

# Assessing software reliability using Modified Genetic Algorithm: Inflection S-shaped model

<sup>1</sup>Dr. R. Satyaprasad, <sup>2</sup>G. Bharathi, <sup>3</sup>Dr. G. Krishna Mohan

<sup>1</sup>Assoc.Professor, Dept. of CSE, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India

<sup>2</sup>Research Scholar, Rayalaseema University, Kurnool, Andhra Pradesh, India.

<sup>3</sup>Professor, K.L University, Vaddeswaram, Guntur, Andhra Pradesh, India.

**Abstract**—In order to assess software reliability, many software reliability growth models (SRGMs) have been proposed in the past 40 years. In principle, two widely used methods for the parameter estimation of SRGMs are the maximum likelihood estimation (MLE) and the least squares estimation (LSE). However, the approach of these two estimations may impose some restrictions on SRGMs as most of the models are non-linear. The existence of derivatives from formulated models or the needs for complex calculation are the drawbacks. In this paper, we propose a modified genetic algorithm (MGA) to assess the reliability of software considering the Time domain software failure data using Inflection S-shaped model which is NonHomogenous Poisson Process (NHPP) based. Numerical experiments based on real software failure data are performed, and the results show that the proposed genetic algorithm is more effective and faster than traditional algorithms.

**Keywords** - *Software reliability, Inflection S-shaped model, Time domain data, Mean Value Function, Modified Genetic Algorithm, NHPP.*

## I. INTRODUCTION

Software reliability assessment is important to evaluate the quality of software system, since it is one of the most important attribute of software. One of the most difficult problems of software industry is to ship a reliable product. Therefore it is necessary to have accurate and fast estimation techniques for verifying software reliability. For Four decades, many Software Reliability Growth Models (SRGMs) have been proposed in estimating reliability growth of software products. SRGMs can be used to depict the behaviour of observed software failures characterized by either times of failures (i.e Time domain data) or by the number of failures at fixed times (i.e Time domain data) [9].

The parameters of SRGMs are generally unknown and have to be estimated based on collected failure data. Two of the most popular estimation techniques are Maximum Likelihood Estimation (MLE) and Least Squares Estimation (LSE) [3,14]. In fact, MLE and LSE involve the property of probability theory and statistical analysis. Thus, this may impose some restrictions on the parameter estimation of SRGMs [1,11] such as the continuity, the unimodality, the existence of derivatives from formulated models, the

complex likelihood function, etc. The method of MLE estimation by solving a set of simultaneous equations and is better in deriving confidence intervals. The method of LSE minimizes the sum of squares of the deviations between what we actually observe and what we expect. Nevertheless, LSE is suitable for fitting data from small to medium sample sizes [18], while MLE is considered to be better statistical estimator for large sample sizes. In particular, when the formulated model of SRGMs is complicated or the sample size of failure data is large, these two estimation techniques may not be effective to find out the optimal solutions and generally require to be solved numerically. Hence, the more effective and applicable approaches for the parameter estimation of SRGMs may be necessary.

The rest of this paper is organized as follows. Section 2 surveys NHPP based SRGMs and in specific Inflection S-shaped Model along with the past researches of GAs in software engineering areas. In Section 3, an effective MGA is proposed to solve the parameter estimation of reliability models. Then, the experimental results based on two failure data are presented and discussed in Section 4. Finally, some conclusions are given in Section 5.

## II. LITERATURE SURVEY

### A. NHPP model

The Non-Homogenous Poisson Process (NHPP) based software reliability growth models (SRGMs) are proved to be quite successful in practical software reliability engineering [13]. The main issue in the NHPP model is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain time point. Model parameters can be estimated by using Modified Genetic Algorithm (MGA). Various NHPP SRGMs have been built upon various assumptions. Many of the SRGMs assume that each time a failure occurs, the fault that caused it can be immediately removed and no new faults are introduced. Which is usually called perfect debugging. Imperfect debugging models have proposed a relaxation of the above assumption [15].

A fault is a statement in a program which causes one or more failures. Software Reliability Growth is defined by the mathematical relationship that exists between the time span of testing a program and the cumulative number of errors discovered. After failure detection, we find a fault and define a fix for the fault. The exponential software reliability growth models are designed to describe the failure detection process.

