

Optimal Testing Effort Control for Modular Software System Incorporating the Concept of Independent and Dependent Faults: A Control Theoretic Approach

Kuldeep CHAUDHARY^a and P. C. JHA^a

^a Department of Operational Research, University of Delhi - India
Email: chaudharyitr33@gmail.com
Email: jhapc@yahoo.com

(Received August 01, 2011; in final form June 02, 2012)

Abstract. In this paper, we discuss modular software system for Software Reliability Growth Models using testing effort and study the optimal testing effort intensity for each module. The main goal is to minimize the cost of software development when budget constraint on testing expenditure is given. We discuss the evolution of faults removal dynamics in incorporating the idea of leading /independent and dependent faults in modular software system under the assumption that testing of each of the modulus is done independently. The problem is formulated as an optimal control problem and the solution to the proposed problem has been obtained by using Pontryagin Maximum Principle.

Keywords: Software reliability growth models; Testing resource allocation; Modular software system; Pontryagin maximum principle; Testing-effort expenditure.

AMS subject classifications: 90B99, 68M99, 49N90, 97M10.

1. Introduction

The software development industry has grown dramatically in scope, complexity and pervasiveness over the last few decades at an unprecedented pace. The Internet has practically transformed the world into a big computer network, where communication, information and business are constantly taking place on-line. Most industries are highly dependent on computers for their basic day-to-day functioning. Safe and reliable software operation is a significant requirement for many types of systems. Aircraft and air traffic control, medical devices, nuclear safety, petrochemical control, high speed rail, financial systems, automated manufacturing, military and nautical systems, aeronautics, space missions and appliance-type applications such as automobiles, washing machines, temperature control and telephony are highly computerized.

The cost and consequences of these systems failure can range from mildly annoying to catastrophic, with serious injury occurring or lives lost. As software assumes more responsibility for providing functionality and control in these systems, it becomes more complex and significant to the overall system performance and dependability. The software reliability assessment is important to evaluate and predict the reliability and performance of a software system. The models applicable to the assessment of software reliability are called Software Reliability Growth Models (SRGMs). Starting from the late sixties, in the approximately past 40 years a vast literature of software reliability models has been developed. Extensive research exists in modeling the failure detection and fault removal phenomenon by an Non-homogeneous poisson process (NHPP) [1,2,3]. These models have been widely studied

Corresponding Author. Email: chaudharyitr33@gmail.com

and applied on real software projects. Many concepts of software reliability modeling have been developed and models are proposed considering the various dynamic aspects of the software testing and debugging. Most of these models assume a perfect debugging environment. This means whenever an attempt is made to remove a detected fault it is removed perfectly and no new faults are generated. The earliest model in this category was due to Goel and Okumoto [4]. Some other models in this category are Yamada et al. [5]; Ohba [6, 7]; Yamada and Osaki [8]; Bittanti et al. [9] and Kapur and Garg [10] etc. Testing efforts plays a very crucial role on the testing progress. For example, at any instance of time during the testing phase testing can be made more rigorous through the additional testing efforts. A model that accommodates the effect of testing effort on the reliability growth more often proves to be more useful in the later phases of the testing as it can be used to determine the amount of additional efforts required to reach a specified reliability objective. Most of these earlier SRGMs use execution time as the unit of fault detection/removal period and either assume that the consumption rate of testing resources is constant or do not explicitly consider the testing effort and its effectiveness. A testing effort function describes the distribution or consumption pattern of testing resources (CPU time, manpower, etc) during the testing period. Putnam [11]; Yamada et al. [12, 13, 14]; Bokhari and Ahmad, [15]; Kapur et al. [16, 17]; Kuo et al. [18] and Huang et al. [19, 20, 21], proposed SRGMs describing the relationship among the testing time (calendar time), testing-effort expenditure and the number of software faults detected. Most existing SRGMs belong to exponential-type models. Kapur et al. [17] and Huang et al. [19, 20, 21] proposed S-shaped testing effort dependent SRGM based on Yamada delayed S-shaped model which also describe the leaning phenomenon of the testing team.

