International Journal of Advanced Science and Research
ISSN: 2455-4227
Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 3; Issue 1; January 2018; Page No. 144-147

A Comprehensive Literature Survey on the Use of Optimization Techniques for Parameter Estimation of Software Reliability Growth Models

Mandeep Kaur
Assistant Professor, Department of Computer Science, Khalsa College for Women, Civil Lines, Ludhiana, Punjab, India

Abstract
Software Reliability is defined as the probability with which the software will operate without any failure for a specific period of time in a specific environment. It is one the essential software quality features. Software reliability estimated in early phases of software development life cycle saves a lot of time and money as it prevents spending huge amount of money on fixation of defects in the software after it has been deployed. Software reliability estimation has thus become an important research area as every organization aims to produce defect free software. There are many software reliability growth models that are used to assess the reliability of the software. These models help in developing robust and fault tolerant systems. This paper presents a review of optimization techniques used by many researchers on the software reliability growth models in order to estimate the software reliability effectively.

Keywords: reliability, software reliability growth models, parameter estimation, optimization

1. Introduction

With increasing complexity, changing requirements and distributive nature of the software system it has become difficult to develop reliable software where reliability is the probability of software failure occurrence. The failure of the software is attributed to errors, faulty functionality, ambiguities, improper requirement analysis, inefficient code, inadequate testing, timing, sequencing, data, and exception handling. Software reliability is an important feature of software quality along with other features like: usability, performance, functionality, maintainability, instability, serviceability, documentation, etc. [1].

Software failure can be classified into various categories like [2]:

a) **Transient failure** which occurs for specific inputs,
b) **Permanent failures** which occur for all inputs,
c) **Recoverable failure** where the system can recover with or without any operator intervention,
d) **Unrecoverable failure** where the system needs to be restarted,
e) **Cosmetic Failures** which cause minor irritations.

Since there is no software system that is failure-free, every software development contract includes the specification of reliability requirements. Software reliability models are statistical models which are used to make predictions about a software system's failure rate on the basis of the failure history of the system. These models make assumptions related to fault discovery and removal process thereby determining the form of the model and the meaning of the model's parameters. Once the parameter is known and the current numbers of defects are discovered, the remaining defects are estimated (Fig.1). These residual defects help to decide whether the code is ready to ship or not and gives an estimate of the number of failures that the customers will encounter while using the software which further helps in estimating the appropriate levels of support and the cost of support that will be required for defect correction once the software has been delivered.

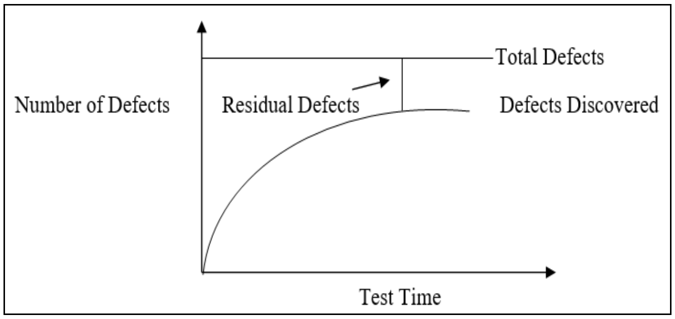

Fig 1: Residual Defects

Software reliability modeling techniques can be broadly classified into two subcategories [3]:

a) **Prediction models**: These models use historical data to predict reliability. This is usually done before development or testing phase. These models support planning and sensitivity analysis. Example: shortcut model, full scale model, Rayleigh model, Neufelder assessment model, etc. [4].
b) **Estimation model**: These models use data from current software development effort. Reliability is estimated during system level testing or operation phase. They help in forecasting the failure rate or Mean time Between Failure (MTBF). Example: Goel-Okumoto model, Weibull model, S- shaped model, etc.

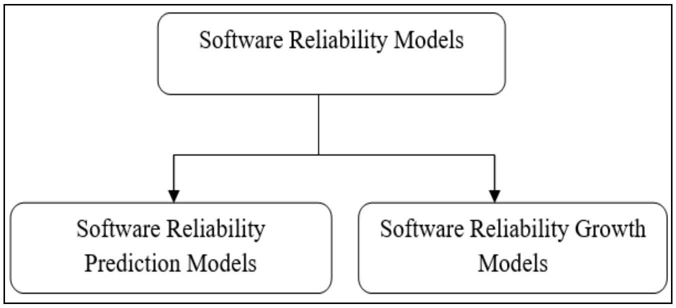

Fig 2: Types of Reliability Models

It is essential to accurately determine the parameters of software reliability. The more accurate the measurement of the parameters, the more accurate the Software Reliability Growth Model (SRGM) will be. So there is a requirement of a process which helps in optimally estimating the parameters of software reliability. Optimization is a technique to achieve highest performance by maximizing the desired features and minimizing the undesired ones. This paper reviews various optimization techniques that have been used by many researchers to optimize the parameters of estimation in software reliability growth models to estimate accurately the reliability of the software.

