

考虑测试工作量与覆盖率的软件可靠性模型^{*}

李海峰¹, 王栓奇², 刘畅¹, 郑军¹, 李震³

¹(中国航空综合技术研究所 质量工程中心, 北京 100028)

²(北京航空航天大学 可靠性与系统工程学院, 北京 100191)

³(江苏科技大学 电子信息学院, 江苏 镇江 212003)

通讯作者: 李海峰, E-mail: lihailfeng@dse.buaa.edu.cn

摘要: 为了进一步提升现有非齐次泊松过程类软件可靠性增长模型的拟合与预计精度, 首先, 提出一个同时考虑测试工作量与测试覆盖率的NHPP类软件可靠性建模框架. 在此基础上, 将变形S型测试工作量函数(IS-TEF)以及Logistic测试覆盖率函数(LO-TCF)带入该建模框架, 建立了一个新的软件可靠性增长模型, 即IS-LO-SRGM. 同时, 还对利用该框架进行建模过程中的两个重要问题进行了描述与分析, 即如何确定具体的TEF和TCF以及模型参数估计. 然后, 在两组真实的失效数据集上, 利用该建模框架建立了最为合适的增长模型, 即IS-LO-SRGM, 并将该模型与8种经典NHPP模型进行对比. 实例验证结果表明, 所提出的IS-LO-SRGM模型具有最为优秀的拟合与预计性能, 从而证明新建模框架的有效性和实用性. 最后, 对不完备排错情况进行了初步的讨论与建模分析.

关键词: 软件可靠性建模; 测试覆盖率; 测试工作量; 非齐次泊松过程

中图分类号: TP311 文献标识码: A

中文引用格式: 李海峰, 王栓奇, 刘畅, 郑军, 李震. 考虑测试工作量与覆盖率的软件可靠性模型. 软件学报, 2013, 24(4): 749-760. <http://www.jos.org.cn/1000-9825/4257.htm>

英文引用格式: Li HF, Wang SQ, Liu C, Zheng J, Li Z. Software reliability model considering both testing effort and testing coverage. Ruanjian Xuebao/Journal of Software, 2013, 24(4): 749-760 (in Chinese). <http://www.jos.org.cn/1000-9825/4257.htm>

Software Reliability Model Considering both Testing Effort and Testing Coverage

LI Hai-Feng¹, WANG Shuan-Qi², LIU Chang¹, ZHENG Jun¹, LI Zhen³

¹(Quality Engineering Center, China Aero-Polytechnology Establishment, Beijing 100028, China)

²(School of Reliability and Systems Engineering, BeiHang University, Beijing 100191, China)

³(School of Electronics Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Corresponding author: LI Hai-Feng, E-mail: lihailfeng@dse.buaa.edu.cn

Abstract: To further improve the fitting and prediction performance of the non-homogeneous Poisson process (MHPP) software reliability growth models (SRGMs), this paper, as the extension work of the NHPP software reliability modeling framework which considers the TEF, will discuss how to integrate both TEF and TCF into the traditional NHPP software reliability modeling process. This is done in order to capture the integrated effect of testing effort and testing coverage on reliability estimation. First, a comprehensive modeling framework for incorporating the TEF and TCF together into the NHPP SRGMs is proposed. Recur to this framework, a new NHPP SRGM (named IS-LO-SRGM) with both the IS-TEF and logistic TCF (LO-TCF) is proposed. Meanwhile, two issues of this proposed framework are discussed respectively (i.e. how to select the most appropriate TEF and TCF for modeling and the parameter estimation). Then, two case studies on two real failure data-sets are presented. The experimental results show that the IS-LO-SRGM nearly yields the best fitting and prediction results compared with the other comparison NHPP SRGMs for two data-sets. Thus, the applicability and effectiveness of this modeling framework are validated. Finally, the imperfect debugging phenomenon is also considered in the modeling framework for a further discussion.

