Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Comparative Study of Cuckoo Search and Simulated Annealing Technique on SRGM Exponential Models

Garima Batra
(Mtech(Software) UIET, KUK)
Haryana, India

Dr. Kulvinder Singh
UIET, Kurukshetra University
Haryana, India

Abstract: ***Software Reliability is a measure which defines that a system will operate error free in a particular environment for a specific length of time. Reliability is measured over execution time so that it more accurately reflects system usage. Software Reliability Growth Models are nonlinear functions which have been proposed by several researchers in the literature. Reliability of software is measured in terms of Mean Time between Failure (MTBF). The exponential models represent the mean failure function and the failure intensity. Goel-Okumotu and Musa Logarithmic execution time model are two of the most prominent and effective NHPP based models. This paper presents a review of the estimation techniques used in SRG models to find the failure rate and expected number of failures before the software is actually programmed. The cuckoo search technique is compared to the most commonly used simulated annealing technique. The cuckoo search is a comparatively newer optimization technique and promises better results than any of its predecessors. We use the cuckoo search for this particular research.***

Index Terms—Cuckoo Search, Software reliability, NHPP, SRG Models

I. INTRODUCTION

Software refers to one or more computer programs and data held in the storage of a computer for some purpose. It is necessary to develop techniques that may assist in evaluation of the quality of particular software based on the defined object-oriented metrics. Software Reliability is a measure which defines that a system will operate error freein a particular environment for a specific length of time. The keen interest of users in Software Reliability models has increased since the software component became an important issue in many Engineering projects [1]. Execution time is an important parameter while measuring reliability since it affects system usage to a great extent. Reliability is not time dependent. Failures occur when the logic path that contains an error is executed. Reliability growth is observed as errors are detected and corrected. Software Reliability Growth Models are nonlinear functions which have been proposed by several researchers in the literature. Reliability of software is measured in terms of Mean Time between Failure (MTBF).

II. EXPONENTIAL MODELS (EXPM)

NHPP (Non Homogenous Poisson's process)- A process in which the no. of events occurring at time 't' are independent of the no. of events at tim e 't-1' is called an NHPP. The various NHPP models for estimating the failure rate of a software system are given below:

These models represent the mean failure function and the failure intensity. The number of detected errors by the current testing-effort expenditures is proportional to the number of remaining errors. The exponential NHPP model is based on the following assumptions. All faults in a program are mutually independent from the failure detection point of view. The number of failure detected at any time is proportional to the current number of faults in a program (this means that the probability of the failure for faults actually occurring i.e., detected is constant) The isolated faults are removed prior to future test occasions. Each time a software failure occurs the software error, which caused it, is immediately removed and no errors are introduced.

- *Goel-Okumotu model* - has these two formulas for finding error rate.

$$m(t) = a(1 - e^{-bt})$$
$$\lambda(t) = abe^{-bt}.$$

Here m(t) denotes expected number of failures predicted by time t,
λ(t) is the failure rate.
In this model, the parameter 'a' reflects the expected number of failures to be observed eventually
The parameter'b' is the fault detection rate per fault.

- *Musa-Okumotu Logarithmic execution time model* -

This model represents a learning process. If the error detection rate with respect to current testing-effort expenditures is proportional to the number of detectable errors in the software and the proportionality increases linearly with each additionalerror removal. In this model [11] the observed number of failures by some time t is assumed to be a NHPP, similar to the Goel-Okumoto model, but with a mean value function which is a function of t.

$$\mu(t) = \frac{1}{\theta}\ln(\lambda_0\theta_t + 1)$$

whereλo and Θt represent the initial failure intensity and the rate of reduction in the normalized failure intensity per failure, respectively.

