Applied Mathematical Modelling 34 (2010) 3560–3570

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Sensitivity analysis of release time of software reliability models incorporating testing effort with multiple change-points

Xiang Li *, Min Xie, Szu Hui Ng

Department of Industrial and Systems Engineering, National University of Singapore, Singapore 119260, Singapore

ARTICLE INFO

Article history:
Received 13 July 2009
Received in revised form 23 February 2010
Accepted 10 March 2010
Available online 18 March 2010

Keywords:
Software reliability
Release time
Sensitivity analysis
Testing effort
Multiple change-points

ABSTRACT

To accurately model software failure process with software reliability growth models, incorporating testing effort has shown to be important. In fact, testing effort allocation is also a difficult issue, and it directly affects the software release time when a reliability criteria has to be met. However, with an increasing number of parameters involved in these models, the uncertainty of parameters estimated from the failure data could greatly affect the decision. Hence, it is of importance to study the impact of these model parameters. In this paper, sensitivity of the software release time is investigated through various methods, including one-factor-at-a-time approach, design of experiments and global sensitivity analysis. It is shown that the results from the first two methods may not be accurate enough for the case of complex nonlinear model. Global sensitivity analysis performs better due to the consideration of the global parameter space. The limitations of different approaches are also discussed. Finally, to avoid further excessive adjustment of software release time, interval estimation is recommended for use and it can be obtained based on the results from global sensitivity analysis.

1. Introduction

During the last three decades, a large number of models have been proposed for software failure process [1–8]. In the recent years, incorporating testing effort into software reliability growth models (SRGMs) has received a lot of attention, probably because testing effort is an essential process parameter for management. Huang et al. [9] showed that logistic testing effort function can be directly incorporated into both exponential-type and S-type non-homogeneous Poisson process (NHPP) models and the proposed models were also discussed under both ideal and imperfect debugging situations. Kapur et al. [10] discussed the optimization problem of allocating testing resources by using marginal testing effort function (MTEF). Later, Kapur et al. [11] studied the testing effort dependent learning process and faults were classified into two types by the amount of testing effort needed to remove them. In addition, some research incorporated change-point analysis in their models as the testing effort consumption may not be smooth over time [12–14]. Specifically, Lin and Huang [14] incorporated multiple change-points into the flexible Weibull-type time dependent testing effort function. The proposed model seems to be more realistic and therefore it is selected in this paper.

As constructing model is not the end, to guide project managers to decide when to release the software is a typical application of the model. The optimal release time problem considering testing effort was also discussed [14–17]. However, most of the research assumes that parameters of the proposed models are known. In fact, there always exist estimation errors as parameters in testing effort function and SRGMs are generally estimated by least square estimation (LSE) method and

* Corresponding author. Tel.: +65 6516 2208.
E-mail address: lixiang@nus.edu.sg (X. Li).

0307-904X/$ - see front matter
doi:10.1016/j.apm.2010.03.006

Notation

$R(x\|t)$	conditional software reliability
R_0	reliability requirement
$W(t)$	cumulative testing effort consumption at time t
$w(t)$	$dW(t)/dt$: current testing effort function at time t
α	total amount of testing effort eventually consumed
β_i	scale parameter in the Weibull testing effort function and $i = 1, 2, \ldots, m + 1$
γ_i	shape parameter in the Weibull testing effort function and $i = 1, 2, \ldots, m + 1$
m	number of change-points in testing effort function
τ_i	time where the change-point occurs in testing effort function and $i = 1, 2, \ldots, m$
$m(t)$	mean value function
a	expected number of initial faults in the software
r	fault detection rate per unit testing effort
$\lambda(t)$	failure intensity function
t_i	a collection of ordered constant testing times and $i = 1, 2, \ldots, n$
w_i	a collection of current testing effort expenditures at testing time t_i and $i = 1, 2, \ldots, n$
n_i	the number of failures in each time interval and $i = 1, 2, \ldots, n$
T	optimal software release time
f	a deterministic function
E_i	main effect of parameter θ_i in DOE and $i = 1, 2, \ldots, 2m + 5$
θ_i	parameter involved in sensitivity analysis and $i = 1, 2, \ldots, 2m + 5$
$V(T)$	the total variance of T used in global sensitivity analysis
V_i	$V[E(T\|\theta_i)]$: measures the main effect of the parameter θ_i and $i = 1, 2, \ldots, 2m + 5$
V_{ij}	measures the second-order interaction effect and $1 \leqslant i \leqslant j \leqslant 2m + 1$
$V_{1,2,\ldots,2m+5}$	measures the $(2m + 5)$th-order interaction effect
S_i	first-order sensitivity index for θ_i and $i = 1, 2, \ldots, 2m + 5$
A,B,C_i	matrix of random numbers and $i = 1, 2, \ldots, 2m + 5$
N	a base sample which can vary from a few hundreds to a few thousands
T_A, T_B, T_{C_i}	defined as $f(A)$, $f(B)$ and $f(C_i)$ respectively
f_0	mean of T_A
θ	a given confidence level
T_L,T_U	lower bound and upper bound of optimal release time

maximum likelihood estimation (MLE) method respectively. It is necessary to conduct the sensitivity analysis to determine which parameter may have significant influence to the software release time. This is even more important when there are an increasing number of parameters involved in the model, such as the model proposed by Lin and Huang [14].

Sensitivity analysis can be used to determine how sensitive the software release time is. It helps to find parameters that could significantly affect the solution to the release time. By showing how the software release time reacts against the changes in parameter values, the model is also evaluated and validated. In this paper, sensitivity of the software release time is studied and different approaches are used, including one-factor-at-a-time approach, design of experiments and global sensitivity analysis.

After the sensitivity analysis, significant parameters can be determined and they should be estimated precisely. However, it may not be possible due to the limited amount of information available. Thus, conservative estimation of release time is needed to avoid releasing the software too optimistically [18]. To this end, interval estimation is recommended for use and the simulation results from global sensitivity analysis can just help in this.

The rest of the paper is organized as follows. Section 2 introduces the general model incorporating testing effort and formulates the software release time problem. Section 3 discusses procedures when using different approaches to sensitivity analysis. In Section 4, an application example is given and some interesting results are obtained. In Section 5, limitations of different approaches are highlighted. The interval estimation of optimal release time is discussed in Section 6 and it can be seen that results from global sensitivity analysis are very helpful in this. Concluding remarks are made in Section 7.

2. General model incorporating testing effort

To accurately model software failure process with SRGMs, incorporating testing effort has shown to be important and it has received a lot of attention. According to Lin and Huang [14], multiple change points should be considered due to the changing testing efforts in reality. This model is adopted here as it is shown to be a general one with fairly accurate prediction capability [14]. Specifically, with the consideration of arbitrary number of change points, the cumulative testing effort function is given by