International Journal of Computer Applications Technology and Research
Volume 3– Issue 6, 377 - 380, 2014

Parameter Estimation of Software Reliability Growth Models Using Simulated Annealing Method

Chander Diwaker
UIET
Kurukshetra, Haryana
India

Sumit Goyat
UIET
Kurukshetra, Haryana
India

Abstract: The parameter estimation of Goel's Okomotu Model is performed victimisation simulated annealing. The Goel's Okomotu Model is predicated on Exponential model and could be a easy non-homogeneous Poisson method (NHPP) model. Simulated annealing could be a heuristic optimisation technique that provides a method to flee local optima. The information set is optimized using simulated annealing technique. SA could be a random algorithmic program with higher performance than Genetic algorithmic program (GA) that depends on the specification of the neighbourhood structure of a state area and parameter settings for its cooling schedule.

Keywords: SA, NHPP, SRGM, MVF, FIF.

1. INTRODUCTION

Due to increasing dependence and demand for software system, it's necessary to develop and maintain its reliableness. The system-reliability drawback is that the series-parallel redundancy allocation drawback wherever either system dependability is maximized or total system testing cost/effort is reduced. The keen interest of users in software system reliability models has enhanced since the software system element became a very important issue in several Engineering projects [1]. Reliableness is measured over execution time so it a lot of accurately reflects system usage. Reliability isn't time dependent. Failures occur once the logic path that contains a mistake is executed. Reliableness growth is determined as errors are detected and corrected. During this paper we have a tendency to present an approach to estimate the parameters of Goel's Okomotu Model victimisation simulated annealing. The planned approach provide similar results because the ancient estimation approach victimization the most probability technique while not using info from previous projects, except that the planned approach is far easier to use and no numerical technique is needed.

2. SOFTWARE RELIABILITY GROWTH MODELS

Software reliability growth model could be an elementary technique that asses the reliableness of the software system quantitatively. The SRGM need smart performance in terms of predictability. Any software system needed to control smart reliableness should endure intensive testing and debugging. These processes might be expensive and time intense and managers still need correct info associated with however software system reliability grows. SRGMs will estimate the amount of initial faults, the software system reliability, the failure intensity, the mean time-interval between failures, etc. These models facilitate to measure & track the expansion of reliability of the software system as software system is improved [7]. Within the literature survey of SRGM, solely random processes like NHPP, S-shaped, Exponential, etc., are considered that model the entire behaviour of the amount of failures as a function of time.

3. GOEL'S OKOMOTU MODEL

This Model is predicated on Exponential model and could be an easy non-homogeneous Poisson process (NHPP) model. Goel-Okumoto (G-O) model curve is incurvature. This model provides an analytical framework for describing the software system failure development throughout testing.

- It is predicated on following observations:-

-Number of failures per unit testing time decreases exponentially.

-Cumulative variety of failures versus check time follows an exponentially growing curve.

- In this model it's assumed that a softwar package is subject to failures haphazardly times caused by faults present within the system

.

$$P\{N(t)=y\} = \frac{(m(t))^y e^{-m(t)}}{y!}, \quad y = 0,1,2,\ldots\ldots\ldots$$

Where

$$m(t) = a(1- e^{-bt})$$
$$\lambda(t) = m'(t) = abe^{-bt}$$

Here m(t) is that the expected variety of failures determined by time t and λ(t) is that the failure rate.

Information Requirements:-

- The failure counts in every of the testing intervals i. e. fi

- The completion time of every period that the software package is below observations i. e. ti

The mean value value (MVF)

$$\mu(t) = a(1-e^{-bt})$$

The Failure Intensity function (FIF)

$$\lambda(t) = abe^{-bt}$$

Where a is taken because the expected total variety of faults within the software system before testing and b is that the failure detection rate.

4. SIMULATED ANNEALING

Simulated annealing (SA) could be a heuristic improvement model which will been applied to resolve several tough issues within the numerous fields like programming, facility layout, and graph colouring / graph partitioning issues. SA algorithmic program is galvanized from the method of hardening in metal work. SA has its name related to the employment of temperature as a quantity which may be modified based on a cooling schedule used as a tunable algorithmic program parameter . Annealing involves heating and cooling a fabric to {change} its physical properties because of change in its internal structure. Travelling salesman problem is that the best example of this algorithmic program. SA could be a random algorithmic program with higher performance than Genetic algorithmic program (GA) that depends on the specification of the neighbourhood structure of a state area and parameter settings for its cooling schedule. The key algorithmic feature of simulated annealing is that it provides a method to flee native optima by permitting hill-climbing moves i.e. moves that worsen the target operate worth. The value function returns the output f related to a collection of variables. If the output decreases then the new variable set replaces the previous variable set. If the output will increase then the output is accepted

$$R <= e^{[f(p_{old}) - f(p_{new})]/T}$$

$$R <= e^{[f(p_{old}) - f(dp_{old})]/T}$$

Where R could be a uniform random variety and T could be a variable analogous to temperature. The new variable set is rejected. The new variable set if found by taking a random step from the previous variable set

$$P_{new} = dp_{old}$$

The variable d is either uniformly or unremarkably distributed regarding P previous. This management variable sets the step size so at the start of the method the algorithmic program is forced to form massive changes in variable values. At this time the values of T and d decreases by a definite percent and also the algorithmic program repeats. The algorithmic program stops once T_0. The decrease in T is thought because the cooling schedule. If initial temperature is T0 and also the ending temperature in tn then the temperature at step n is given by

$$T_n = f(T_0, T_N, N, n)$$

Where f decrease with time.