The optimal control theory is a branch of mathematics developed to find optimal ways to control a dynamic system. Most early developments are reported in engineering literature. Optimal control problems with state variable inequality constraints arise frequently not only in Mechanics and aerospace engineering but also in area of management sciences and economics [22, 23, 24, 25, 26, 27]. In the present paper, we applied optimal control theory to solve

a general and important problem related to modular software system for SRGMs using testing effort as a control variable.

In this paper, we discuss modular software system for SRGM using testing effort and how to allocate optimal testing effort to each module such that the total amount of testing effort expenditure is fixed. Here we assume that software firm wants to plan the testing process for the different modules with the objective to minimize the total software testing cost. Testing of each modules is done independently. We formulate an optimal control theory problem for modular software system and study the optimal testing effort expenditure rate towards each module. And, we apply Maximum Principle to solve testing effort allocation problem in modular software system.

We organized the paper as follows. In Section 2, we discuss the evolution of faults removal dynamics in modular software system under the assumption that testing of each modules is done independently and practitioner may choose independently the testing effort intensity directed to each different module. The problem is discussed and solved using Pontryagin's Maximum principle with Particular cases in Section 3.

2. Model Formulation

Testing is an important stage of Software Development Life Cycle as it provides the measure of software reliability and assists to judge the performance, safety, fault tolerance or security of the software. As most software systems are modular, it is of great importance for the management to allocate the limited testing effort recourses among the software modules in an optimal way. Testing effort can be measured by the manpower, the number of test cases, the number of CPU hours etc. There exist a number of successful S-shaped models in literature of the software reliability but we are mainly concerned here with the Kapur and Garg model [10] due to its simplicity and capability to study the failure growth rate of software. It has been observed that during testing, the team can remove some faults in the software, without these faults causing any failures. Fault which is removed consequent to failure is known as a leading fault. While removing the leading faults, some other faults are removed which may have caused failure. These faults are known as dependent faults.

We begin our analysis by describing a model for modular software system with a very few number of assumptions [19, 1]. Here are the modeling assumptions:

- 1) The fault removal process follows the Non Homogenous Poission Process (NHPP).
- 2) The software system is composed of N independent modules that are tested independently.
- 3) The total amount of testing resources expenditure available for the module testing processes is fixed and denoted by W .
- 4) The system manager has to allocate the total testing resources W to each software module to minimize the number of faults remaining in the system during the testing period.
- 5) If any of the software modules is faulty, the whole software system fails.
- 6) The detection of the error causing failure also results in detection of the remaining errors without those errors causing any failure.

With these assumptions, the number of faults detected by the current testing effort expenditure is proportional to the number of remaining faults. The evolution of fault removal dynamics for each modules can be described by the following differential equation:

$$\frac{dm_i(t)}{dt} = w_i(t) \left(p_i (a_i - m_i(t)) + q_i \frac{m_i(t)}{a_i} (a_i - m_i(t)) \right)$$

$$m_i(0) = 0, \quad \forall i = 1 \text{ to } n \quad (1)$$

Where, $m_i(t)$ is the expected mean number of faults detected in time $(0, t)$ in i^{th} module, a_i is initial number of faults in i^{th} module in the software before start of testing phase, p_i is the fault detection rate, q_i is the fault detection rate of additional faults and w_i is current testing effort expenditure rate in the time interval $(0, t]$ for i^{th} modules. $W_i(t)$ is the allocated amount of testing resources expenditure for i^{th} module and

$$\frac{d}{dt} W_i(t) = w_i(t).$$

We can evaluate the total software cost by using cost criterion, the cost of testing-effort expenditures during software development and testing phase, and the cost of correcting errors before and after release [28]. Therefore, the total software cost for each module as follows

$$C(t) = \sum_{i=1}^n \left\{ \begin{array}{l} C_{1i}(w_i) m_i(T) + C_{2i}(w_i) (m_i(\infty) - m_i(T)) \\ + C_{3i} \int_0^T w_i(t) dt \end{array} \right\} \quad (2)$$

