Resource Allocation to Software Modules in Software Testing with Imperfect-debugging SRGM

SK.Md.Rafi
Assoc.Professor
Sri Mittapalli Institute of Technology for women
Affiliated to J.N.T.U, Kakinada

Shaheda Akthar
Assoc.Professor
Sri Mittapalli college of Engineering
Affiliated to J.N.T.U, Kakinada

ABSTRACT
Resource control and resource maintenance during the software testing is one of the finest optimization problems. During the software testing many of the resources like time, effort and budget were consumed. The main aim of the manager is allocate the resources in a constrained manner such that the effort can be optimally allocated and overall budget is minimized. In this paper we proposed a imperfect debugging SRGM during testing and resource allocation is done based on optimizing the effort and reliability. An experimental result shows the proposed model well fitted for software testing.

Keywords- Resource allocation, software testing, Imperfect-debugging SRGM, Lagrange multipliers.

Notation
m(t) :mean value function
w(t) :Current testing expenditure
W(t) :Cumulative testing expenditure
a :initial fault content $a > 0$
r :Fault detection rate
I :subscript for each module number i=1,2,3…..M
v_i :weight for each module, $v_i > 0$
z_i number of software errors remains in the softwaremodule
Z number of software errors remains in total software
q_i :amount of testing resource for the module i
W total amount of testing resource expenditure during testing
R(s) :software reliability in the interval $(0,s \mid s > 0)$
γ :constant parameter ;related to failure rate
R_0 :objective reliability
A_i :$v_i a_i r_i(1-\beta_i)$

1. INTRODUCTION

Reliability is one of the finest areas of software quality. By improving the reliability it increases the software quality. Software reliability is defined as the failure free software over a period of time in given environmental conditions. Software Reliability growth models represents the mathematical model of software testing. Several software reliability growth models are proposed in literature which describes the real time environment of software testing [1] [2]. Software reliability growth models can be classified in terms of time dependent and data dependent [14]. Generally any software development process consisting of four phases starting from requirement, design, coding and testing [4]. Testing can be organized in terms of unit or module, integration and system testing. During very testing the resources are consumed based on test criteria. Testing resources are better described by testing time, testing effort and number of test cases used during the testing. The manager has to decide how better he will allocate the resources to the testing so that testing time, testing effort and budget is optimized and quality is improved. Software reliability growth models with resources allocation problem is studied by many authors [3][4][5][6][7][8][9][10]. Several authors had used perfect SRGM in resource allocation during module testing. But it is observed that SRGM were not perfect in nature during error detection and debugging there is a change of new errors may introduce in to the testing.

In this paper we had studied the resource allocation during the module testing with an imperfect debugging software reliability growth model with testing effort. The rest of the paper is organized as follows.

Section-2 proposed a testing effort dependent software reliability growth model section-3 describes the testing resource allocation problem. Section-4 describes the numerical examples.

II. REVIEW OF SOFTWARERE LIABILITY GROWTH MODEL WITH LOGISTIC-EXPONENTIAL TEF WITH IMPERFECT DEBUGGING ENVIRONMENT.

The following are the assumptions for SRGM with logistic-exponential testing-effort [12][13]

1) the fault removal process follows the NHPP
2) The software is subject to failure at random time's causes by faults remaining in the system.
3) The mean number of faults detected in the time interval (t, t+Δt) by the current testing-effort is proportional to the mean number of remaining faults in the system.
4) The fault detection rate is constant
5) The consumption of testing-effort is modeled by a logistic-exponential TEF
6) Each time a failure occurs it is removed and there is chance of introducing new errors in the software during testing.

Cumulative testing effort is described by Logistic-exponential testing-effort has a great flexibility in accommodating all the forms of the hazard rate function, can be used in a variety of problems for modeling software failure data.

The logistic-exponential cumulative TEF over time period (0,t] can be expressed as [15]

$$W(t) = \alpha \times \frac{(e^{h\times t}-1)^k}{(1+(e^{h\times t}-1)^k)} \quad t>0 \quad (1)$$

α is the total expenditure , k positive shape parameter and h is a positive scale parameter

$$\frac{dm(t)}{dt} \times \frac{1}{w(t)} = r(t) \times [n(t) - m(t)] \quad (2)$$

Here we assume $n(t) = a + \beta \times m(t)$ (3)

Solving above by considering m(0)=0 and r(t)=r we obtain the mean value function

$$m(t) = \frac{a}{(1-\beta)} \times (1-\exp[-r\times(1-\beta)\times W^*(t)]) \quad (4)$$

In this case, we have

$$z(t)=n(t)-m(t)= a \times \exp[-r \times (1-\beta) \times W^*(t)] \quad (5)$$

Now we have the following reliability after the testing

$$R(s) = \exp(-\gamma \times z(t) \times s) = \exp[-\gamma \times a \times \exp[-r\times(1-\beta)\times W^*(t)] \times s] \quad (6)$$

III TESTING RESOURCE ALLOCATION PROBLEM

In this paper we used two testing resource allocation schemes one by minimizing the number of remaining faults and allocating the resources to attain the maximum reliability.