5. LITERATURE SURVEY

.Karambir et al. [1] represented that it's harder to measure and improve the reliableness of internet applications as a result of the big system has extremely distributed nature. Hardware faults might be simply expected instead of the software system faults. During this paper the author used the Goel Okumoto SRGM to observe the amount of faults during a fixed time and estimate its reliableness in regard of internet applications. The speed of modification was calculated by executing the check cases for actual defects per day. The Goel-Okumoto model used exponential distribution to predict the amount of faults in internet applications. This work don't predict the reliability of internet applications that could be a limitation to the present proposed work. Praveen Ranjan Srivastava et al. [2] steered that software system testing could be a crucial a part of the software system Development Life Cycle. The amount of faults occurred and glued throughout the testing section might probably improve the standard of a software package by increasing the likelihood of product success within the market. The method of deciding the time of allocation for testing section is a crucial activity among quality assurance. Extending or reducing the testing time was enthusiastic about the errors uncovered within the software system parts that will deeply have an effect on the general project success. Since testing software system incurs significant project value over-testing the project that resut in higher expenditure whereas inadequate testing would go away major bugs undiscovered resulting in risking the project quality. Therefore prioritizing the parts for checking was essential to realize the optimum testing performance within the assigned test time. This paper conferred a check point Analysis primarily based Module Priority approach to work out the optimum time to prevent testing and unleash the software system. Razeef Mohd et al. [3] planned variety of analytical models throughout the past 3 decades for assessing the reliableness of the software. During this paper the author summarize some existing SRGM give a critical assessment of the underlying assumptions and assess the relevance of those models throughout the software system development cycle using an example. Latha Shanmugam et al. [4] planned that several software system reliableness growth models had been steered for estimating reliableness of software system as software system reliableness growth models. The Functions steered were non-linear in nature therefore it had been tough to estimate the correct parameters. During this paper the discussion includes an Estimation technique supported ant Colony algorithmic program during which parameters were estimated. Using existing ways information sets cannot be obtained wherever as within the planned technique at least one answer might be obtained. The accuracy of the results using planned technique in comparison with PSO algorithmic program has higher accuracy for a minimum of ten times for majority of the models. This work cannot support ways for dividing the answer area and setting the initial worth parameters. D. Haritha et al. [5] steered that the usage of software system reliableness growth model place a really vital role in observation progress accurately predicting the amount of faults within the software system throughout each development and testing method. The high complexness of software system is that the major contributory issue of software system reliableness problem. The amount of faults determined in real software system development surroundings. During this paper the author explore the utilization of particle swarm optimization algorithmic program to estimate software system reliableness growth models parameters. This work didn't develop a polynomial structure model to supply a correct result for the higher model within the computer code reliableness prediction method. Pradeep Kumar et al. [6] planned a NHPP primarily based computer code reliableness growth model for three-tier shopper server systems. The conferred model was composed of 3 layers of client-server design associated with presentation logic, business logic and information keep at backend. Presentation layer contains forms or server pages that presents the program for the applying, displays the information,

collects the user inputs and sends the requests to next layer. Business layer, which offer the support services to receive the requests for information from user tier evaluates against business rules passes them to the information tier and incorporates the business rules for the applying. Information layer includes information access logic, information drivers, and query engines used for communication directly with the information store of an information. The limitation of this work is that this algorithmic program didn't have any mechanism once to prevent the testing method and unleash the products to the top user with higher quality inside budget and without any delay. Shih-Wei Lina b et al. [7] steered that Support vector machine (SVM) could be a novel pattern classification technique that's valuable in several applications. Kernel parameter setting within the SVM coaching method beside the feature choice significantly affected classification accuracy. The target of this study was to get the higher parameter values whereas additionally finding a set of options that don't degrade the SVM classification accuracy. This study developed a simulated Annealing (SA) approach for parameter determination and has choice within the SVM termed SA-SVM. To live the planned SA-SVM approach many datasets in UCI machine learning repository are adopted to calculate the classification accuracy rate. The planned approach was compared with grid search that could be a standard technique of performing arts parameter setting and numerous alternative ways. The disadvantage of this analysis is that this cannot be applied on real world issues. Chin-Yu Huang *et al.* [8] steered that Software Reliability Growth Models (SRGM) have been developed to greatly facilitate engineers and managers in tracking and measuring the growth of reliability as software is being improved. However some research work indicates that the delayed S-shaped model may not fit the software failure data well when the testing-effort spent on fault detection is not a constant. Thus in this paper the author first review the logistic testing-effort function that can be used to describe the amount of testing-effort spent on software testing. The author describes how to incorporate the logistic testing-effort function into both exponential- type and S-shaped software reliability models. Results from applying the proposed models to two real data sets are discussed and compared with other traditional SRGM to show that the proposed models can give better predictions and that the logistic testing-effort function is suitable for incorporating directly into both exponential-type and S-shaped software reliability models. Gaurav Aggarwal *et* ***al.*** [9] categorized Software Reliability Model into two types, one is static model and the other one is dynamic model. Dynamic models observed that the temporary behaviour of debugging process during testing phase. In Static Models, modelling and analysis of program logic was performed on the same code. This paper reviewed various existing software reliability models and there failure intensity function and the mean value function. On the basis of this review a model was proposed for the software reliability having different mean value function and failure intensity function. Lilly Florence *et al.* [10] described that the ability to predict the number of faults during development phase and a proper testing process helped 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 were studied and the estimation accuracy was calculated. The Enhanced method showed significant advantages in finding the goodness of fit for software reliability model such as finite and infinite failure Poisson model and binomial models.