Where, C_{1i} is the cost of testing effort (w_i) utilized for correcting an error during testing for i^{th} module, C_{2i} is the cost of testing effort (w_i) utilized for correcting an error in operational use for i^{th} module ($C_{2i} > C_{1i}$) and C_{3i} is the cost of testing per unit testing effort expenditure for i^{th} module. Note that the expected cost of fixing a fault after release is usually higher than the expected cost of fixing a fault during testing. As $t \rightarrow \infty$, the expected number of fault to be detected is $m_i(\infty)$ and it should be finite value i.e. $m_i(\infty) = A_i = \text{constant}$, then the equation (2) can be re-write as follow:

$$C(t) = \sum_{i=1}^n \left\{ \begin{array}{l} C_{1i}(w_i) m_i(T) + C_{2i}(w_i) (A_i - m_i(T)) \\ + C_{3i} \int_0^T w_i(t) dt \end{array} \right\} \quad (3)$$

Here, optimization problem is how to allocate optimal testing effort to each module such that the total amount of testing effort expenditure for all modules is fixed. We formulate an optimal control problem to allocate an optimal amount of testing effort to each modules to minimize the total software testing cost. The problem consisting of choosing the control variable $w_i(t)$ such that it minimizes the total testing cost at time t , with the requirement that there are N modules in system. Let T is the release time of software product. Now the objective can be written as

$$\min_{w_i(t)} J = \int_0^T \left(\sum_{i=1}^n \left\{ \begin{array}{l} C_{1i}(w_i) \dot{m}_i(t) \\ + C_{2i}(w_i) (A_i - \dot{m}_i(t)) + C_{3i} w_i \end{array} \right\} \right) dt \quad (4)$$

subject to (1) and the testing resources expenditure constraints for all modules

$$\int_0^T \left(\sum_{i=1}^n w_i \right) dt \leq W \quad (5)$$

Constarint (5) corresponds to the common testing effort allocation resource capacity that is allocated among all the modules. This is the only

constraint that is coupling the modules and prevents us from simply solving n times a single-module problem.

Therefore, the modules dependent testing problem is equivalent to determine the minimal cost for all modulus and testing effort rate $w_i(t) \geq 0$ and associated fault detection rate $m_i(t)$ satisfy the system equation (1).

The optimal control formulation can be written as:

$$\begin{aligned} \max_{w_i(t)} J &= \int_0^T \left(\sum_{i=1}^n \left\{ (C_{2i}(w_i) - C_{1i}(w_i)) \dot{m}_i(t) \right\} \right) dt \\ \text{s.t.} & \\ \frac{dm_i(t)}{dt} &= w_i(t) \left(\begin{array}{l} p_i(a_i - m_i(t)) \\ + q_i \frac{m_i(t)}{a_i} (a_i - m_i(t)) \end{array} \right), m_i(0) = 0 \\ \int_0^T \left(\sum_{i=1}^n w_i \right) dt &\leq W \end{aligned} \tag{P1}$$

3. Solution and Results of Proposed Optimal Control Problem

To solve the above optimal control theory problem (P1), we have introduced a new state variable $B(t)$ into the problem such that the integral constraint can be replaced by a condition in terms of $B(t)$.

Let us define $B(t) = -\int_0^t \sum_{i=1}^n w_i(\tau) d\tau$ Where, the upper limit of integration is the variable t , not the terminal time T . The derivative of this variables is $\dot{B}(t) = -\sum_{i=1}^n w_i(t)$ (Equation of motion for $B(t)$). And the terminal values of $B(t)$ in the testing period are

$$B(0) = 0 \text{ and } B(T) = -\int_0^T \sum_{i=1}^n w_i(\tau) d\tau \geq -W \tag{6}$$

Using new state variable with terminal conditions (6), we can restate optimal control problem (P1) as follow:

$$\begin{aligned} \max_{w_i(t)} J &= \int_0^T \left(\sum_{i=1}^n \left\{ (C_{2i}(w_i) - C_{1i}(w_i)) \dot{m}_i(t) \right\} \right) dt \\ \text{s.t.} & \\ \frac{dm_i(t)}{dt} &= w_i(t) \left(\left(p_i + q_i \frac{m_i(t)}{a_i} \right) (a_i - m_i(t)) \right) \\ m_i(0) &= 0 \\ \dot{B}(t) &= -\sum_{i=1}^n w_i(t), B(0) = 0, B(T) \geq -W \\ u_i^0 \leq w_i(t) \leq \bar{u}_i, 0 < u_i^0 < 1, 0 < \bar{u}_i < 1, u_i^0 < \bar{u}_i \end{aligned} \tag{P2}$$

Thus, we have an optimal control problem with n control variable($w_i(t)$) and $n+1$ state variable ($m_i(t), B(t)$) for all modules.