Key words: software reliability modeling; testing coverage; testing effort; non-homogeneous Poisson process

* 收稿时间: 2011-11-23; 定稿时间: 2012-04-26

作为最重要的面向用户的软件质量特征,软件可靠性水平已成为软件发布时用户最为关心的验证指标.软件可靠性通常可以被定义为在特定时间内,特定使用环境中,软件不发生失效的概率^[1-3].因此,作为软件可靠性定量评估与预计的重要手段,近些年来,许多基于时间域的软件可靠性增长模型(*software reliability growth model*,简称 SRGM)被提了出来,并被成功地应用于各种类型安全关键软件的开发过程^[4].这些软件可靠性增长模型根据失效数据来定量描述测试阶段的或运行阶段的失效行为.非齐次泊松过程(*non-homogeneous Poisson process*,简称 NHPP)类模型由于具有易理解与易使用等优良特性,已成为目前研究最为广泛的一类 SRGMs^[5,6].已有研究成果表明^[7-11],若能够在建模过程中考虑某些重要测试因素对于软件可靠性的影响效果,使其可以更为准确地描述实际测试过程中失效的探测与排除过程,将有可能进一步提升模型的拟合与预计精度.测试工作量(*testing effort*,简称 TE)以及测试覆盖率(*testing coverage*,简称 TC)是这些因素中最重要的两个.

测试工作量可定量刻画软件测试过程中所消耗的各类测试资源,常通过人时、测试用例数目、CPU 时间等测试信息来度量^[10,11].因为最终的软件可靠性水平与用于缺陷探测和修正的测试工作量高度相关^[12],所以测试工作量随测试时间的变化情况对软件可靠性增长曲线的形状有着更为显著的影响^[11-13].早期的 SRGMs 或者不考虑测试工作量或者假设测试工作量随时间的消耗率是一个常值^[1,4],这是不合理的.因此,如何准确地定量描述测试工作量随测试时间的累积消耗情况(即测试工作量函数, *testing effort function*,简称 TEF),逐渐引起研究人员的热切关注^[11-30].现有测试工作量函数主要有如下几种:威布尔函数^[14]、指数威布尔函数^[15]、Burr-X 函数^[17]、Logistic 函数^[11]、广义 Logistic 函数 1 和广义 Logistic 函数 2^[19]、变形 S 型函数(*IS-TEF*)^[30].

作为测试充分性和测试效率的有效度量指标,测试覆盖率能够帮助软件开发人员合理评估被测软件的质量.根据覆盖信息的不同,测试覆盖率可以被若干种测试信息所度量,例如语句覆盖、分支覆盖、C-use 覆盖以及 P-use 覆盖等^[31].有研究表明,测试覆盖率与软件可靠性以及缺陷覆盖情况之间有着很明显的相关性^[9,31-38].因此,已有若干测试覆盖率函数(*testing coverage function*,简称 TCF)被提了出来,用于定量描述测试过程中软件测试覆盖率随测试时间的变化情况.例如指数型、威布尔型、S 型以及 Log-Logistic 函数^[36]、Pham-Zhang 函数^[37]、Logistic 函数(*LO-TCF*)^[38].

综上,考虑测试工作量的 NHPP 模型(*TEF-SRGM*)或者考虑测试覆盖率的 NHPP 模型(*TCF-SRGM*)均可不同程度地改进传统 NHPP 软件可靠性模型的拟合或预计精度.因此,若能将测试工作量函数与测试覆盖率函数同时引入 NHPP 模型的建模过程,则有可能进一步提升现有 NHPP 模型的拟合与预计精度.

基于上述分析,本文将针对考虑测试工作量和测试覆盖率的 NHPP 软件可靠性建模方法展开理论与实例研究.首先,介绍考虑测试工作量或测试覆盖率的 NHPP 建模方法,同时介绍 IS-TEF 与 LO-TCF 以及相应的 NHPP 模型(*IS-TEFM* 与 *LO-TCFM*).在此基础上,提出同时考虑测试工作量与测试覆盖率的 NHPP 软件可靠性建模框架.借此框架,将 IS-TEF 和 LO-TCF 引入 NHPP 软件可靠性建模过程,进而提出考虑 IS-TEF 和 LO-TCF 的 NHPP 模型(*IS-LO-SRGM*).此外,还对新建模框架的两个相关问题(框架中 TEF 和 TCF 的确定以及模的参数估计问题)进行讨论与分析.在两组经典的真实失效数据集上,利用本文提出的建模框架进行建模,并将所提出的 IS-LO-SRGM 分别与若干典型的传统 NHPP 模型(如 GO 模型)、TCF-SRGMs(如 LO-TCFM)以及 TEF-SRGMs(如 IS-TEFM)进行实例验证对比,以验证 IS-LO-SRGM 的模型拟合与预计能力,进而证明本文提出的考虑测试工作量与测试覆盖率 NHPP 建模框架的有效性与其可行性.最后,在不完美排错的假设下,进一步提出考虑 TEF、TCF 及不完美排错的 NHPP 软件可靠性建模框架.

