
Ant k searches in its current, area locally;
Calculate its new fitness J_k as (1);

else
Ant *k* searches globally;
Calculate its new fitness J^1_k as (1);
end if
if $J'k < J_k$
Update the ant's position;
else
The ant holds still;
end if

Calculate the ant's pheromone τ_k as (3);
$K \leftarrow k+1$;
end while
Record the best value and the ants' position;
$N \leftarrow N+1$
end while
Output the position of the ant which has the best value, namely the fit result;
end

8. EXPERIMENT EVALUATION

In this paper, six classic Software Reliability Models were chosen to calculate estimation accuracy. Out of which four models were compared with PSO algorithm.

G-O Model: This is a continuous time –independent and identical error behavior model. This is extension of J-M Model with the possibility of imperfect debugging. It represents the number of errors X (t) in the software at time t by a Markov Process whose transition probabilities are governed by the proud of imperfect debugging. Time between the transactions of x (t) is taken to be exponentially distributed with rates dependent on the current fault content of the system. The Prediction of the model can be given as **μ (t) =** a $(1\text{-}e^{-bt})$

Delay S Shape Model will be illustrative of the gamma distribution class. Here the per fault failure distribution is gamma. The number of failures per time period, however, is a Poisson type model using the classification scheme of Musa and Okumoto rather than considered as Binomial. The Prediction of Model form **μ (t) =** a $(1\text{-}(1+bt)\ e^{-bt}$

Weibull Model: one of the most widely used models for hardware reliability modeling is the Weibull distribution. It can accommodate increasing, decreasing, or constant failure rates because of the great flexibility to the infinite failures category and is of binomial type using the Musa Classification. Failure Prediction can be determined by the model form **μ (t) =** a $(1\text{-}(e\text{-}^{\beta t})\ ^{\alpha})$

M-O Model: This is continuous time-independently distributed inter failure times model. Moranda modified J-M Model by introducing a geometrically decreasing hazard function, in that he considered that only one fatal error is removed during each debugging interval. Faults are not removed until the occurrence of a fatal one at which time the accumulated group of faults is removed. The hazard function after a restart is a fraction of the rate which was attained when the system crashed. The Prediction Model form is given as μ **(t) = a** ln (l+bt).

Jelinski-Moranda Model: This is a continuous time-independently distributed inter failure times and independent and identical error behavior model. This model makes the following assumption. There are N faults in the software at the start of testing. Fault is independent of the other, each fault is removed with certainty in a negligible time, No faults are introduced during the debugging, the software failure rate of hazard function at any time is proportional to the current fault content of the program. The model form is $\mu(t) = N(1-\exp(-\varphi t))$.

Dunane Model: The Duane model basically is a graphical approach to perform analysis of reliability growth data and is simple and straightforward to understand. The two important benefits of this approach are the straight line used by the Duane plot can be fitted by eye to the data points, Various facts can be depicted by the Duane plot which otherwise could be hidden. By statistical analysis. For example, even though the application of a goodness-of-fit test may conclude the rejections of a certain reliability growth model, it will not provide any possible reasons for the rejection. On the other hand, a plot of the same data might provide some possible reasons for the problem. In contrast, the drawbacks of this model are the reliability parameters cannot be estimated as well in comparison to a statistical model and no interval estimates can be calculated. The Prediction Model form is given as $\mu(t) = \alpha t^{\beta}$

The Software Reliability Models can be viewed in [11].

Table 1. Model Form of Software Reliability Growth Models

Type	SRM	μ(t)	Parameters
Finite Failure Poisson	G-O Model	$a(1-e^{-bt})$	M1
Finite Failure Poisson	Delay S Shaped Model	$a(1-(1+bt)e^{-bt}$	M2
Finite Failure Binomial	Weibull Model	$a(1-(e^{-\beta t})^{\alpha})$	M3
Infinite Failure Poisson	M-O Model	a ln (l+bt)	M4
Binomial	JM Model	$N(1-\exp(-\varphi t)$	M5
Infinite Failure	Dunane Model	αt^{β}	M6