To solve the , we define the Hamiltonian as

$$H = \sum_{i=1}^n \left\{ (C_{2i}(w_i) - C_{1i}(w_i) + \lambda_i(t)) \dot{m}_i(t) \right\} - \mu(t) \left(\sum_{i=1}^n w_i \right) \tag{7}$$

A simple interpretation of the Hamiltonian is that it represents the sum of current cost $(C_{2i}(w_i) - C_{1i}(w_i)) \dot{m}_i(t)$ and future cost $\lambda_i(t) \dot{m}_i(t)$.

Assuming the existence of an optimal control solution, the maximum principle provides the necessary optimality conditions [29]. There exists a piecewise continuously differentiable function $\lambda_i(t)$ and $\mu(t)$ for all $t \in [0, T]$. Where $\lambda_i(t)$ and $\mu(t)$ are known as adjoint variables and the value of $\lambda_i(t)$ and $\mu(t)$ at time t describe the marginal valuation of state variables $m_i(t)$ and $B(t)$ at time t respectively (or $\lambda_i(t)$ stands for future cost incurred as one more fault introduced in the system at time t and $\mu(t)$ is the future cost of testing per unit testing effort expenditure at time t). It describes the similar behavior in optimal control theory as dual variable have in non linear programming.

The following necessary optimality conditions holds for an optimal solution:

$$\begin{cases} \frac{\partial H^*}{\partial w_i(t)} = 0 \\ \frac{d\lambda_i(t)}{dt} = -\frac{\partial H^*}{\partial m_i(t)}, \lambda_i(T) = 0 \\ \frac{d\mu(t)}{dt} = -\frac{\partial H^*}{\partial B(t)}, \\ \mu(T) \geq 0, B(T) + W \geq 0, \mu(T)(B(T) + W) = 0 \end{cases} \quad (8)$$

Where $\mu(T) \geq 0, B(T) + W \geq 0,$

$\mu(T)(B(T) + W) = 0$ are transversality condition for $B(t)$.

The Hamiltonian is independent of $B(t)$, we have $\dot{\mu} = -\partial H / \partial B = 0 \Rightarrow \mu(t) = \text{constant}$. It is clear that the multiplier associated with any integral constraint is constant over time irrespective of their nature (i.e. whether equality or inequality).

From the optimality conditions, we get the testing effort rate $w_i^*(t)$ for each module.

$$w_i^*(t) = \frac{1}{\left(\frac{\partial C_{2i}}{\partial w_i} - \frac{\partial C_{1i}}{\partial w_i} \right)} \left\{ \frac{\left(C_{3i} + \mu(t) + \frac{\partial C_{2i}}{\partial w_i} A_i \right)}{\left(\left(p_i + q_i \frac{m_i(t)}{a_i} \right) (a_i - m_i(t)) \right) - (C_{2i}(w_i) - C_{1i}(w_i) + \lambda_i(t))} \right\} \quad (9)$$

As $m_i(t) \rightarrow a_i$, value of testing effort rate increasing and becomes very large.

From equations (1), (8) and (9), the adjoint and state variables respectively can be defined as

$$\frac{dm_i(t)}{dt} = w_i^*(t) \left(\left(p_i + q_i \frac{m_i(t)}{a_i} \right) (a_i - m_i(t)) \right) \quad (10)$$

$$\frac{d\lambda_i}{dt} = \dot{\lambda}(t) = -\frac{\partial H^*}{\partial m_i} = -(C_{2i} - C_{1i} + \lambda_i) \frac{\partial \dot{m}_i}{\partial m_i} \quad (11)$$

with terminal conditions $m_i(0) = 0$ and $\lambda_i(T) = 0$.