1 考虑测试工作量或测试覆盖率的 NHPP 建模方法

1.1 考虑测试工作量的 NHPP 类软件可靠性建模框架^[7,10]

考虑测试工作量的 NHPP 软件可靠性模型的基本假设如下:

- 1) 软件失效的探测与排除过程随时间服从 NHPP 过程;
- 2) 当前失效是由软件中残存的缺陷引发的;
- 3) 在时间间隔 $[t, t+\Delta t]$ 内,所探测到的缺陷数均值和当前测试工作量消耗率的比值,与软件残存缺陷数均

值成正比,这个比值即为当前的缺陷探测率.它可以是一个常数 r ,也可以是时间相关的函数 $r(t)$;

- 4) 缺陷一旦被发现,立即被排除且不引入新的缺陷.也就是说,软件中的缺陷总数是一个常值 N .
根据上述几条假设,即可得到如公式(1)所示的考虑测试工作量的 NHPP 软件可靠性建模框架:

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

其中, $m(t)$ 表示时间间隔 $[0, t]$ 内的探测缺陷数的期望均值函数; $w(t)$ 表示测试工作量消耗速率函数,它是测试工作量函数(记为 $W(t)$)对测试时间 t 的导数,即 $w(t) = dW(t)/dt$.在边界条件为 $m(0)=0, W(0)=0$ 时,将不同形式的测试工作量函数 $W(t)$ 代入上述微分方程并求解,即可得到各种不同的 NHPP 软件可靠性模型.

1.2 考虑测试覆盖率的NHPP类软件可靠性建模框架^[37,38]

考虑测试覆盖率的 NHPP 软件可靠性模型的基本假设如下:

- 1) 软件失效的探测与排除过程随时间服从 NHPP 过程;
- 2) 当前失效是由软件中残存的缺陷引发的;
- 3) 在时间间隔 $[t, t+\Delta t]$ 内,由当前测试工作量所探测到的缺陷数均值与软件中的残存缺陷数均值成正比,这个比值即为当前的缺陷探测率,它是一个与时间相关的函数 $r(t)$,由 t 时刻的测试覆盖增长速率函数 $C'(t)$ ($C(t)$ 表示测试覆盖率函数)与当前未覆盖代码比例 $1-C(t)$ 的比值来表示;
- 4) 缺陷一旦被发现,立即被排除且不引入新的缺陷.也就是说,软件中的缺陷总数是一个常值 N .

据上述几条假设,即可得到如公式(2)所示的考虑测试覆盖率的 NHPP 软件可靠性建模框架:

$$\frac{dm(t)}{dt} = \frac{C'(t)}{1-C(t)}(N - m(t)) \quad (2)$$

在边界条件为 $m(0)=0$ 时,将不同形式的 $C(t)$ 代入公式(2)并求解,即可得到各种不同的 NHPP 软件可靠性模型.

1.3 变形S型测试工作量函数与Logistic测试覆盖率函数及相应的NHPP模型^[30,38]

由于软件结构特征和学习因素的综合影响,在实际的测试过程中,测试工作量或测试覆盖率的增长速率随测试时间经常会呈现出一种先增后减的 S 型增长趋势^[11,38].为更加准确地描述这种 S 型增长趋势,在之前的研究工作中,我们先后提出了 IS-TEF 以及 LO-TCF,并在此基础上分别构建了 IS-TEFM 和 LO-TCFM.