In this M1, M2, M4, M5 & M6 has 2 Parameters. M3 has three parameters. So we believe all these six models can represent most SRGMs, we have taken three models to compare with PSO Algorithm. The Experimental data has been selected from Musa Data set for SYS1, SYS2, and SYS3. Fitness *J* represents the distance between the predicted failure number and the actual failure number. The smaller *J* is, the better the predicted result is shown in the figure. It is shown clearly that the fitting results by proposed algorithm are more closely to the actual data, but PSO algorithm doesn't suit well. Comparing with PSO algorithm, most results by the proposed algorithm are much better. Nearly all the results by ANTs are much less than those by PSO. Hence Goodness of Fit can be determined by Ant Colony algorithm. Musa Data sets have been used to compare the estimation accuracy from Data Analysis centre for Software's Reliability Data set.

Table 2. Parameter Estimation Accuracy with PSO and ANT for System 1

Model	G – O	Delay S Shape	Weibull	M-O
PSO	168.567	142.394	139.724	35.744
ANT	68.871	146.591	33.439	32.922

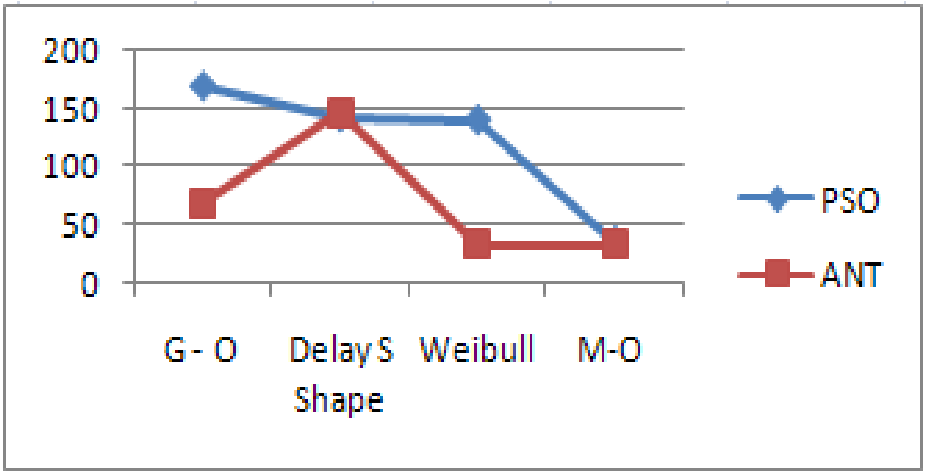

Figure 1. Graph Depicting the Estimation Accuracy with PSO for System – 1

The Ant Colony Algorithm method has been applied for the four software Reliability Models. The result shows that G-O model and Weibull Model has tremendous fitness and the Delay S Shape and M-O Model has little better fitness among PSO method and Ant Colony algorithm. This experiment has been applied for System 1 Failure Data.

Table 3. Parameter Estimation Accuracy with PSO and ANT for System 2

Model	G – O	Delay S Shape	Weibull	M-O
PSO	100.932	230.212	166.654	10.454
ANT	14.491	46.895	38.712	17.418

The results shown that G-O model, Delay S Shape and Weibull Model has tremendous fitness and the M-O Model has little better fitness among PSO method and Ant Colony algorithm. This experiment has been applied for System 2 Failure Data.