Let  $\{N(t), t \geq 0\}$  be the cumulative number of software failures by time 't'.  $m(t)$  is the mean value function, representing the expected number of software failures by time 't'.  $\lambda(t)$  is the failure intensity function, ordinarily decreases as faults are detected and removed. Thus

$$m(t) = a(1 - e^{-bt}) \quad \text{and} \quad \lambda(t) = \frac{dm(t)}{dt} \quad . \quad \text{where 'a'}$$

denotes the initial number of faults contained in a program and 'b' represents the fault detection rate. In software reliability, the initial number of faults and the fault detection rate are always unknown. The maximum likelihood technique can be used to evaluate the unknown parameters. This paper deals with the application of Inflection S-shaped model on application test data collected from literature, which is of Time domain data (i.e ungrouped).

SRGMs are a statistical interpolation of defect detection data by mathematical functions. They have been grouped into two classes of models-Concave and S-shaped. SRGMs treat the software debugging process as a fault counting process [5,8]. A well-known family of SRGMs is the Non-Homogeneous Poisson process (NHPP) family of models. Various SRGMs have been proposed for estimation of reliability growth.

The only way to verify and validate the software is by testing. This involves running the software and checking for unexpected behaviour of the software output [7]. SRGMs are used to estimate the reliability of a software product. In literature, we have several SRGMs developed to monitor the reliability growth during the testing phase of the software development. Software reliability is defined as the probability of failure-free software operation for specified period of time 't' in a specified environment.

$$R(x/t) = e^{-[m(t+x)-m(t)]}$$

### B. Inflection S-shaped model

Software reliability growth models(SRGM's) are useful to assess the reliability for quality management and testing-progress control of software development. They have been grouped into two classes of models concave and S-shaped.

The most important thing about both models is that they have the same asymptotic behavior, i.e., the defect detection rate decreases as the number of defects detected (and repaired) increases, and the total number of defects detected asymptotically approaches a finite value. The inflection S-shaped model was proposed by Ohba in 1984. This model assumes that the fault detection rate increases throughout a test period. The model has a parameter, called the inflection rate, that indicates the ratio of detectable faults to the total number of faults in the target software. True, sustained exponential growth cannot exist in the real world. Eventually all exponential, amplifying processes will uncover underlying stabilizing processes that act as limits to growth. The shift from exponential to asymptotic growth is known as sigmoidal, or S-shaped, growth. A Control Chart Procedure and Sequential Probability Ratio Test for Software Reliability is proposed using Inflection S-Shaped Model [16,17].

Ohba models the dependency of faults by postulating the following assumptions:

- Some of the faults are not detectable before some other faults are removed.
- The detection rate is proportional to the number of detectable faults in the program.
- Failure rate of each detectable fault is constant and identical.
- All faults can be removed.

$$\text{Assuming : } b(t) = \frac{b}{1 + \beta e^{-bt}}$$

This model is characterized by the following mean value function:

$$m(t) = \frac{a}{1 + \beta e^{-bt}} (1 - e^{-bt})$$

where ‘b’ is the failure detection rate, and ‘β’ is the inflection factor. The failure intensity function is given as:

$$\lambda(t) = \frac{abe^{-bt}(1 + \beta)}{(1 + \beta e^{-bt})^2}$$

### III. MODIFIED GENETIC ALGORITHM

Genetic Algorithm (GA) has been popularly used to solve various optimization problems. GA has advantages of easy implementation with large search space and rapid convergence on good quality solutions. It does not impose restrictions on the continuity, the existence of derivatives, and the unimodality of evaluation functions.

In recent years, the Genetic Algorithms (GAs) has gained popularity in solving the optimization problem of scientific fields [4,12]. Because, the parameter estimation can be reformulated as a searching process within the domain of all the feasible solutions [10, 6], it may be attractive to introduce GA into the process of software reliability modeling [2]. Therefore, in this paper we proposed a modified genetic algorithm (MGA) to estimate the parameter of the SRGMs. We will attempt to modify GA’s operators with weighted bit mutation and a rebuilding mechanism to improve the performance and efficiency of estimations. Finally, the applicability of proposed MGA, the result of parameter estimation and the reliability with

Inflection S-shaped model is also demonstrated through real data.