Table 1: Mean value function and Intensity function of various models [14]

Model Name	**Mean Value Function**	**Intensity Function**
1. Jelinski-Moranda model	$m(t)=n0(1-\exp^{-\Phi t})$	$\lambda(t)=(N-k)\mu$
2. Goel-Okumoto Model	$m(t)=a(1-exp^{-bt})$, $a>0,b>0$	$\lambda(t)=ab*exp^{-bt}$,$a>0,b>0$
3. Generalized Goel NHPP Model	$m(t)=a(1-exp^{-btc})$, $a>0,b>0,c>0$	$\lambda(t)=abctc-1exp^{-btc}$,$a>0,b>0,c>0$
4. Inflected S-Shaped Model	$m(t)=a*(1-exp^{-bt})1+\psi r*exp^{-bt}$, $\psi r=1-rr$, $a>0,b>0,r>0$	$\lambda(t)=abexp^{-bt}1+\beta t\ 1+\beta*exp^{-bt}2$, $a>0,b>0,\beta>0$
5. Logistic Growth Curve Model	$m(t)=a(1+k*exp^{-bt})$, $a>0,b>0,k>0$	$\lambda(t)=abexp^{-bt}1+k*exp^{-bt}2$, $a>0,b>0,k>0$
6.Musa-Okumoto Model	$m(t)=a*ln(1+bt)$, $a>0,b>0$	$\lambda(t)=ab(1+bt)$, $a>0,b>0$

III. OPTIMIZATION TECHNIQUES

This section details the different optimization techniques prevailing and their application criteria. The genetic algorithm, ant colony, particle swarm optimization, simulated annealing etc. are the traditional techniques prevalent. The cuckoo search is a comparatively new optimization technique and promises better results than any of its predecessors. We use the cuckoo search for this particular research.

A. Genetic Algorithm

GA's are heuristic search algorithms designed to simulate processes in natural system. These are adaptive heuristic search algorithms postulated on the evolutionary ideas of natural selection and genetic. The main disadvantage of GA's:- while solving optimum problems with pure continuous variables they are less efficient than the gradient-based algorithms, as indicated by the fact that a lot more iterations are required for convergence. The problem of premature convergence with GA is well known [12]. GA uses the three principle of natural evolution in nature: Reproduction, natural selection and diversity. The difference from current generation to previous generation maintains the diversity [10] by the use of genetic algorithm for allocating the testing resources.

B. Particle Swarm Optimization

PSO is a particle swarm optimization algorithm for global optimization. PSO is similar to continuous genetic algorithm [14].It is highly desirable to get the accurate estimates of cost and effort, but no prototype has proved to be effective at efficiently and reliably predicting software development cost because of the uncertainties, contingencies and imprecision. The disadvantages of PSO is local search ability is very weak in optimizing realistic problems. PSO is an iterative process. The particles exchange information about their discoveries of the places they have visited. After each iteration, in the PSO main processing loop, the current velocity of each particle's is first updated based on the current velocity particle, the particle's local information and global swarm information. Then, each particle's position is updated using the new velocity of particle.

C. Ant Colony Optimization

ACO is used to solve problems of researchers using various meta-heuristic approaches. The ACO is inspired by the technique of food search behaviour of real ants and their ability to choose the optimum paths. It is a population-based search technique for the solution of difficult combinatorial optimization problems. The ACO algorithm is a bionic simulated evolutionary algorithm. ACO has been applied to many optimization problems like protein folding methods, quadratic assignment and in other implementations [10]. In this, initially ants are randomly located and they go for searching food and when come back to colony they leave a path to that food source so that the reaming ants in the colony won't go through random paths, rather they follow the path that was laid down by the first set of ants. Also the result accuracy of Enhanced ACO is dependent on the solution space and the parameter 'a'. Time and space complexity also got reduced in EACO. The limitation of ACO was: - It can solve optimization problems but it is proved fail at the time of convergence.

D. Simulated Annealing Algorithm

Simulated Annealing (SA) is a heuristic optimization model that can been applied to solve many difficult problems in the various fields such as scheduling, facility layout, and graph colouring / graph partitioning problems. SA algorithm is inspired from the process of annealing in metal work.SA has its name associated with the use of temperature as a variable quantity which can be changed based on a reduction in temperature parameter which acts as tunable algorithm parameter. Annealing involves heating and cooling a material to alter its physical properties due to change in its internal structure. Decreasing temperature in the cooling schedule corresponds to narrowing of the random search process in the neighborhood of the current solution. As the temperature parameter is decreased to zero hill-climbing moves occur less

frequently and the solution distribution associated with the inhomogeneous Markov chain that models the behavior of the algorithm converges to a form in which all the probability is concentrated on the set of globally optimal solutions provided that the algorithm is convergent; otherwise the algorithm will converge to a local optimum which may or not be globally optimal.