2. Software Reliability Growth Models

A reliability growth model is used in testing phase to determine how the system reliability changes over time. Software Reliability Growth Model (SRGM) are statistical models used to detect defects using mathematical functions which help in predicting failure rate or residual defects [5]. Reliability growth models are divided into two types: concave and s-shaped.

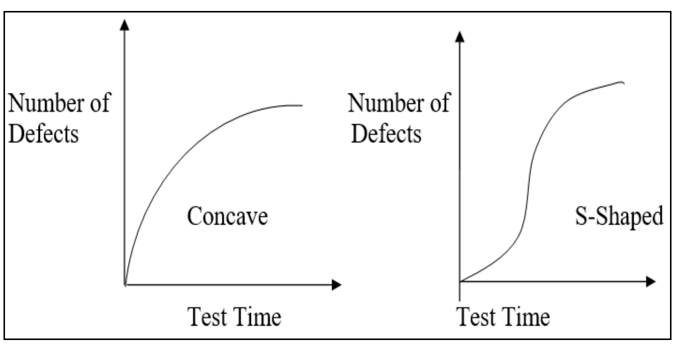

Fig 3: Types of Software Reliability Growth Models

Software reliability growth model data is classified as [6]:

a) **Test Time Data**: It consists of calendar time, execution time, number of test runs.
b) **Defect Data**: It consists of major defect data which can tolerate the situation for some time and critical defect data which requires the solution immediately.
c) **Grouped Data**: It is the combination of test time and amount of failures detected.

2.1 Software Reliability Modeling Process

a) **Examination of data**: It includes plotting data with respect to time and analysis of this data.
b) **Model selection**: This is done on the basis of data, test process, assumptions related to model.
c) **Parameter estimation**: This includes use of statistical functions such least square method or maximum likelihood.
d) **Obtaining fitted model**: This includes substitution of estimated parameters into selected model.
e) **Evaluation of fitness**: A goodness of fit test is conducted to check if the selected model is appropriate or not.
f) **Reliability predictions**: Reliability predictions are made based on the selected model.

2.2 Examples of software reliability growth models:

Table 1: Software Reliability Growth Models and their prediction functions

Software Reliability Growth Model	μ (t)
G-O Model	$a(1-e^{-bt})$
Delayed S- shaped model	$a(1-(1+bt)e^{-bt}$
Weibull model	$a(1-e^{-bt})^{\alpha}$
M-O model	a ln(1+bt)
JM model	N(1-exp(-φt))

2.3 Factors affecting the performance of Software Reliability Growth Models [7]

a) Unsolved problematic issues
b) Complexity of the reliability models
c) The Misconception of fault and failure phenomena
d) Unfounded types of assumptions
e) Inaccurate modeling parameters
f) Difficulty in selecting the reliability models
g) Weakness of the reliability model
h) Complexity of the software

3. Optimization Techniques

a) **Cuckoo Search**: Cuckoo search is inspired by the bird cuckoo called “Brood Parasites” birds who never build their own nest and lays their eggs in the nest of another host bird. If the host bird finds that the eggs in its nest are not their own eggs then it will either throw those eggs away or abandon its nest and build a new nest. In a nest, each egg represents a solution and cuckoo egg represents a new and good solution. The obtained solution is a new solution based on the existing one and the modification of some characteristics [8].
b) **Ant Colony**: Ants leave a pheromone trail from the source of food to the nest and vice versa. This pheromone has shorter evaporation time; hence the shortest path is followed by more ants and that path has stronger pheromone concentration.
c) **Firefly Optimization**: Firefly optimization is a nature inspired technique which is based on behavior of fireflies which use flashes for communication, for attracting prey and as protective warning mechanism. The brighter the flash, the more the fireflies with lesser flashes are attracted towards them. [9]
d) **Particle Swarm Optimization** (PSO): PSO is based on natural behavior of bird flocks and fish school. The particles move around in the search space. Each particle

has an associated position and velocity. In every iteration particles best position is evaluated and a global best position in the search space is evaluated. This helps to move the swarm towards the best solutions.