1.3.1 IS-TEF 与 IS-TEFM^[30]

假设利用变形 S 型函数来描述测试工作量随时间的变化情况,则在 $[0, t]$ 内消耗的累积测试工作量为

$$W(t) = W_{\max} \frac{1 - e^{-bt}}{1 + \psi e^{-bt}} \quad (3)$$

其中, b 表示测试工作量的消耗率; ψ 是一个常量; W_{\max} 表示最终可消耗的总测试工作量,即 $W_{\max} = W(\infty)$.

将公式(3)代入公式(1),假定缺陷探测率为一常数 r ,对公式(1)中的微分方程求解,即可得 IS-TEFM 如下:

$$m(t) = N \left\{ 1 - \exp \left[-r W_{\max} \frac{1 - \exp(-bt)}{1 + \psi \exp(-bt)} \right] \right\} \quad (4)$$

1.3.2 LO-TCF 与 LO-TCFM^[38]

假设利用 Logistic 函数来描述测试覆盖率随时间的变化情况,则在 $[0, t]$ 内所达到的累积测试覆盖率为

$$C(t) = \frac{C_{\max}}{1 + A \exp(-\alpha t)} \quad (5)$$

其中, $C_{\max} = C(\infty)$, A 为常量, α 表示测试覆盖率增长速率参数.一般可假设 $C_{\max} = 1$,则公式(5)可化为式(6):

$$C(t) = \frac{1}{1 + A \exp(-\alpha t)} \quad (6)$$

将公式(6)代入公式(2),对公式(2)中的微分方程进行求解,即可得 LO-TCFM 如下:

$$m(t) = N \left[1 - \exp \left(- \frac{\alpha t}{1 + A \exp(-\alpha t)} \right) \right] \quad (7)$$

1.4 相关NHPP软件可靠性模型汇总

本节将若干经典的传统 NHPP 模型(模型类型记为 A)、典型的 TEF-SRGM(模型类型记为 B)、典型的 TCF-SRGM(模型类型记为 C)列举于表 1,这些模型均是同类研究中最典型或最新的研究成果。

Table 1 An overview of the related existing NHPP SRGMs
表 1 相关 NHPP 模型一览

SRGMs	Mean value function ($m(t)$)	Type	Comments
GO ^[11]	$N[1 - \exp(-rt)]$	A	—
IS ^[30]	$N(1 - e^{-rt}) / (1 + \psi e^{-rt})$	A	—
LL-TCFM ^[36]	$N \frac{(\lambda t)^k}{1 + (\lambda t)^k}$	C	$C(t) = \frac{(\lambda t)^k}{1 + (\lambda t)^k}$
LO-TCFM ^[38]	$N \left[1 - \exp \left(- \frac{\alpha t}{1 + A \exp(-\alpha t)} \right) \right]$	C	$C(t)$ is shown in Eq.(6)
PZ-TCFM ^[37]	$N \left(1 + \beta t - \frac{bt-1}{\exp(bt)} - \frac{N\beta(1+bt)}{b \exp(bt+1)} \left[\ln(bt+1) + \sum_{i=0}^{\infty} \frac{(1+bt)^{i+1} - 1}{(i+1)!(i+1)} \right] \right)$	C	$C(t) = 1 - (1+bt)e^{-bt}$
LO-TEFM ^[11]	$N \left(1 - \exp \left(-r \left(\frac{W_{\max}}{1 + A \exp(-\alpha t)} - \frac{W_{\max}}{1 + A} \right) \right) \right)$	B	$W(t) = \frac{W_{\max}}{1 + A \exp(-\alpha t)}$
EW-TEFM ^[15]	$N(1 - \exp(-rW_{\max}(1 - \exp(-bt^m))^{\theta}))$	B	$W(t) = W_{\max}(1 - \exp(-bt^m))^{\theta}$
IS-TEFM ^[30]	$N \left\{ 1 - \exp \left[-rW_{\max} \frac{1 - \exp(-bt)}{1 + \psi \exp(-bt)} \right] \right\}$	B	$W(t)$ is shown in Eq.(3)

2 考虑测试工作量与测试覆盖率的 NHPP 软件可靠性建模方法

2.1 考虑测试工作量与测试覆盖率的 NHPP 软件可靠性建模框架

2.1.1 基本假设

- 1) 软件失效的探测与排除过程随时间服从 NHPP 过程;
- 2) 当前失效是由软件中残存的缺陷引发的;
- 3) 在时间间隔 $[t, t+\Delta t]$ 内,所探测到的缺陷数均值与当前的测试工作量消耗率的比值,与软件残存缺陷数均值成正比,这个比值即为当前的缺陷探测率函数 $r(t)$,可由 t 时刻的测试覆盖增长速率函数 $C'(t)$ 与当前未覆盖代码比例 $1-C(t)$ 的比值来表示;
- 4) 缺陷一旦被发现,立即被排除且不引入新的缺陷.也就是说,软件中的缺陷总数是一个常值 N .