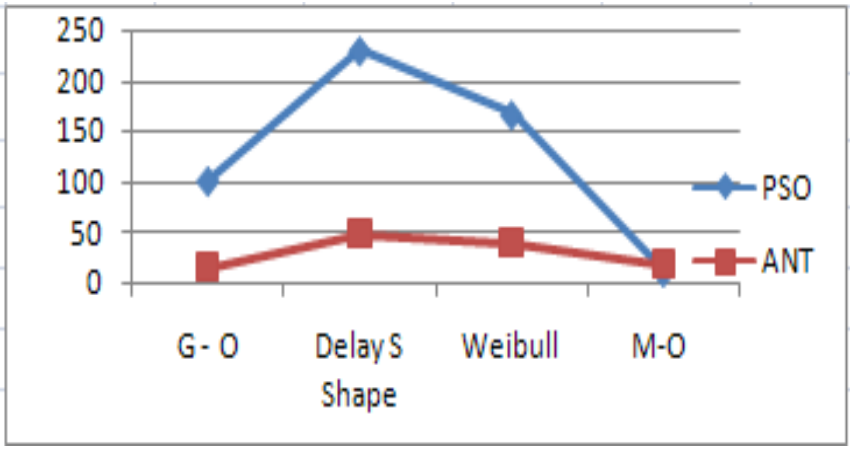

Figure 2. Graph Depicting the Estimation Accuracy with PSO for System – 2

Table 4. Parameter Estimation Accuracy with PSO and ANT for System 3

Model	G – O	Delay S Shape	Weibull	M-O
PSO	456.39	200.335	676.78	48.492
ANT	47.975	154.375	82.607	47.522

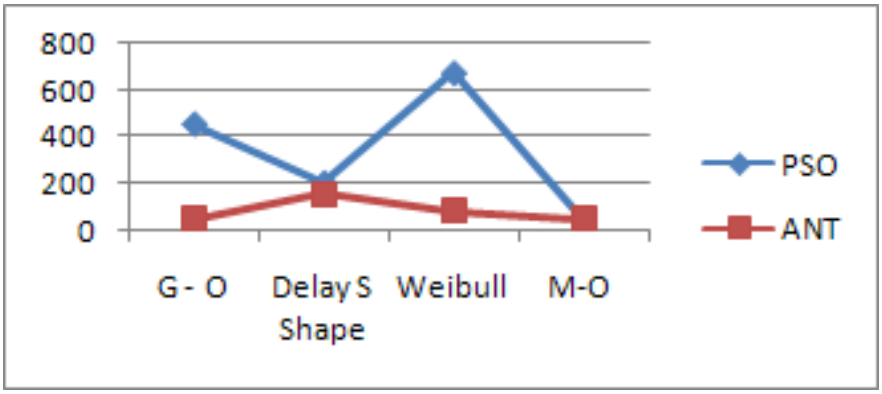

Figure 3. Graph Depicting the Estimation Accuracy with PSO for System – 3

8.1 Comparative Analysis for ACO and PSO

How far the best approximation by using the above algorithms to estimate accuracy is obtained as follows. As per Reliability Estimation Theory, the smaller value of Prediction has more fitness compare to the larger value. [6]. In System 1 Reliability Estimation, G-O, Weibull & Delay Shape have remarkable improvement in the fitness. In System 2 G-O, Delay S-Shape & Weibull Model has remarkable improvements in the fitness. In System 3 all the three models have remarkable improvements in the fitness and marginal improvement in M-O Model.

8.2 Estimation Accuracy with Software Reliability Growth Models.

As per the result of Experiment Evaluation, the four software reliability models were compared with

Particle Swam Optimization method and Ant Colony Optimization method. All the Models are satisfied with characteristics of Software Reliability Growth Model, In General, Our Experiment gives the better results for all the four models, we hope the other two models (Jelinski - Moranda Model and Dunane Model) also can give the better results as given in the Table 5 by Ant Colony Optimization method.

Table 5. Parameter Estimation method – ACO Method

Musa Data	Algorithm	G-O	Delay S Shape	Weibull	M-O	JMO	Dunane
Project 1	Ant Colony	76.7510	72.1505	32.7460	36.9970	49.6214	72.1505
Project 2		60.0371	52.8854	22.8850	26.0385	36.7179	52.8854
SS1		71.5489	57.5800	33.9657	36.1891	53.8793	57.5800
SS2		71.4015	53.2234	37.2993	33.1728	39.1817	53.2234
SS3		33.2301	35.0850	23.4493	23.7345	35.0850	35.0800

The Failure Data have been taken from Musa Data set from the Data Analysis Centre for Software. Reliability Estimation accuracy has been calculated for other real time data, namely Project1,Poject 2, SS1, SS2, SS3 from the DACS Reliability Data Set.

9. CONCLUSIONS

An important aspect in reliability engineering is estimating parameters of Software Reliability Growth Model. Many of these parameters are difficult to estimate because of nonlinear nature of the functions. In this paper, parameter estimation method is suggested based on Ant Colony algorithm. The outcome of the experiment using six typical models demonstrated that this algorithm can be applied for estimating parameters. higher precision and faster convergence speed is achieved through this method when compared with PSO algorithm. In future, work can be carried out in researching in initial value setting of parameters and various methods for dividing the solution space.