The solution of the equation(10) is

$$m_i^*(t) = a_i \left(\frac{1 - e^{-(p_i + q_i) \int_0^t w_i^*(\tau) d\tau}}{1 + (q_i / p_i) e^{-(p_i + q_i) \int_0^t w_i^*(\tau) d\tau}} \right)$$

Define $W_i^*(t) = \int_0^t w_i^*(\tau) d\tau$ as the cumulative testing effort towards i^{th} module. The above equation takes the form

$$m_i^*(t) = a_i \left(\frac{1 - e^{-(p_i + q_i) W_i^*(t)}}{1 + \frac{q_i}{p_i} e^{-(p_i + q_i) W_i^*(t)}} \right) \quad (12)$$

Equation (12) gives the optimal number of faults detected in time $(0, t)$ in i^{th} module.

Integrating equation(11) with condition $\lambda_i(T) = 0$, we have the future cost of removing one more fault can be given as

$$\lambda_i(t) = h(T) \exp \left(-\int_t^T \left(\frac{\partial \dot{m}_i}{\partial m_i} \right) dt \right) - (C_{2i}(w_i(t)) - C_{1i}(w_i(t)))$$

Where, $h(T) = (C_{2i}(w_i(T)) - C_{1i}(w_i(T)))$

Particular Cases:

In the next section, we examine the behaviour of the proposed optimal control problem by changing the correcting costs.

3.1. When the costs are linear function of testing effort

In this section, we assume that the costs are linear function of testing effort i.e. $C_{1i}(w_i) = C_{1i} w_i$ and $C_{2i}(w_i) = C_{2i} w_i$. Where C_{2i} and C_{1i} are constant values.

Now Hamiltonian can be written as

$$H = \sum_{i=1}^n \left\{ \left((C_{2i} - C_{1i}) w_i + \lambda_i(t) \right) \dot{m}_i(t) \right\} - \mu(t) \left(\sum_{i=1}^n w_i \right) \quad (13)$$

From the optimality conditions, we have optimal value of testing effort rate $w_i^*(t)$ for each module. The optimal testing effort for each module is

$$w_i^*(t) = \frac{1}{2(C_{2i} - C_{1i})} \left\{ \frac{(C_{3i} + \mu(t) + C_{2i} A_i)}{\left(\left(p_i + q_i \frac{m_i(t)}{a_i} \right) (a_i - m_i(t)) \right)} - \lambda_i(t) \right\} \quad (14)$$

The optimal value of adjoint and state equations are

$$\lambda_i(t) = (C_{2i} - C_{1i}) \left(w_i(T) \exp \left(-\int_t^T \left(\frac{\partial \dot{m}_i}{\partial m_i} \right) dt \right) - w_i(t) \right)$$

$$m_i^*(t) = a_i \left(\frac{1 - e^{-(p_i+q_i) \int_0^t w_i^*(\tau) d\tau}}{1 + \frac{q_i}{p_i} e^{-(p_i+q_i) \int_0^t w_i^*(\tau) d\tau}} \right)$$

The physical interpretation of the above optimal policies for the testing effort allocation can be described as: Initially $m_i(t)$ increases due to S-shaped nature of fault detection function, then allocation path of testing effort expenditure will decrease. Later allocation path of testing effort expenditure will start increasing, when $m_i(t)$ slow down due to S-shapedness in fault detection rate.

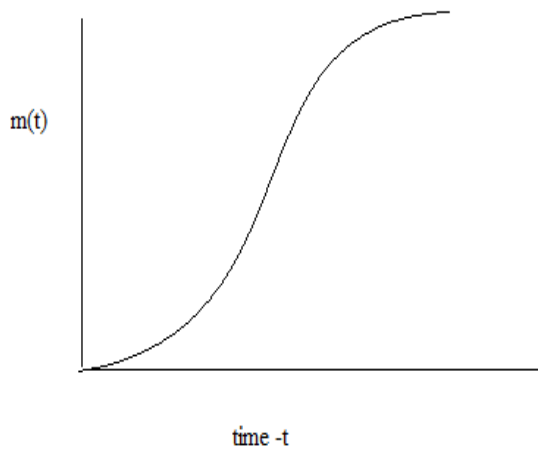


Figure 1. Optimal path of fault detection.