值得注意的是,假设 3)和假设 4)分别描述了测试工作量和测试覆盖率对可靠性评估与预计精度的影响效果,从而显式地将测试工作量与测试覆盖率引入 NHPP 软件可靠性的建模过程,这也是后续建模的核心思想。

2.1.2 建模框架

根据上述建模假设,可得到同时考虑测试工作量和测试覆盖率的 NHPP 建模框架如式(8)所示:

$$\frac{dm(t)}{dt} \frac{1}{w(t)} = \frac{C'(t)}{1-C(t)} [N - m(t)] \quad (8)$$

根据公式(8)中的等式可以看出:

- 1) 时刻 t 的失效强度值(即 $\frac{dm(t)}{dt}$)不仅依赖于当前测试覆盖增长速率函数与当前未覆盖代码比例的比值,而且还依赖于当前时刻的测试工作量消耗率;
- 2) 明显地,第 1.1 节中的公式(1)以及第 1.2 节中的公式(2)均是公式(8)的特例;
- 3) 通过公式(8)中的通用建模框架可以建立一系列的 NHPP 模型,会涵盖许多已有的典型 NHPP 模型;

4) 假定 $B(t) = \int_0^t \frac{C'(W(\tau))}{1-C(W(\tau))} d\tau$, 由公式(8)可获得考虑测试工作量和测试覆盖率的形式化 NHPP 模型

如下:

$$m(t) = Ne^{-B(W(t))} \int_0^t e^{B(W(\tau))} \frac{C'(W(\tau))}{1-C(W(\tau))} d\tau \quad (9)$$

2.1.3 相关讨论

问题 1:模型参数估计过程.

由公式(8)所建立的 NHPP 模型,一般情况下都是多参数模型(即模型参数大于 3 个).对于该类多参数的 NHPP 模型,文献[9]中推荐整体估计过程,亦即将所有的模型参数视为一个整体,根据测试过程中采集到的失效数据信息来对所有参数同时进行估计.文献[9]认为,这种估计过程可有效描述参数之间的依赖性.现有的考虑测试覆盖率的建模研究采用的均是整体估计过程.而考虑测试工作量的建模研究则采用逐步式估计过程^[7,15],即,首先将模型中与测试工作量函数相关的参数通过测试工作量数据信息(例如测试用例数)来进行估计,然后将剩余的模型参数通过失效数据信息来进行估计.综合上述两种方法,本文推荐如下逐步式估计过程:

第 1 步:公式(8)中与测试工作量函数相关的参数(例如公式(3)中的 W_{\max} , ψ 以及 b)应由测试过程中所采集到的测试工作量数据集来进行估计,然后将这些参数的估计值代入公式(8);

第 2 步:此时,公式(8)中与测试覆盖率函数相关的参数(例如公式(6)中的 α 和 A)以及传统 NHPP 模型参数(例如公式(8)中的 N)则可以由测试过程中所采集到的失效数据信息统一进行估计.

需要说明的是:1) 如果测试过程中很难采集准确的测试工作量数据集,那么也可以采用文献[9]推荐的整体估计过程来对公式(8)中的所有模型参数进行估计,但这样所获得的与测试工作量函数相关的模型参数的估计值的可信程度比利用测试工作量数据信息所获得的参数估计值要差;2) 虽然多参数 NHPP 模型的参数求解过程会比较繁复,但由此所带来的额外计算量完全可以由自动化的工具来实现.

问题 2:确定公式(8)中的 TCF 和 TEF.