E. Cuckoo Search Technique

This is a heuristic search technique which takes its idea from nature like many earlier techniques like Particle swarm optimization, ant colony optimization, simulated annealing. [13] Cuckoos are fascinating birds because of their aggressive reproduction strategy. Some species such as the ani and Guira cuckoos lay their eggs in communal nests.
Three implementation rules:

- The cuckoo bird lays one egg at a time and then it puts its egg in a randomly chosen nest.
- The nests with contain the best quality of eggs take these eggs to the next generations.
- There are a fixed number of available host nests, and the probability that a hostbird is able to discover the cuckoo egg laid is probability pa ∈ [0, 1]. In case he gets one, then the host can either reject the egg and throw away or abandon the present nest, and build a new nest.

For simplicity, this last assumption can be approximated by the fraction pa of the n nests are replaced By new nests (with new random solutions). For a maximization problem, the quality or fitness of a solution can simply be proportional to the value of the Objective function. Other forms of fitness can be defined in a similar way to the fitness function in genetic algorithms. For Simplicity, we can use the following simple representations that a solution can be represented by an egg in a nest, and a cuckoo Egg symbolizes a new solution.The algorithm works by using the new solutions and replacing the not so good solutions with potentially better solutions (cuckoos) in the nests. This algorithm can be further extended to the more complicated case where each nest can have multiple Eggs thereby representing a set of solutions. For this present work, we will use the simplest approach where each nest has only a Single egg.

Pseudocode

- begin
- Objective function f(x), x = (x1, ..., xd)^T
- Generate initial population ofn host nests x_i (i = 1, 2, ..., n)
- while (t <MaxGeneration) or (stop criterion)
    - Get a cuckoo randomly by Levy flights
    - evaluate its quality/fitness F_i
    - Choose a nest among n (say, j) randomly
    - if (F_i>F_j),
        - put the new solution in place of j;
    - end.
    - The worse nestsare abandoned with a probability Pa and the new ones are built;
    - The best solutions(or nests with quality solutions) are retained;
    - Rank the solutions and find the current best

  end while
- Post-process results and visualization
- End

IV. RELATED WORK

This section presents the literature of different SRGM techniques and optimization techniques currently in existence. The different researches explain the results of different estimation techniques used on various SRGM models.
Anniprincyet. al in **[2]** suggested that Gompertz testing effort function for software reliability estimation and analysis was done on optimal release time. The author intended to isolate the software error by using Yamada delayed S-shaped model.The number of persons involved in project, the number of test cases and the calendar time when it spanned were few parameters it was based on. The technique was experimented on real datasets. The research presented a SRGMpredicted the reliability by estimating the software error and the failure rate. The author developed a general approach in deriving more general models using Gompertz testing effort function based on simple assumptions, constant with the basic software reliability growth modeling. Incorporating the dynamics of testing time of the software and the testing coverage had allowed the model to be a two dimensional framework. The reliability and cost values were measured and the performances of the proposed methods were evaluated. The results proved that the proposed method of software reliability growth modelingprovided better results when compared to other related works.**GauravAggarwal*et.***al. in **[3]**his paper reviewed various contemporary software reliability models along with their failure intensity and the mean value functions. Based on this review,the author put forward a new model for measuring the software reliability which had different mean value and failure intensity functions. The authorcategorized Software Reliability Model into two types - static model and the dynamic model. Thestatic Models used the same code, to model and analysethe program logic.The temporary behaviour of debugging process was analysed by dynamic models during

testing phase. **Ranaet. al**in **[5]**, compared between the two estimation procedures for their usability and applicability in context of SRGMs. The author also highlighted a couple of practical considerations reliability practitioners had to be aware of when applying SRGMs. These SRGMs could be used after the testing phase forassessing the maturity level of the software by determining the latent faults prediction. Alternatively, these SRGMs could be used during the project to facilitate the decision processin testing resource allocation. There exist a number of SRG models and to apply a given reliability model, the model equations were with defect inflow data. The two most prominent and effective techniques involved in parameter estimation task were maximum likelihood and method of least squares. It was noted in the study that while MLE was the recommended estimator with superior statistical properties, its usability and applicability in all situations was questionable. Further MLE was difficult to apply which limits its use in industry, especially due to lack of tools support.