6. PROPOSED WORK

In Earlier Researches PSO (Particle Swarm Optimization) and ACO (Ant Colony Optimization) algorithmic program are explored to estimate SRGM parameters. These algorithms were accustomed handle the modeling issues for the facility model, the Delayed s-shaped model and also the Exponential model. The planned Model can estimate the parameters victimization Simulated annealing (SA) algorithmic program. Simulated annealing could be a common native search meta-heuristic accustomed address separate and, to a lesser extent, continuous optimization issues. The Simulated annealing algorithmic program is employed to handle these models and can show potential benefits in finding the matter. Initial answer is assumed as ω. The temperature counter is modified and t_k is meant to be the temperature cooling schedule. Assume an initial temperature T and variety of iterations are performed at every tk. choose a repetition schedule Mk that represent the repetition rate of every issue. This simulated annealing formulation leads to M0 + M1 +…….. + Mk total iterations being executed, wherever k corresponds to the value for at that the stopping criteria is met. Additionally if for all k, then the temperature changes at every iteration.

7. CONCLUSION AND FUTURE SCOPE

The optimized result of the proposed model using simulated annealing are far better than optimizing the software reliability models using other techniques such as PSO, ACO, Neural Network. Using PSO it is hard to balance development time and budget with software reliability.The PSO produce good results but the number of iteration were increased which indirectly decreases the efficiency. The SA will increase the efficiency and reliability of the software. The failure rate will be reduced. In future works this technique would be applicable to various real life domains such as biometrics, VLSI design, Data mining etc.

8. REFERENCES

1. Mr. Karambir, Jyoti Tamak "Use of Software Reliability Growth model to Estimate the Reliability of Web Applications" International Journal of Advanced Research in Computer Science and Software Engineering , ISSN : 2277-128X, June 2013, Volume 3, Issue 6, pp:-53-59.
2. Praveen Ranjan Srivastava, SubrahmanyanSankaran, Pushkar Pandey "Optimal Software Release Policy Approach Using Test Point Analysis and Module Prioritization" MIS Review Vol. 18, No. 2, pp: 19 – 50, March 2013
3. RazeefMohd., MohsinNazir "Software Reliability Growth Models: Overview and Applications" Journal of Emerging Trends in Computing and Information Sciences VOL. 3, NO. 9, pp: 1309 – 1320, SEP 2012
4. LathaShanmugam and Dr. Lilly Florence "A Comparison Of Parameter Best Estimation Method For Software Reliability Models" International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.5, pp:91 - 102, September 2012
5. D. Haritha, T.Vinisha, K.sachin "Detection of reliable software using particle swarm optimization" Vinisha et al, GJCAT, Volume 1 (4), pp: 489-494, 2011
6. Pradeep Kumar and Yogesh Singh "A Software Reliability Growth Model for Three-Tier Client Server System" International Journal of Computer Applications, pp: 9 – 16,2010
7. Shih-Wei Lina,b , Zne-Jung Lee b, Shih-ChiehChenc and Tsung-Yuan Tseng "Parameter determination of support

8. vector machine and feature selection using simulated annealing approach" Applied Soft Computing 8, pp: 1505–1512,2008
9. Chin-Yu Huang, Sy-Yen Kuo, "An Assessment of Testing-Effort Dependent Software Reliability Growth Models" IEEE Transactions On Reliability, ISSN :0018-9529,June 2007, Vol. 56, No. 2, pp:- 198-211
10. Gaurav Aggarwal and Dr. V.K Gupta " Software Reliability Growth Model" International Journal of Advanced Research in Computer Science and Software Engineering, January 2014,Volume 4, Issue 1,pp :- 475-479,
11. Lilly Florence, Latha Shanmugam "Enhancement And Comparison Of Ant Colony Optimization For Software Reliability Models" Journal of Computer Science 9 (9), ISSN: 1549-3636, 2013,pp: 1232-1240.