3.1 resource allocation by minimizing the number of remaining faults[4][5]

1) The software system is composed of M independent modules. The number of faults remain in the software module is estimated through the software reliability growth model.
2) The total amount of testing resource for module testing is specified
3) The manager has to allocate the testing resource to all modules in such way the total number of errors present in the software after testing can be minimized.

From (5) the estimated number of remaining faults is formulated by

$$z_i = a_i \times \exp[-r_i \times (1-\beta_i) \times q_i] \quad (i = 1,2,\ldots\ldots M). \quad (7)$$

Test resource allocation is formulated as

$$\text{Minimize} \sum_{i=1}^{M} v_i \times a_i \times \exp[r_i \times (1-\beta_i) \times q_i] \quad (8)$$

$$\text{Suchthat} \sum_{i=1}^{M} q_i \le W, \; q_i \ge 0 \;(i = 1,2,3\ldots\ldots M) \quad (9)$$

The parameters of a_i , r_i and β_i are already estimated by the model introduced in section II

To solve the above problem, we consider the following Lagrangian equation:

$$L = \sum_{i=1}^{M} v_i \times a_i \times \exp[-r_i \times (1-\beta_i) \times q_i] + \lambda \times (\sum_{i=1}^{M} q_i - W) \quad (10)$$

Now differentiate above equation with respect to q_i

$$\frac{\partial L}{\partial q_i} = v_i \times a_i \times r_i \times (1-\beta_i) \times \exp(-r_i \times (1-\beta_i) \times q_i) + \lambda \ge 0 \quad (11)$$

$$\frac{\partial L}{\partial q_i} = 0 \quad (i = 1,2\ldots\ldots M) \quad (12)$$

The following condition is satisfied for tested modules:

$$A_1 \ge A_2 \ge A_3 \ge \ldots\ldots A_{k-1} \ge A_k \ge \ldots\ldots\ldots\ldots \ge A_M \quad (13)$$

Above sequence is arranged according to the fault detect ability for the tested modules. Now if $A_k > \lambda \ge A_{k+1}$ now we have

$$[\ln(A_i) - \ln(\lambda)] = r_i \times (1-\beta_i) \times q_i \quad (14)$$

$$q_i = \frac{(\ln A_i - \ln \lambda)}{r_i.(1-\beta_i)} \quad (15)$$

Now next differentiate with respect to λ then

$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{M} q_i - W = 0 \text{ Then } \sum_{i=1}^{M} q_i = W \quad (16)$$

$$\sum_{i=1}^{M} \frac{(\ln A_i - \ln \lambda)}{r_i.(1-\beta_i)} = W \quad (17)$$

Solving above equation

$$\ln \lambda = \frac{(\sum_{i=1}^{M} \frac{1}{r_i \times (1-\beta_i)} \ln A_i - W)}{\sum_{i=1}^{M} \frac{1}{r_i \times (1-\beta_i)}} \quad (18)$$

Optimal Lagrange multiplier λ^* exist in the set $\{\lambda_1, \lambda_2 \ldots\ldots\ldots, \lambda_M\}$ .Optimal λ^* can be calculated by setting k=1 and then compute λ_k by eq (18) for which $A_k > \lambda_k > A_{k+1}$ then $\lambda^* = \lambda_k$.

3.2. Minimizing the remaining faults by considering the reliability

Following are the test resource allocation problem [5]

1) The software system is composed of M independent modules. The number of faults remain in the software module is estimated through the software reliability growth model.
2) The total amount of testing resource for module testing is specified
3) The manager has to allocate the testing resource to all modules by setting the reliability objective in such way the total number of errors present in the software after testing can be minimized.