e) **Grey Wolf**: The GWO algorithm is based on the wild behavior of the grey wolves during hunting. The GWO divides the animal population into four categories: alpha, beta, delta, and omega. The optimization process is same as hunting, guided by the highest rank leaders: alpha, beta and delta which represent the best three solutions in the search space. The omega wolves rank lowest in the hierarchy and represent the rest of the solutions that must adjust their positions to follow the other dominant wolves.

f) **Bee Colony optimization**: In Bee colony optimization the search space is explored by every artificial bee which apply a predefined number of moves, thereby constructing and/or improving the solution, yielding a new solution. Then all the bees move back to the hive and share information about the quality of their solutions i.e. the value of the objective functions is computed. Then every bee decides randomly whether to continue its own exploration and become a recruiter or a follower. This process is repeated if solutions are incomplete.

4. Related Work

Sheta and Raouf [10] used grey wolf optimizing algorithm with the objective of minimizing the difference between the actual failures and the estimated accumulated failures. The methodology employed estimating the parameters of three software reliability growth models: the exponential model, power model, and S-shaped model. The proposed models were applied to three real measured test/debug datasets. The results show that the proposed methodology was able to successfully estimate the parameters of SRGMs.

Jin and Wei-Jin [11] suggested that optimization of the parameters of reliability model is a necessary task. The research used swarm intelligent optimization algorithm-quantum particle swarm optimization (QPSO) algorithm to optimize these parameters of SRGMTEF. The performance of the proposed model with optimized parameters was compared with other existing models. The results obtained showed that the proposed parameter optimization approach using QPSO was very effective and flexible, and the better software reliability growth performance was obtained based on SRGMTEF on the different software failure datasets.

Saati and Marwa [12] used Cuckoo Search (CS) for estimating the parameters of software reliability growth models: the Exponential, Power, S-Shaped, and M-O models. The results of Cuckoo Search were compared to Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and extended ACO. It was found that the performance of CS was better as compared to both PSO and ACO in finding better parameters which were tested using identical datasets. In this research the percentages of training data to testing data were also investigated to show their impact on the results. It was observed that the CS algorithm could be successfully applied to find good and acceptable solutions to the problem of parameter estimation of Software Reliability Growth Models. The search strategy of the CS efficiently navigated throughout the search space of the problem and located very good solutions using fewer iterations and smaller populations. It was concluded that increasing the percentage of training data, makes training hard and testing very simple and not sufficient enough. On the contrary, when percentage of training was decreased, the training data became insufficient to train and the testing data became very large and unnecessary.

Choudhary et al. [13] presented an effective parameter estimation technique for software reliability growth models using firefly algorithm. The authors suggested some limitations of software reliability prediction techniques like: use of numerical estimation methods for parameter estimation, local optimization, biasness and model's parameter initialization, which were not able to find the optimal model parameters. The research used firefly algorithm to overcome these limitations and provided an optimal solution for parameter estimation of software reliability growth models. Goel-Okumoto model and Vtub based fault detection rate model were selected to validate the results. Seven real world datasets were used to compare the proposed technique against Cuckoo search and CASRE tool. The results indicated superiority of proposed model over other existing estimation techniques.

Bidhan and Awashti [14] used swarm intelligence based technique named as Particle Swarm Optimization for the evaluation of growth models which presented better and optimized results. It helped in avoiding problems that occurred while estimating software reliability growth parameters using traditional methods. Particle Swarm Optimization was used along with some modifications for estimation of the NHPP based Reliability Growth Models.

Mallikharjuna and Anuradha [15] suggested that each software reliability growth model with testing effort function (SRGMTEF) contains some undetermined parameters which require optimization. The researchers used swarm intelligent optimization algorithm called Modified Genetic Swarm Optimization algorithm to optimize these parameters of SRGMTEF. The performance of the proposed model was compared with other existing models like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The experiment results showed that the proposed parameter optimization approach using Modified Genetic Swarm Optimization was very effective, flexible and provided better software reliability growth performance. The paper also provided comparison of ten SRGMs like Goel-Okumoto Model, Delayed S-shaped Growth Model, Yamada Imperfect Debugging Models, Yamada Rayleigh Model, Inflection S-shaped Model, etc.

Malhotra and Negi [16] proposed the use of particle swarm optimization (PSO) to resolve the problem of software reliability growth models parameter estimation. The research compared the results of the proposed model with that of genetic algorithm. The research was conducted on 16 projects and the results obtained showed high predictive ability of PSO with low error predictions.

