



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Parameters Estimation of Goel's Okumotu Model using Simulated Annealing

Chander Diwaker¹, Pradeep Tomar²

CSE Department, U.I.E.T., Kurukshetra University, Kurukshetra, INDIA¹

CSE Department, School of ICT, GBU, Greater Noida², Uttar Pradesh, INDIA²

Abstract: Goel's Okomtu model is based on the Exponential model and simple non-homogeneous Poisson process (NHPP) model. Some optimization technique has been discussed which can be used to optimize the Goel's Okomtu models. Simulated Annealing (SA) is a heuristic optimization modelling technique that can be applied to solve many difficult problems in the various fields such as scheduling, modelling etc. PSO (Particle Swarm Optimization) and ACO (Ant Colony Optimization) algorithms are explored to estimate Software reliability Growth model (SRGM) parameters. These algorithms were used to handle the modelling problems for the Power model, the Delayed S-Shaped model and the Exponential model. The optimized result to estimate the parameters using Simulated Annealing (SA) algorithm are better as compared to other techniques.

Keywords: SRGM, Goel's Okumotu Model, Optimization Techniques, Simulated Annealing.

I. Introduction

Software Reliability is a critical Component of Computer System availability. SRGM attempts to correlate defect detection data with estimated residual defects and time. SRGMs help in decision making in many software development activities such as number of initial faults, reliability within a specified interval of time period, number of remaining faults, cost analysis and release time, failure intensity etc. There are two methods which are used to predict the software Reliability Models such as Maximum Likelihood Estimation (MLE) and Least Squares Estimation (LSE) methods[1].

MLE have the following general properties:

- MLE are asymptotically normally distributed.
- MLE are asymptotically efficient—no asymptotically unbiased estimator has a smaller asymptotic variance.
- MLE are asymptotically unbiased although they may be biased in finite samples.

Maximum-likelihood estimators are consistent.

LSE methods are still attractive in terms of goodness of fit performance and predictive performance in many cases. LSE methods that are used are Basic weighted least squares method, Modified Weighted Least Squares Method and Root Least Squares Method.

II. Optimization Techniques

There are following optimization techniques for parameter estimation in reliability models:

1) **Simulated Annealing Algorithm:** Annealing involves heating and cooling of a material to modify its physical properties due to change in inner structure. Travelling Salesman Problem is an example of this algorithm. The algorithm is based on randomization techniques having its iterative improvement based on local search. Decreasing temperature in the cooling schedule corresponds to narrowing of the random search process in the neighbourhood of the current solution. In comparison to GA the SA algorithm is less complicated and more effective [2].

The key algorithmic feature of simulated annealing means to escape local optima by allowing hill-climbing moves. As there is decrease in the temperature parameter towards zero hill climbing moves occur less frequently and the solution distribution associated with the in homogeneous Markov chain is concentrated on the set of globally optimal solutions [3]. Heating means randomly modifying the variable values. Higher heat implies greater random fluctuations. The cost function returns the output f associated with a set of variables.

Advantages of using this technique

- The failure rate is reduced.
- Increase the software reliability and efficiency of the software.
- Easy to implement than any other optimization technique.

2) **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 [4]. 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 [5] by the use of genetic algorithm for allocating the testing resources. Using simple optimization techniques it's somewhat not easy to allocate resources optimally to software during testing phase. The dynamic nature of problem is also not easily solved by ordinary optimization techniques.

3) *Particle Swarm Optimization (PSO)*: PSO is a particle swarm optimization algorithm for global optimization. PSO is similar to continuous genetic algorithm. 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. On 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, position of the each particle is updated using the new velocity of particle [6].

4) *Ant Colony Optimization (ACO)*: 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 [1]. 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 remaining 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 [8]. The limitation of ACO was: - It can solve optimization problems but it is proved fail at the time of convergence.

III. Proposed Work

Proposed Model will estimate the following parameters using Simulated Annealing (SA) algorithm.

1. To generate best values of a, b of Goel's Okumotu model using simulated annealing
2. To predict the best failure rate of new project using above optimized values generated.
3. To improve the reliability of proposed model.

- *Algorithm of Proposed Model*

1. Load all existing project failure data into a properties object p₁.
2. Initialize b = 0.098076 (Best result produced by a premium project historical data)
3. Initialize dist = MAX. VALUE
4. Initialize temp = 100, change = 0.99
5. Repeat steps 6 to 10 while temp >= 0.1
6. Generate a new particle C₁. Call C₁ = generate(size, b, p₁)
7. Generate all values of a parameter using b parameters with formula

$$a = \sum_{i=1}^n a / (1 - e^{-bt})$$

8. Call dist2 = distance (a, C₁, b, p₁)
9. if dist2 < dist then
 - dist = dist2
 - best = C₁
 - else
 - if ((0.2 + random * 1.5) <= e^{-(dist2-dist)/temp})
 - then
 - dist = dist2
 - best = C₁
 - [Endif]
- [Endif]
10. temp = temp * change
- [End of Repeat]
11. Result = best
- Algo distance (a, b, p₁)
1. Initialize sum = 0

2. Repeat for $I = 1$ to $p_1.size$
3. $mt = p_1(1)$
4. $sum = sum + (mt - avg(b))^2$
[End of Repeat]
5. Return

Implementation and Result Analysis: Net-Beans IDE is used to implement the proposed work. The NetBeans IDE is open source and is written in the Java programming language.