Karmbir*et. al.* in **[7]** stated that the reliability of web applications was more difficult to measure and improve. Hardware faults could be easily predicted rather than the software faults. The customer required more security and accuracy in web applications. The reliability of web applications was more complex rather than the other software systems. In this paper, the author used the Goel'sOkomotu SRGM for the detection of number of faults in a specified time and estimated its reliability in regard of web applications. The rate of change was calculated by executing the test cases for actual defects per day. The Goel-Okumoto used exponential distribution to predict the number of faults in web applications. The author assessed the software reliability of web applications by using SRGM. This work did not predict the reliability of web applications which was a limitation to this proposed work.**Hai-Feng Li *et. al* *in* [10]**applied GEP(Gene Expression Programming) into non-parametric software reliability modeling, due to its unique characteristics such as genetic encoding method translation process of chromosomes. The proposed GEP-based modeling approach considered some important characteristics of reliability modeling in several main components of GEP i.e. function set, terminal criteria, fitness function, and then obtained the final NPSRM (GEP-NPSRM) by training on failure data.In contrast to PSRMs, which were not very consistent, the non-parametric SRGMs which used machine learning (ML) techniques like artificial neural networks (ANN), support vector machine (SVM) and genetic programming (GP) for reliability modelingpromised better prediction results across various projects. Finally on several real failure data-sets based on time or coverage four case studies were proposed by respectively comparing GEP-NPSRM with several representative PSRMs, NPSRMs based on ANN, SVM and GP in the form of fitting and prediction power which showed that compared with the comparison models the GEP-NPSRM provided a significantly better power of reliability fitting and prediction. In other words the GEP was promising and effective for reliability modeling. It was difficult to model the defect prediction function based on test coverage using this technique.**LathaShanmugam*et. al.*** in **[9]**provided an overview of software reliability models. The author described that Software Reliability was a useful measure in planning and controlling the resources during the development process so that high quality software could be developed. It was also a useful measure for giving the user confidence about software correctness.Planning and controlling the testing resources via Software Reliability Measures could be done by balancing the additional cost of testing and the corresponding improvements in Software Reliability.

Pradeep Kumar *et. al*. in **[15]** proposed software reliability growth model for multi-tier client server systems. The presented NHPP based model comprised of three layers of client-server architecture which were presentation layer, business logic layer and database stored at backend. The forms or server pages which presented the user interface for the application comprised the presentation layer andhelped display the data, collected the user inputs and sent the requests to next layer. Business logic layerprovided the main logic containing functionalityof system to receive the requests for data from user tier. Itincorporated the business rules for the application, evaluated against business rules and passed them to the data tier. Data layer was made up of data access logic, communicateddirectly with the data store of database using SQL engine and database drivers. The model was experimented and validated through standard dataset consisting of software failure data forsome standard projects, released from the organization dataset for software reliability and verified it on a commercial application. The limitation of this work was thatit did not have any mechanism to decide when to stop the testing process and release the products to the customer with higher quality, within budget and timely.

V. PROBLEM FORMULATED

In Earlier Researches Simulated Annealing, ACO (Ant Colony Optimization) and PSO (Particle Swarm Optimization) algorithmswere utilizedforestimating software reliability. Different researches used these algorithms to handle the modeling problems for the Power model, the Delayed S-Shaped model and the Exponential model. The Proposed Model will estimate the parameters using Cuckoo Search algorithm. The CS algorithm, which is a newer technique in comparison to its contemporary algorithms, has proven better results in function optimization, engineering design, neural network training, and other continuous target optimization problems and has solved the knapsack and nurse-scheduling problems. This model will show significant advantages in handling a variety of modeling problems such as EXPM, POWM and DSSM.