From eq.(6)we obtain

$$R(s) = \exp[-\gamma_i \times a_i \times \exp[-r_i \times (1-\beta_i) \times q_i] \times s] > R_0 \quad (19)$$

From (19) we obtain the q_i as

$$q_i \ge -\frac{1}{r_i \times (1-\beta_i)} \ln[\frac{-\ln R_0}{\gamma_i \times a_i \times s}] \quad (20)$$

The right hand side of the equation denotes D_i. We have the following equation we show the relation between allocated resource and maximum allocated resource.

$$q_i \ge c_i \quad (c_i = \max\{0, D_i\}; i = 1,2,3\ldots\ldots M)$$

By setting the $x_i = q_i - c_i$,, we obtain the transform eq.(8) and (9)

$$\text{Minimize} \sum_{i=1}^{M} v_i \times a_i \times \exp[-r_i \times (1-\beta_i) \times (x_i + c_i)] \quad (21)$$

Suchthat

$$\sum_{i=1}^{M} x_i \le W - \sum_{i=1}^{M} c_i, x_i \ge 0 \; (i = 1,2,3.\dots\dots\dots,M)$$

(22)

By suitable transformation above problem can be deduced to eq (8) and (9)

IV. NUMERICAL EXAMPLES

Parameters of the testing effort function can be obtained by MLE and LSE. Here we assume that all parameters like a_i, r_i and β_i (i=1,2………,M) are already been calculated by using the software fault detection data and the testing resource data observed in module testing. Present scenario we considered a software data consists of 10 modules we estimated the parameters a_i, r_I and β_i given in the Table.1

Optimal q_i^* can be obtained from (15) are shown in the Table.1. The estimated optimal λ^* =0.000911.
Then total numbers of remaining faults are estimated from (6)

$$Z = \sum_{i=1}^{10} a_i \times \exp[-r_i \times (1-\beta_i) \times q_i] = 104.52 \quad (23)$$

The total number of faults are decreased from 251 to 104.52 by use of test resource expenditure 97,000 man hours; about ((251-104.52)/251*100) =58.35% reduction in the remaining faults.

Table .1 the values of v_i , a_i , r_i , and β_i and allocated testing resource expenditure q^* for minimizing the remaining faults W=97,000.

Module	v_i	a_i	$r_i(10^{-4})$	β_i	q_i	z_i
1	1	63	0.5332	0.0010	24478.7	17.1
2	1	13	2.5230	0.0023	5079.9	3.61
3	1	6	5.2620	0.0014	2362.6	1.73
4	1	51	0.5169	0.0045	20561.4	17.7
5	1	15	1.7070	0.0012	6054.21	5.34
6	1	39	0.5723	0.0040	15650.38	15.9
7	1	21	0.9938	0.0010	8339.2	0.17
8	1	9	1.7430	0.0038	3107.9	5.24
9	1	23	0.5057	0.0087	4699.2	18.1
10	1	11	0.8782	0.0065	597.3	10.4

Next by using the values of v_i , a_i , r_i , β_i and γ_i given in the Table.2 we have the optimal solution q^* shown in the Table.2. Now reliability R_0 =0.9 is objective value for the software and s=1. Then total numbers of faults are decreases from 251 to 104.45 by using the total resource expenditure 97,000 man hours; about 58.38% of reduction in the remaining faults.

Then total numbers of remaining faults are estimated from (6)

$$Z = \sum_{i=1}^{10} a_i \times \exp[-r_i \times (1-\beta_i) \times q_i] = 104.45 \quad (24)$$

Table 2: values of v_i , a_i , r_i , β_i and γ_i given and allocated testing resource expenditure q^* for minimizing the remaining faults W=97,000.

Module	a_i	β_i	$r_i(10^{-4})$	$\gamma_i(10^{-2})$	D_i	q_i	z_i	v_i
1	63	0.0010	0.5332	0.1800	1377.7	24498	18.3	1
2	13	0.0023	2.5230	1.124	1293.2	5081	5.11	1
3	6	0.0014	5.2620	3.167	119.1	2362.8	3.57	1
4	51	0.0045	0.5169	0.2180	1035.4	20576.5	18.98	1
5	15	0.0012	1.7070	0.856	1157.1	6059	6.75	1
6	39	0.0040	0.5723	0.2900	1233.6	15664.1	17.27	1
7	21	0.0010	0.9938	0.5470	868.65	8348.9	10.51	1
8	9	0.0038	1.7430	1.4060	1046.8	3108.9	6.66	1
9	23	0.0087	0.5057	0.4830	1038.0	4769	19.46	1
10	11	0.0065	0.8782	1.0850	1410.4	600.73	11.8	1

V. CONCLUSION AND REMARKS

In this paper we have studied two resource allocation policies during software testing. We used a imperfect-debugging software reliability growth model during the resource allocation phenomenon. For above models were proved through the values given in the tables. In future some more validations are required to prove the efficiency of the model.