Fig. 1 Input Screen

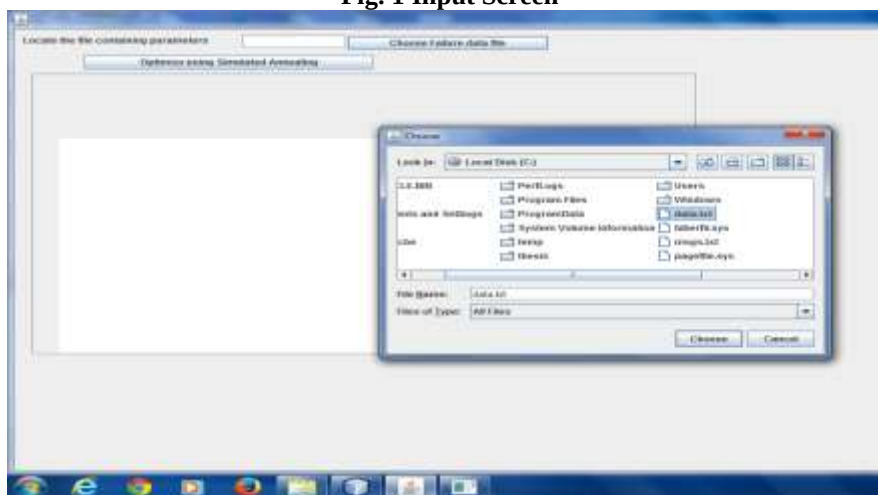


Fig. 2 Select the data to be optimized

In fig. 2 the data of traditional model is selected from the specified location which is used as the starting values in the proposed model.

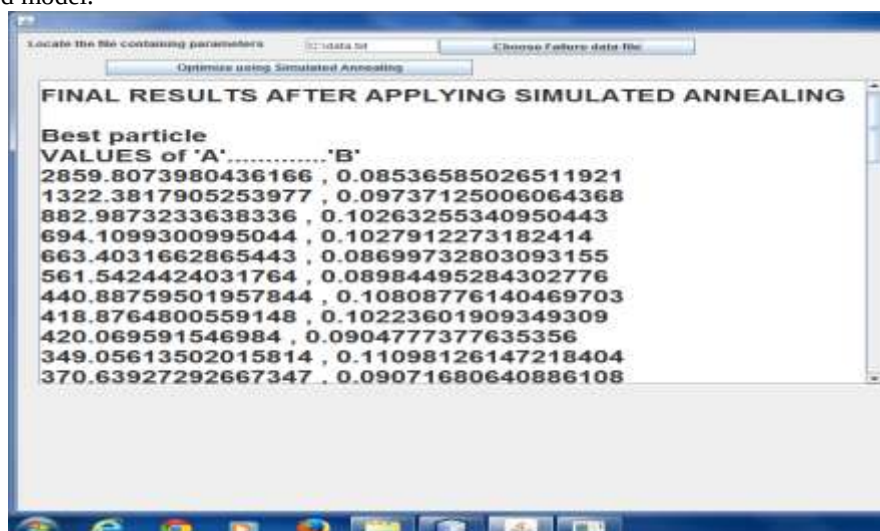


Fig. 3 Results after applying simulated annealing

In fig. 3 the values of 'a' and 'b' for proposed model are calculated using the data to traditional model. The values of 'a' and 'b' are much more optimized in the ending weeks of the project execution time as compared to the earlier weeks.