VI. PROPOSED METHODOLOGY

In proposed model the optimization of parameters of SRGM exponential models (GoelOkumotu and Musa logarithmic execution time model) using Cuckoo search and simulated annealing techniques, will result in finding the exact effort required to find high productivity of the model. The main virtue of proposed approach is to assess the effectiveness of these two techniques on these models. Simulated annealing is a popular local search meta-heuristic used to address discrete and, to a lesser extent, continuous optimization problems. Here we use the two heuristic algorithms to estimate the parameters of these two models and compare their effectiveness.

VII. CONCLUSION

The optimized result of the model under consideration using simulated annealing like PSO, ACO, Neural Network etc. can be improved by using Cuckoo search which is a newer technique and has produced better results for different optimization works. The PSO technique cannot maintain a balance between software reliability and development time and budget. The PSO is not very efficient in performance due to its large number of iterations although it produces good results. This cuckoo search will automatically refine the failure rate and produce better results than the traditional model. Reliability of proposed model to predict failures is higher than the traditional model.

ACKNOWLEDGEMENT

I owe my special thanks to Dr. Kulvinder Singh for all his guidance and support without which this research was not possible. I also thank my parents and the almighty for their blessings. And also special thanks to all my friends for all the moral support.

REFERENCES

[1] B. AnniPrincy1, Dr.S.Sridhar2, "*Measuring Software Reliability and Release Time Using SRGM Tool*", International Journal Of Scientific Research And Education Volume 2 Issue 5 Pages 785-796 2014 ISSN (e): 2321-7545.

[2] B. Anni Princy1, Dr.S.Sridhar2, "*Prediction Of Software Reliability Using Cobbdouglas Model In Srgm*", Journal of Theoretical and Applied Information Technology 20th April 2014. Vol. 62 No.2. Pp. 355-363.

[3] GauravAggarwal and Dr. V.K Gupta, "*Software Reliability Growth Model*" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 1,pp :- 475-479, January 2014.

[4] M. Z. Rashad, A E Kashk, M A ErDosuky and M M Kamal, "*Genetic Cuckoo Optimization Algorithm (GCOA)*", International Journal of Computer Applications (0975 – 8887) Volume 90 – No.3, March 2014. Pp. 7-12.

[5] RakeshRana, MiroslawStaron, Christian Berger and Jorgen Hansson, "*Comparing between Maximum Likelihood Estimator and Non-Linear Regression estimation procedures for Software Reliability Growth Modelling*", IEEE 2013.

[6] N´estor R. Barraza, Buenos Aires "*Parameter Estimation for the Compound Poisson Software Reliability Model*" International Journal of Software Engineering and Its Applications , V o l. 7,Issue 1, No. 1, pp:- 137-148, Jan, 2013

[7] Mr. Karambir, JyotiTamak "*Use of Software Reliability Growth model to Estimate the Reliability of Web Applications*" International Journal of Advanced Research in Computer Science and Software Engineering , Volume 3, Issue 6, pp:- 53-59,June 2013

[8] LathaShanmugam and Lilly Florence, "*Enhancement And Comparison Of Ant Colony Optimization For Software Reliability Models*" Journal of Computer Science 9 (9): 1232-1240, 2013

[9] 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

[10] Hai-Feng Li, Min-Yan Lu, Min Zeng And Bai-Qiao Huang "*A Non-Parametric Software Reliability Modelling Approach by Using Gene Expression Programming*" Journal Of Information Science And Engineering 28, pp:- 1145-1160, 2012

[11] ChangyouZheng, Xiaoming Liu, Song Huang, Yi Yao "*A Parameter Estimation Method for Software Reliability Models*" pp:- 3477 – 3481, IEEE 2011

[12] Roberto Pietrantuono, Stefano Russo and Kishor S. Trivedi "*Software Reliability and Testing Time Allocation: An Architecture-Based Approach*" Transactions On Software Engineering, IEEE, VOL. 36, NO. 3, pp:- 323-337, MAY/JUNE 2010

[13] Xin She Yang et. al "*Cuckoo search via levy flights*" IEEE 2009.

[14] AlaaSheta "*Parameter Estimation of Software Reliability Growth Models by Particle Swarm Optimization algorithm*" AIML Vol 7 issue 1 Jun 2007.

[15] Amrit L. Goel "*Software reliability models-assumptions, limitations and applicability*" IEEE Vol SE-11 No. 12 Dec 1985.

[16] 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