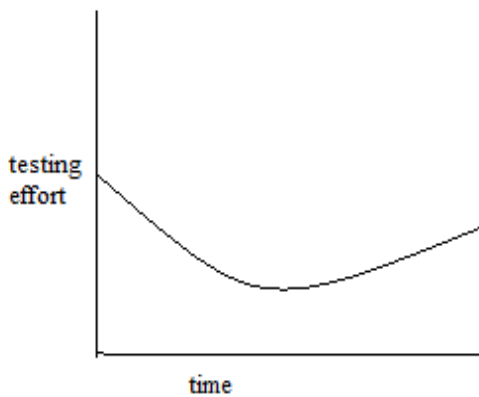


Figure 2. Optimal testing effort allocation policy.

3.2. When the costs are constant

In this section, we assume that the costs C_{1i} and C_{2i} are not directly dependent on testing effort rate i.e. $C_{1i}(w_i) = C_{1i}$ and $C_{2i}(w_i) = C_{2i}$. Where C_{2i} and C_{1i} are constant values [20].

To solve the optimal control problem, we can

define the Hamiltonian as

$$H = \sum_{i=1}^n \left\{ (C_{2i} - C_{1i} + \lambda_i(t)) \dot{m}_i(t) \right\} - \mu(t) \left(\sum_{i=1}^n w_i \right) - \sum_{i=1}^n (C_{2i} A_i - C_{3i} w_i(t)) \tag{15}$$

The $\sum_{i=1}^n C_{2i} A_i$ term in the Hamiltonian function is constant. Moreover, the fixed value $\sum_{i=1}^n C_{2i} A_i$ can be omitted in the equation.

Since the Hamiltonian is linear in control variable, the maximum principle given the following condition for the optimal testing effort $w_i^*(t)$:

$$w_i^*(t) = \begin{cases} u_i^0 & , \text{if } D_i(t) < 0 \\ \text{Singular} & , \text{if } D_i(t) = 0 \quad \forall i \\ \bar{u}_i & , \text{if } D_i(t) > 0 \end{cases} \tag{16}$$

Here,

$$D_i(t) = \{ (C_{2i} - C_{1i}) + \lambda_i(t) \} \frac{\partial \dot{m}_i}{\partial w_i} - C_{3i} - \mu(t) = H_{w_i}$$

is the coefficient of w_i in H and it is called the “switching function” and is illustrated in Fig.3. This type of optimal control is known as “bang-bang” in the terminology of optimal control theory. However, interior control is possible on an arc along which $D_i(t) = 0$. Such an arc is known as a “singular arc”. More generally, the extremal arcs ($H_{w_i} = 0$) on which the matrix $H_{w_i w_i}$ is singular arcs [27].

Now, the equation (16) can be written as

$$w_i^*(t) = \begin{cases} u_i^0 & , \text{if } (C_{2i} - C_{1i} + \lambda_i) \frac{\partial \dot{m}_i}{\partial w_i} < C_{3i} + \mu \\ \text{Singular} & , \text{if } D_i(t) = 0 \\ \bar{u}_i & , \text{if } (C_{2i} - C_{1i} + \lambda_i) \frac{\partial \dot{m}_i}{\partial w_i} > C_{3i} + \mu \end{cases} \tag{17}$$

The interpretation of the optimal policies given by equation (17) can be described as follows: If the total cost of correcting an error is less than the unit testing cost, then all testing effort expenditure rate should be minimum for all modules. On the other hand, if total cost of correcting an error is greater than the unit testing cost, then all testing effort expenditure rate should be maximum for all modules.

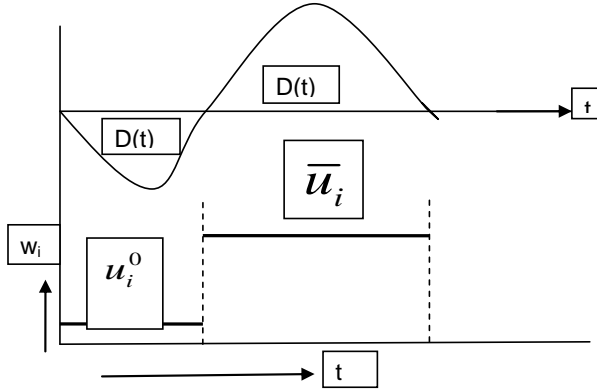


Figure 3. Optimal testing effort allocation policy.