VI. REFERENCES

[1] M. R. Lyu, *Handbook of Software Reliability Engineering*, McGraw Hill, 1996.

[2] J. D. Musa, *Software Reliability Engineering: More Reliable Software, Faster Development and Testing*, McGraw-Hill, 1999.

[3] Y. W. Leung, "Dynamic Resource Allocation for Software Module Testing," *The Journal of Systems and Software*, Vol. 37, No. 2, pp. 129-139, May 1997.

[4] H. Ohtera, and S. Yamada, "Optimal Allocation and Control Problems for Software-Testing Resources," *IEEE Trans. on Reliability*, Vol. 39, No. 2, pp. 171-176, 1990.

[5] S. Yamada, T. Ichimori, and M. Nishiwaki, "Optimal Allocation Policies for Testing Resource Based on a Software Reliability Growth Model," *International Journal of Mathematical and Computer Modelling*, Vol. 22, pp. 295-301, 1995.

[6] M. Nishiwaki, S. Yamada, and T. Ichimori, "Testing-resource Allocation Policies based on an Optimal Software Release Problem," *Mathematica Japonica*, Vol. 43, No. 1, pp.91-97, 1996.

[7] M. R. Lyu, S. Rangarajan, and A. P. A. van Moorsel, "Optimal Allocation of Test Resources for Software Reliability Growth

Modeling in Software Development," *IEEE Trans. on Reliability*, Vol. 51, No. 2, pp. 183-192, June 2002.

[8] O. Berman, and N. Ashrafi, "Optimization Models for Reliability of Modular Software Systems," *IEEE Trans. on Software Engineering*, vol. 19, No. 11, pp. 1119-1123, Nov.1993.

[9] C. Y. Huang, J. H. Lo, J. W. Lin, C. C. Sue, and C. T. Lin, "Optimal Resource Allocation and Sensitivity Analysis for Software Modular Testing," *Proceedings of the IEEE 5th International Symposium on Multimedia Software Engineering (ISMSE 2003)*, pp. 231-238, Dec. 2003, Taichung, Taiwan.

[10] C. Y. Huang, J. H. Lo, S. Y. Kuo, and M. R. Lyu, "Optimal Allocation of Testing Resources for Modular Software Systems," *Proceedings of the IEEE 13th International Symposium on Software Reliability Engineering (ISSRE 2002)*, pp.129-138, Nov. 2002, Annapolis, Maryland.

[11] C. Y. Huang, J. H. Lo, S. Y. Kuo and M. R, "Optimal Allocation of Testing-Resource considering Cost, Reliability, and Testing-Effort," Dependable Computing, 2004. Proceedings. 10th IEEE Pacific Rim International Symposium on 3-5 March 2004, pp. 103-112.

[12] Sk.Md.Rafi ,K.Nageshwara Rao, shaheda akthar "software reliability growth model with logistic-exponential TEF and Analysis of software release policy" International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 387-399.

[13] Sk.Md.Rafi ,K.Nageshwara Rao, "SRGM with logistic-exponential Testing-effort function with change-point and Analysis of Optimal release policies based on increasing the test efficiency" International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010, 504-516.

[14] A.L. Goel and K. Okumoto, A time dependent error detection rate model for a large scale software system, *Proc. 3rd USAJapan Computer Conference*, pp. 3540, San Francisco, CA (1978).

[15] Y. Lan, and L. Leemis, (Aug. 2007) "The Logistic-Exponential Survival Distribution," *Naval Research Logistics (NRL)* volume 55, number 3, pp. 252-264.

AUTHORS PROFILE

Sk.MD.Rafi received B.Tech (comp) from Jawaharlal Nehru Technological University, M.Tech (comp) from Acharya Nagarjuna University. Pursuing PhD from Jawaharlal Nehru Technological University. Presently working as Associate. Professor in Sri Mittapalli Institute of Technology for women, affiliated to J.N.T.U, Kakinada. Published so many international papers on software reliability and quality control and software architecture recovery. My area of interest is Software Reliability, Software Architecture Recovery, Network Security, and Software Engineering.

Shaheda Akthar received Bachelor of computer science & master of computer science from Acharya Nagarjuna University, M.S (Software Systems) from BITS, Pilani, pursuing Ph.D from Acharya Nagarjuna University. Presently working as Asscociate .Professor in Sri Mittapalli College of engineering, affiliated to J.N.T.U, Kakinada. Published so many international papers on software reliability and quality control and software architecture recovery My area of interest is Software Reliability, Software Architecture Recovery, Network Security, and Software Engineering .