ACKNOWLEDGEMENTS

The authors would like to thank the Management, Principal Dr. K.S.Badarinarayan of MVJ College of Engineering, Bangalore for their moral support and guidance for doing research.

REFERENCES

[1] New Paradigm for Software Reliability Estimation by Ritika Wason, P. Ahmed, M. Qasim Rafiq in International Journal of Computer Applications (0975-887) vol 44-No14. April 2012.

[2] Software Reliability: Models and Parameters Estimation by Vladimir Zeljikovic, Nela Radovanovic, Dragomir Ilic, Scientific Technical Review, 2011, Vol.61, No.2, pp. 57 -60.

[3] Estimation of parameters of the Gompertz distribution using the least squares method by Jong-Wuu Wu, Wen-Liang Hung, Chi-Hui Tsai in Applied Mathematics and Computation (2004) 133-147, available @ 2003 Elsevier.

[4] T.Ando, H. Okamura, T.Dohi, Estimating Markov Modulated Software Reliability Models via EM Algorithm[C]. Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing. 2006.

[5] T.Minohara and Y.Tohma, Parameter Estimation of Hyper-Geometric Distribution Software Reliability Growth Model by Genetic Algorithms[C], Proceedings of the 6th IEEE International Symposium on Software Reliability Engineering (ISSRE 1995), pp.324-329, Toulouse, France, 1995.
[6] A Parameter Estimation Method for Software Reliability Models, Changuou Zheng, Xiaoming Liu, Song Huang and Yi Yao in Advanced in Control Engineering and Information Science, available SciVerse Science Direct Procedia Engineering 2011.
[7] ZHANG Ke-han, LI Ai-guo, SONG Bao-wei. Estimating Parameters of Software Reliability models using PSO. Computer Engineering and Applications, 2008, 44(11):47-49.
[8] What is Hampering the performance of Software Reliability Models? A Literature review in Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009 Vol 1, IMECS 2009.
[9] Software Reliability Growth Modeling: Models and Applications by Shigeru Yamada and Shunji Osaki. IEEE Transactions on Software Engineering, Vol, SE-11, No.12 December 1985.
[10] A Coloni, M Dorigo and V Maniezzo. Ant system: Optimization by a colony of cooperating agent [J]. IEEE Trans. Systems Man and Cybemetics-Part B: Cybemetics. 1996, 26(1):29-41 [7]
[11] Lyu M R. Handbook of software reliability engineering [M]. New York: Mc Graw-Hill and IEEE Computer Society Press, 1996.
[12] Particle Swam Optimization: Techniques System and challengeous. By DianPalupiRini, Siti Mariyam, Shamsuddin, Siti Sophyyati Yuhaniz. International Journal of Computer Applications (0975-8887) Vol 14- No. 1 Jan 2011.
[13]Macro Dorigo, Luca M. Gambardella Mauro Biratlari, Alcherio Martino, Riceardo Poli and Thomas stutze, editors, LNCS 4150, Ant Colony Optimization and swam intelligence, 5th International workshop, ANTs 2006 pages 72-83, Berlin, Germany 2006, Springer – Verlag.
[14]R. Jiang "Required Characteristics for Software Reliability Growth Models" in World Congress on Software Engineering IEEE 2009. DOI 10.1109/WCSE.2009.157

Authors

Mrs. Latha Shanmugam is currently pursuing Ph.D. in Anna University, Tamil Nadu. She is presently working as a Associate Professor & Head for Department of MCA, MVJ College of Engineering, Bangalore, Karnataka. She has published papers in reputed national and international conferences recently. Her area of interest is Software Engineering. Contact Email: lathashanmugam4@gmail.com

Dr. Lilly Florence is currently working as a Professor, Department of MCA, Adiyamaan College of Engineering, Hosur, Tamil Nadu. Her area of interest is Software Reliability, Neural Networks. Contact Email: lilly_swamy@yahoo.com