Traditional GA has several steps for searching process:

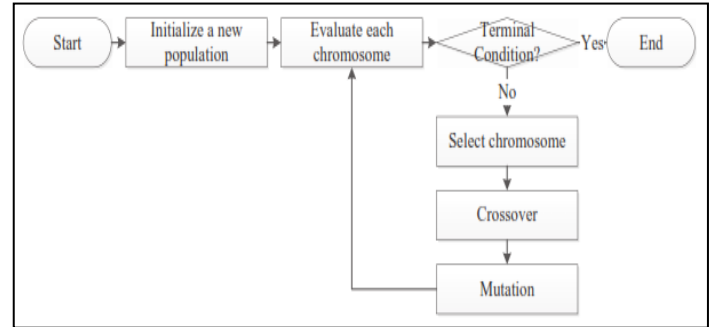
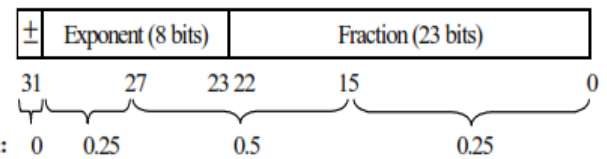


Figure: Genetic Algorithm process

- **chromosome representation**

GA simulates the initial population of parametric solution represented as chromosomes. Each chromosome is encoded as string of bits. Since the parameters of SRGMs are usually real numbers, we proposed an IEEE floating-point standard to encode chromosomes.



Chromosome Representation and Weighted Bit Mutation

- **fitness function;**

- *least squares estimation (LSE)*

$$\text{fitness\_LSE} = \frac{1}{\text{MSE}}$$

Where, MSE is a measure to compare the differences between actual values and estimators.

- **Selection scheme:** The selection scheme is to select the candidate chromosomes from the current population based on their fitness values. Our goal is to maximize fitness function for finding the best parameters. With these fitness values, we can further adopt roulette wheel selection and uniform crossover to choose candidate chromosomes. The roulette wheel selection does not guarantee that the fittest chromosome will always be selected for generating offspring in searching process. This may spend more numbers of generations on finding a solution. Thus, we propose a rebuilding mechanism. Among each generation, one best chromosome is kept at the end of the population to

avoid disappearance from the selection scheme. This mechanism does not violate GA's original purpose. If the next generation produces a much more suitable chromosome, the previous kept chromosome will be replaced.

- **Crossover operator:** Two chromosomes are chosen from the population and are exchanged in part with each other in order to improve their fitness value. The uniform crossover is one of the simplest forms [4]. The crossover may happen at different bits with a probability called crossover rate,  $P$ . This rate typically ranges from 0.5 to 0.8 from GA literatures [6]. It is decided to adopt uniform crossover in our experiments.
- **Mutation operator:** We found that some bits are less efficient during bit mutation based on IEEE floating-point format. If we mutate at sign bit, the whole string will be changed from a plus to a minus. Because the estimated parameters are usually a positive real number, this mutation may be useless. Similarly, if we mutate at a very high exponential bit or at a very low fractional bit, the whole string will respectively be  $2^{\pm 128}$  times the original or only be changed slightly. In fact, these mutations may be too severe or negligible. **Sensitivity analysis** on different bit mutations will be investigated. Depending on this analytic result, we further provide a weighted bit mutation.
- **Stopping criteria:** The searching process will iteratively evolve parametric solutions until the maximal generations equal to 10000 trials or the best fitness function does not change in the past 1000 trials.

#### A. Algorithm for parameter estimation

In this section, we show how to modify the traditional GA to estimate the parameters of SRGMs. The detailed algorithm of MGA is shown below. It is noted that all the proposed mechanisms of MGA are built by using Java programming language.