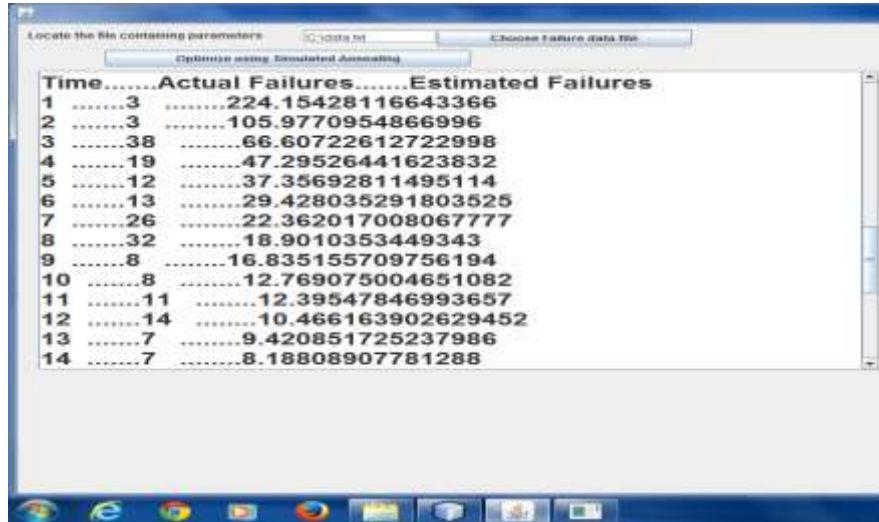


Fig. 4 Comparison between actual data and estimated data

RESULTS Sum of Failure

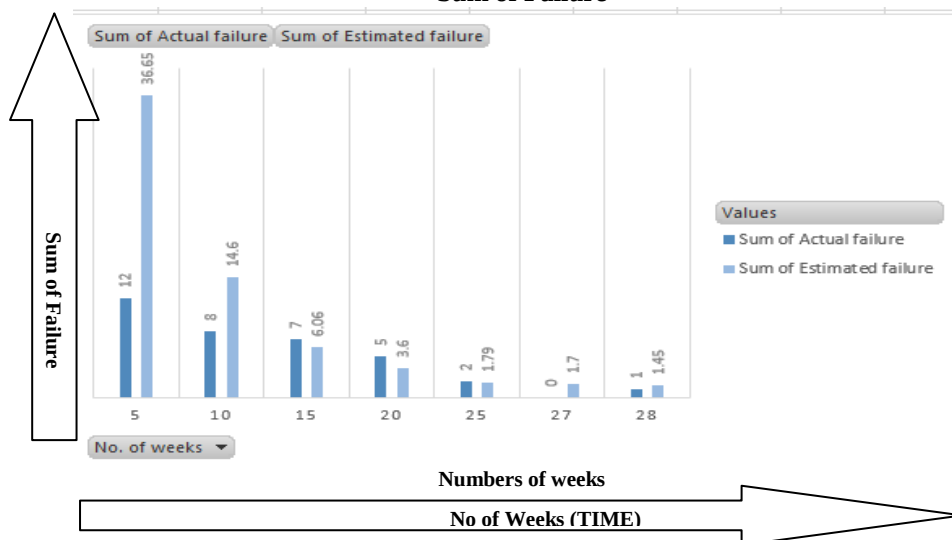


Fig. 5 Graph for Actual failure and estimated failure vs. time

No. of weeks	Actual time	Estimated time
5	12	36.65
10	8	14.6
15	7	6.06
20	5	3.6
25	2	1.79
27	0	1.7
28	1	1.45

Table 1 Data set for fig. 5

In fig. 6 x-axis is depicting number of weeks. The number of times each project is working according to the days. The y- axis is depicting the number of failures per week. The comparison between actual failures and estimated failures is depicted in fig. 6.

IV. Conclusion

The optimized results using simulated annealing are far better than other techniques such as PSO, ACO, and Neural Network. The development time and budget are high in PSO as compared to SA. In PSO, good results may be obtained by increasing the number of iteration which indirectly decreases the efficiency. But in SA, efficiency may be increased with decreasing the numbers of iteration. The SA will increase the reliability of the software as a result of decreasing in failure rate.

V. References

- [1] Latha Shanmugam and Dr. Lilly Florence, "A Comparison of Parameter Best Estimation Method for Software Reliability Models", International Journal of Software Engineering & Applications, Vol.3, No.5, pp. 91 – 102, 2012.
- [2] Zainab Al-Rahamneh, Mohammad Reyalat and Alaa F. Sheta, " A New Software Reliability Growth Model: Genetic-Programming-Based Approach", Journal of Software Engineering and Applications, pp. 476-481, 2011.
- [3] Swati Chowdhary 1, Prof.Mamta M Sarde, "Review of Various Optimization Techniques for FIR Filter Design", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2014, Vol. 3, Issue 1, pp. 6566-6571.
- [4] Dr. Satya Prasad Ravi, N.Supriya and G.Krishna Mohan, "SPC for Software Reliability: Imperfect Software Debugging Model", International Journal of Computer Science, Vol. 8, Issue 3, No. 2, 2011.
- [5] N. Ahmad P.G. and S. M. K Quadri, "Comparison of Predictive Capability of Software Reliability Growth Models with Exponentiated Weibull Distribution" International Journal of Computer Applications, Vol. 15 (16), 2011.
- [6] D. Haritha, T.Vinisha and K. Sachin, "Detection of reliable software using particle swarm optimization", GJCAT, Vol 1 (4), pp. 489-494, 2011.
- [7] Lilly Florence and Latha Shanmugam, "Enhancement and Comparison of Ant Colony Optimization for Software Reliability Models", Journal of Computer Science 9 (9), pp. 1232-1240, 2013.