Using equation (8) and (17), we can define the adjoint variable and state variable as follows

$$\begin{cases} \frac{d\lambda_i}{dt} = \dot{\lambda}_i(t) = \frac{\partial \dot{m}_i^*}{\partial m} (C_{2i} - C_{1i} + \lambda_i) \\ \frac{dm_i^*}{dt} = w_i^* \left(p_i + q_i \frac{m_i(t)}{a_i} \right) (a_i - m_i(t)) \end{cases} \quad (18)$$

with the transversality condition of $t=T$, $\lambda_i(T) = 0$ and initial condition $m_i(t=0) = 0$, we have

$$\lambda_i(t) = (C_{2i} - C_{1i}) \left(\exp \left(\int_t^T \frac{\partial \dot{m}_i^*}{\partial m} dt \right) - 1 \right) \quad \forall i \quad (19)$$

$$m_i^*(t) = a_i \left(\frac{1 - e^{-(p_i + q_i) \int_0^t w_i^*(\tau) d\tau}}{1 + \frac{q_i}{p_i} e^{-(p_i + q_i) \int_0^t w_i^*(\tau) d\tau}} \right) \quad \forall i \quad (20)$$

From equation (19), we show that $\lambda_i(t)$ is the negative and decreasing function. The physical interpretation of the $\lambda_i(t)$ is marginal value of faults at time t , which should be negative because increase in the number of faults will increase the correcting cost. Equation (20) shows the optimal expected mean number of faults detected in time $(0, t)$ with optimal testing effort w_i^* ($= u_i^0$ or \bar{u}_i) depend on switching function.

4. Conclusion

In this paper, we present an optimal allocation problem of testing resource expenditure for modular software system under dynamic condition. We propose an optimal control theoretic approach to optimize the resource allocation problem. It minimizes the cost of software development with a given fixed amount of testing resources expenditure.

Acknowledgments

One of the authors, Kuldeep Chaudhary, gratefully acknowledges the financial support of the Council of Scientific and Industrial Research, New Delhi, India through his Junior Research Fellowship (JRF) scheme (CSIR Award no.: 09/045(0692)/2007 EMRI) for his research work. We would also like to thank referees for their comments and suggestions which contributed to the progress of this paper invaluablely.

References

- [1] Lyu, M. R., Rangarajan, S., VanMoorsel A. P. A., Optimal allocation of testing resource for software reliability growth modeling in software development. *IEEE Trans. of Reliability*, 51(2), 183-192, (2002).
- [2] Musa, J. D., Iannino, A., and Okumoto, K., *Software Reliability, Measurement, Prediction and Application*. McGraw Hill, (1987).
- [3] Pham, H., *Software Reliability*. Springer-Verlag, (2000).
- [4] Goel, A. L., Okumoto, K., Time dependent error detection rate model for software reliability and other performance measures. *IEEE Transactions on Reliability*, R-28 (3), 206-211, (1979).
- [5] Yamada, S., Ohba, M., Osaki, S., S-shaped software reliability growth modeling for software error detection. *IEEE Transactions on Reliability*, 32(5), 475-484, (1983).
- [6] Ohba, M., Software reliability analysis models. *IBM Journal of Research and Development*, 28(4), 428-443, (1984).
- [7] Ohba, M., Inflection S-shaped software reliability growth models. Osaki, S., and