1. Initialize a population of chromosomes randomly
2. FOR (Iteration  $i=1$ ;  $i \leq$  Maximum generation & termination condition = FALSE;  $i=i+1$ )
  - a. Calculate fitness for all individual chromosomes
  - b. Reproduce offspring by roulette selection

- c. Choose two chromosomes from the population in order and randomize a probability  $p$
  - d. IF  $p <$  Crossover rate THEN
    - i. Generate two offsprings by recombining two chromosomes.
 ENDIF
  - e. Choose a chromosome from the population in order and randomize a probability  $q$
  - f. IF  $q <$  Mutation rate THEN
    - i. mutate the chosen chromosome at a weighted bit position
 ENDIF
  - g. Keep the fittest parent in the end of population
  - h. Check termination condition
3. ENDFOR
  4. Output estimated parameters

### IV. ILLUSTRATING THE MGA

#### A. Data Analysis

There are two common types of failure data: time-domain and Time-domain. Some software reliability models can handle both types of data. The time domain approach involves recording the individual times at which failure occurred. The Time domain approach is characterized by counting the number of failures occurring during a fixed period (e.g., test session, hour, week, day). The collected data is the Time Between Failures. Based on the failure data collected from the literature, we used cumulative failures data for software reliability using Inflection S-shaped model. Assuming  $\beta = 0.05$ , the estimated parameter are as follows.

Fitness function is an indicator of the fitness of every chromosome that shows how close this chromosome is to the desired solution. MGA selects the candidate chromosomes from the current population based on their fitness values. The LSE-based fitness function is given as follows:

$$fitness_{LSE} = \frac{1}{\sum_{i=1}^n (m(t) - m_t)^2}$$

**B. Results**

**TABLE I. Estimated parameters**

Data Set	LSE based fitness	
	a	b
DS1	84.963130	0.039024
DS2	22.000137	0.093427

**TABLE II. Reliability Growth**

Failure Number	Time Between failures	m(t)	R(x/t)
1	5.5	15.775220	0.010092
2	7.33	20.371264	0.001732
3	10.08	26.729496	0.000000
4	80.97	81.185432	0.584259
5	84.91	81.722843	0.238618
6	99.89	83.155735	0.795571
7	103.36	83.384430	0.601630
8	113.32	83.892542	0.681113
9	124.71	84.276569	0.690426
10	144.59	84.647015	0.920318
11	152.4	84.730052	0.903729
12	167	84.831278	0.953727
13	178.41	84.878656	0.956826
14	197.35	84.922790	0.963498
15	262.65	84.959975	0.999995
16	262.69	84.959980	0.996878
17	388.36	84.963107	0.999978
18	471.05	84.963129	1.000000
19	471.51	84.963129	0.999999
20	503.12	84.963130	1.000000
21	632.43	84.963130	1.000000
22	680.03	84.963130	

**TABLE III. Reliability Growth**

Failure Number	Time Between failures	m(t)	R(x/t)
1	0.5	0.958339	0.117618
2	1.7	3.098653	0.014811
3	4.5	7.311025	0.041029
4	7.2	10.504501	0.074641
5	10	13.099564	0.117216
6	13	15.243301	0.355965
7	14.8	16.276225	0.633593
8	15.7	16.732574	0.527631
9	17.1	17.371933	0.277617
10	20.6	18.653445	0.404150

11	24	19.559415	0.772881
12	25.2	19.817046	0.839211
13	26.1	19.992339	0.745454
14	27.8	20.286100	0.811018
15	29.2	20.495565	0.715626
16	31.9	20.830163	0.739507
17	35.1	21.131934	0.834791
18	37.6	21.312508	0.889539
19	39.6	21.429559	0.822275
20	44.1	21.625240	0.900773
21	47.6	21.729742	0.901214
22	52.8	21.833755	0.921774
23	60	21.915210	0.947745
24	70.7	21.968880	

These parameter estimates ‘a’ and ‘b’ represent the initial number of failures and the failure intensity function for the considered failure data sets. For both the data sets R(x/t) is increasing and exhibiting the reliability growth.

**V. CONCLUSION**

A number of estimates of software quality are based on the parameter estimates of SRGMs. Therefore, the quality estimates can be derived based the quality estimates of parameters. With the estimation of SRGMs parameters, project managers can get useful information for improving the reliability of developed software products. In this paper, to estimate the Software reliability, a robust method of estimating parameter is employed on Time domain software failure data. The results of software reliability over the two failure data sets with Inflection S-shaped model using Modified Genetic Algorithm is presented in table II and Table III.

**REFERENCES**

[1] Costa, E. O., de Souza, G. A., Pozo, A. T. R and Vergilio, S. R. (2007). "Exploring Genetic Programming and Boosting Techniques to Model Software Reliability," IEEE Transactions on Reliability, vol.56, no. 3, pp. 422-434.