Mallikharjuna and Auradha [17] used artificial bee colony optimization for parameter estimation in software reliability growth models. It was found that the optimization technique provided better results and hence reliability growth was considerably improved.

Zheng et al. [18] proposed Ant Colony Optimization for estimating the reliability of software. Numerical examples based on three sets of real failure data were presented. It was shown that the traditional method can't get possible solutions for some models and datasets, while the proposed method always can get one and results of proposed PSO algorithm had nearly 10 times higher accuracy for most of the models.
Shanmugam and Florence [19] used Ant Colony Algorithm for parameter estimation of software reliability growth models. The research used numerical examples based on five sets of real failure data. The results obtained from proposed model showed higher precision, faster convergence speed and higher accuracy for at least 10 times for majority of the models.

5. Conclusion

Reliability is an important characteristic of software quality. It helps in determining the trust worthiness of software and help developers in developing a fault tolerant and efficient system. Various software growth models are used to estimate the reliability of software. Optimization of these parameters helps to predict the failure rate or residual defects efficiently. This paper provided a survey of various optimization techniques like ant colony, bee colony, grey wolf, firefly, particle swarm, etc. that have been used by many researchers to optimize the parameters of software reliability models. It has been observed that using optimization techniques for parameter estimation in software reliability growth models is computationally much easier and efficient and these optimization techniques provide better results as compared to traditional techniques.

6. References

1. Pressman RS. Software Engineering: A Practitioner's Approach, McGraw-Hill series III Computer Science, New York, 2001.
2. Mall R. Fundamentals of Software Engineering, PHI Learning Private Limited, New Delhi, 2008.
3. Pan J. Software Reliability, Carneige Mellon University, 1999.
4. https://en.wikipedia.org/wiki/List_of_software_reliability_models
5. Wood A. Tandem Software Reliability Growth Models, Technical Report, 1996.
6. https://www.slideshare.net/himanshuhora/software-reliability-growth-model
7. Faqih KMS. What is Hampering the performance of Software Reliability Models? A Literature review, Proceedings of the International Multi Conference of Engineers and Computer Scientists, 2009, 1.
8. Pentapalli VVG, Varma RK. Cuckoo Search Optimization and its Applications: A Review, International Journal of Advanced Research in Computer and Communication Engineering, 2016, 5 (11).
9. Kaur M. Estimation of Effort Using Nature Inspired Optimization Techniques, International Journal of Academic Research and Development, 2018, 3(1).
10. Sheta AF, Raouf A. Growth Models Using the Grey Wolf Optimization Algorithm, International Journal of Advanced Computer Science and Applications. 2016, 7(4): 499-505.
11. Jin C and Wei-Jin S. Parameter Optimization of Software Reliability Growth Model with S-Shaped Testing-Effort Function Using Improved Swarm Intelligent Optimization, Applied Soft Computing, 2016, 40: 283-291
12. Saati NAA, Marwa AA. The Use of Cuckoo Search in Estimating the Parameters of Software Reliability Growth Models, International Journal of Computer Science and Information Security, 2013, 11 (6).
13. Choudhary A, Baghel AS, Sangwan OP. Parameter Estimation of Software Reliability Model Using Firefly Optimization, Data Engineering and Intelligent Computing, 2017:407-415.
14. Bidhan K, Awasthi A. Estimation of Reliability Parameters of Software Growth Models Using a Variation of Particle Swarm Optimization, International Conference on Confluence The Next Generation Information Technology Summit (Confluence), 2014.
15. Mallikharjuna RK, Anuradha K. A New Method to Optimize the Reliability of Software Reliability Growth Models using Modified Genetic Swarm Optimization, International Journal of Computer Applications, 2016, 145 (5): 1-8.
16. Malhotra R, Negi A. Reliability Modeling Using Particle Swarm Optimization, International Journal of System Assurance Engineering and Management, 2013, 4 (3):275-283.
17. Mallikharjuna RK, Anuradha K. An Efficient Method for Parameter Estimation of Software Reliability Growth Models using Artificial Bee Colony Optimization, International Conference on Swarm, Evolutionary and Memetic Computing, 2014:765-777.
18. Zheng C, Liu X, Huang S, Yao Y. Parameter Estimation Method for Software Reliability Models, Procedia Engineering, 2011, 15:3477-3481.
19. Shanmugam L, Florence L. A Comparison of Parameter Best Estimation Method for Software Reliability Models, International Journal of Software Engineering & Applications, 2012, 3(5).