- Hatoyama, Y. (Eds.). *Stochastic Models in Reliability Theory*. Berlin. Germany. Springer-Verlag, 44-162, (1984a).
- [8] Yamada, S., Osaki, S., *Software reliability growth modeling: models and applications*. IEEE Transactions on Software Engineering, 11, 1431-1437, (1985).
- [9] Bittanti, S., Bolzern, P., Pedrotti, E., Pozzi, N., Scattolini, R., *A flexible modeling approach for software reliability growth*. Goos, G., and Harmanis, J., (Ed.). *Software Reliability Modelling and Identification*. Springer Verlag, Berlin, 101-140, (1988).
- [10] Kapur, P. K., Garg, R. B., *A Software growth model for an error removal phenomenon*. Software Eng. J., 7, 291-294, (1992).
- [11] Putnam, L. H., *A general empirical solution to the macro software sizing and estimating problem*. IEEE Transactions on Software Engineering, 4, 345-367, (1978).
- [12] Yamada, S., Ohtera, H., Narihisa, H., *Software reliability growth models with testing-effort*. IEEE Transactions on Reliability, 35, 19-23, (1986).
- [13] Yamada, S., Hishitani, J., Osaki, S., *Test-effort dependent software reliability measurement*. International Journal of Systems Science, 22 (1), 73-83, (1991).
- [14] Yamada, S., Hishitani J., Osaki, S., *Software reliability growth model with Weibull testing effort: a model and application*. IEEE Transactions on Reliability, 42, 100-105, (1993).
- [15] Bokhari, M. U., Ahmad, N., *Analysis of software reliability growth models: the case of log-logistic test-effort function*. Proceedings of the 17th IASTED International Conference on Modeling and Simulation. Montreal. Quebec. Canada, 540-545, (2006).
- [16] Kapur, P. K., Goswami, D. N., Gupta, A., *A software reliability growth model with testing effort dependent learning function for distributed systems*. International Journal of Reliability, Quality and Safety Engineering, 11(4), 365-377, (2004).
- [17] Kapur, P. K., Jha, P. C., Bardhan, A. K., *Optimal allocation of testing resource for a modular software*. Asia Pacific Journal of Operational Research, 21(3), 333-354, (2004).
- [18] Kuo, S. Y., Huang, C. Y., Lyu, M. R., *Framework for modeling software reliability using various testing-efforts and fault-detection rates*. IEEE Transactions on Reliability, 50 (3), 310-320, (2001).
- [19] Huang, C. Y., Kuo, S. Y., *Analysis of incorporating logistic testing effort function into software reliability modeling*. IEEE Transactions on Reliability, 51 (3), 261-270, (2002).
- [20] Huang, C. Y., Lyu, M. R., *Optimal release time for software systems considering cost, testing effort and test efficiency*. IEEE Transactions On Reliability, 54 (4), 583 - 591, (2005).
- [21] Huang, C. Y., Kuo, S. Y., Lyu, M. R., *An assessment of testing-effort dependent software reliability growth models*. IEEE Transactions on Reliability, 56 (2), 198-211, (2007).
- [22] Bryson, A. E., Ho, Y. C., *Applied Optimal Control*. Blaisdell Publishing Co. Wal-tham Mass, (1969).
- [23] Feichtinger, G., (Ed.). *Optimal Control Theory and Economics Analysis*. Vol. 3. North-Holland. Amsterdam, (1988).
- [24] Hull, D., *Optimal Control Theory for Applications*. Springer-Verlag, (2003).
- [25] Sethi, S. P., *Dynamic optimal control models in advertising: a survey*. SIAM Review, 19(4) , 685-725, (1977).
- [26] Sethi, S. P., Thompson, G. L., *Optimal Control Theory: Applications to Management Science and Economics*. Kluwer Academic Publishers, Dordrecht, (2000).
- [27] Sethi, S. P., *Optimal advertising for the Nerlove-Arrow model under Budget constraint*. Operational Research Quarterly, 28(3), 683-693, (1977).

- [28] Yamada, S., Ichimori, T., Nishiwaki, M., Optimal allocation policies for testing resource based on a software reliability growth model. *Int. Journal of Mathematical Modelling*, 22, 295-301, (1995).
- [29] Seierstad, A., Sydsaeter, K., Optimal control theory with economic applications. North-Holland, Amsterdam, (1987).

Kuldeep Chaudhary received the M.Sc. degree in Applied Mathematics from the Indian Institute of Technology Roorkee, Roorkee, India. He is currently a Senior Research Fellow in the Department of Operational Research, University of Delhi, Delhi. His research interests include Modelling, Optimal Control and Optimization in

Marketing and Software Reliability.

Dr. P.C. Jha obtained his Ph. D., M. Phil and M.Sc. degrees in Operational Research from University of Delhi in 2004, 1988 and 1986 respectively. He is the Associate in the Department of Operational Research, University of Delhi. He has published more than 45 research papers in the areas of software reliability, Marketing and Optimization in peer reviewed journals and edited books. He has guided master's projects, MBA and M. Phil. Dissertations and Ph.D. thesis in Operational Research. His research interests include modelling and optimization in software reliability and Marketing.