[2] Dai, Y. S. Poh, K. L and Yang, B. (2003). "Optimal Testing-Resource Allocation with Genetic Algorithm for Modular Software Systems," Journal of Systems and Software, vol. 66, no. 1, pp. 47-55.

[3] Goel, A. L. (1985). "Software Reliability Models: Assumptions, Limitations, and Applicability," IEEE

Transactions on Software Engineering, vol. 11, no. 12, pp. 1411-1423.

[4] Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[5] Huang, C. Y., Lin, C. T., Kuo, S. Y., Lyu, M. R and Sue, C. C. (2004). "Software Reliability Growth Models Incorporating Fault Dependency with Various Debugging Time Lags," Proceedings of the 28<sup>th</sup> IEEE International Computer Software and Applications Conference (COMPSAC 2004), pp.186-191 vol.181.

[6] Jiang, H. Y. (2006). "Can the Genetic Algorithm Be a Good Tool for Software Engineering Searching Problems?," Proceedings of the 30<sup>th</sup> IEEE International Computer Software and Applications Conference (COMPSAC 2006), pp. 362-366, Chicago, USA.

[7] Kapur, P.K., Sunil kumar, K., Prashant, J. Ompal, S. (2009). "Incorporating concept of two types of imperfect debugging for developing flexible software reliability growth model in distributed development environment", Journal of Technology and Engineering sciences, Vol.1, No.1; Jan-Jun.

[8] Lin.C.T and Huang. C. Y., (2008). "Enhancing and Measuring the Predictive Capabilities of Testing-Effort Dependent Software Reliability Models," Journal of Systems and Software, vol. 81, no. 6, pp. 1025-1038.

[9] Lyu, M. R. Handbook of Software Reliability Engineering, McGraw-Hill, 1996.

[10] Harman. M and B. F. Jones, (2001). "Search-Based Software Engineering," Information and Software Technology, vol. 43, no. 14, pp. 833-839.

[11] Minohara, T and Tohma, Y. (1995). "Parameter Estimation of Hyper-Geometric Distribution Software Reliability Growth Model by Genetic Algorithms," Proceedings of the 6<sup>th</sup> IEEE ISSRE 1995, pp. 324-329, Toulouse, France.

[12] Mitchell, M. An Introduction to Genetic Algorithms, The MIT Press, 1998.

[13] Musa, J.D., Iannino, A., Okumoto, k., 1987. "Software Reliability: Measurement Prediction Application". McGraw-Hill, New York.

[14] Ohba, M. (1984). "Software Reliability Analysis Models," IBM Journal of Research and Development, vol. 28, no. 4, pp. 428-443.

[15] Pham. H., 1993. "Software reliability assessment: Imperfect debugging and multiple failure types in software development". EG&G-RAAM10737; Idaho National Engineering Laboratory.

[16] R. Satya Prasad, K. Prasada Rao and G. Krishna Mohan. "Control Chart Procedure for Software Reliability: Inflection S-Shaped Model", International Journal of Electronics Communication and Computer Engineering. Volume 3, Issue 6, Nov-Dec, 2012.

[17] R. Satya Prasad, K. Prasada Rao and G. Krishna Mohan. "Software Reliability using SPRT: Inflection S-shaped Model", International Journal of Application or Innovation in Engineering & Management. Volume: 2, Issue: 6, June, 2013.

[18] Wood, A. (1996). "Predicting Software Reliability," IEEE Computer, vol. 29, no. 11, pp. 69-77.

#### Authors Profile:

Dr. R. Satya Prasad received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Andhra Pradesh. He received gold medal from Acharya Nagarjuna University for his outstanding performance in Masters Degree. He is currently working as Associate Professor in the Department of Computer Science & Engineering, Acharya Nagarjuna University. His current research is focused on Software Engineering. He has published 100 papers in National & International Journals.



G. Bharathi, Working as an Asst. Professor in the Department of Computer Science, St. Marys Group of Institutions, Guntur. She obtained her M.C.A. degree from Madurai Kamaraj University, Madurai, M.Tech. (CSE) from Acharya Nagarjuna University, Guntur. Her research interests lies in Software Engineering and Cloud Computing.