目前已经存在着多种不同形式的 TCF 或者 TEF,如何根据具体的失效数据集(含测试工作量数据集)来确定公式(8)中最为合适的 TCF 和 TEF 是一个很重要的问题.最理想的办法当然是将每种 TCF 和 TEF 两两组合分别代入公式(8)中,得到一系列的新模型,再根据失效数据集对每个模型的拟合能力进行比较以选取最为优秀的模型.但这种理想方法所需要的工作量太大,所以本文推荐如下确定 TCF 和 TEF 的方法:

首先,根据测试工作量数据集对每个备选 TEF 的拟合能力进行比较,选出最优的 TEF 作为公式(8)中的 TCF;然后,根据失效数据集对每个备选的考虑 TCF 的 NHPP 模型的拟合能力进行比较,将最优模型中的 TCF 作为公式(8)中的 TCF.需要说明的是:1) 在确定最优 TEF 的过程中所获得的 TEF 参数估计值可直接作为公式(8)中相应参数的估计值;2) 但是,确定最优 TCF 过程中所获得的 TCF 参数估计值却不能作为公式(8)中相应参数的估计值.因为问题 1 中已明确,公式(8)中 TCF 相关参数要与 NHPP 类软件可靠性模型参数视为整体一起进行估计.

2.2 考虑 IS-TEF 和 LO-TCF 的 NHPP 软件可靠性增长模型

本节将提出一个新的考虑测试工作量和测试覆盖率的 NHPP 软件可靠性模型.

首先,将公式(6)中的 LO-TCF 代入公式(8)后可得如下微分方程:

$$\frac{dm(t)}{dt} \frac{1}{w(t)} = \frac{\alpha(N-m(t))}{1+A \exp(-\alpha t)} \quad (10)$$

然后,再将公式(3)中的 IS-TEF 代入公式(10),并在边界条件 $m(0)=0, W(0)=0$ 下,对化简后的微分方程进行求解,即可获得考虑 IS-TEF 和 LO-TCF 的多因素 NHPP 模型(IS-LO-SRGM),如式(11)所示:

$$m(t) = N \left[1 - \exp \left(- \frac{\alpha W_{\max} \frac{1 - e^{-bt}}{1 + \psi e^{-bt}}}{1 + A \exp \left(-\alpha W_{\max} \frac{1 - e^{-bt}}{1 + \psi e^{-bt}} \right)} \right) \right] \quad (11)$$

由第 2.1.3 节可知,实际上,公式(11)中 IS-TEF 的 3 个参数(即 W_{\max} , ψ 和 b)的估计值可以根据测试工作量数据集来获得,而真正需要根据失效数据集来进行估计的参数只有 3 个,即 N, A 和 α .

3 实例验证

3.1 实例概述

(1) 对比 NHPP 模型:选取表 1 列举的 8 个已有的相关 NHPP 模型作为对比模型;

(2) 失效数据集:选择两组真实的且包含测试工作量数据信息的失效数据集 DS1^[39]和 DS2^[40],受篇幅所限,这两组失效数据集的具体内容请参见相应文献;

(3) 本实例验证过程可分为如下两个步骤:

步骤 1. 首先,应用本文提出的建模框架来分别建立适于两组真实失效数据集的考虑测试工作量和测试覆盖率的 NHPP 模型;

步骤 2. 将步骤 1 中建立的 NHPP 模型分别与表 1 中的相关典型 NHPP 模型进行实例对比研究;

(4) 模型参数估计方法:最大似然估计(maximum likelihood estimation,简称 MLE)及最小二乘估计(least squares estimation,简称 LSE).限于篇幅,在步骤 1 中采用 LSE 方法,在步骤 2 中采用 LSE 方法和 MLE 方法.

(5) 模型拟合性能比较准则:根据参数估计方法的不同,可选用如下两种模型拟合性能比较准则,即均方误差(MSE,见公式(12))和 Akaike 信息准则(AIC,见公式(13)):

1) 均方误差(MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (m(t_i) - m_i)^2 \quad (12)$$

其中, m_i 为真实的探测缺陷数, $m(t_i)$ 为模型计算出的对应 m_i 的拟合值.当参数估计方法为 LSE 时,对应的拟合性能比较准则为 MSE.MSE 值越小,则模型拟合效果越好.

2) Akaike 信息准则(AIC)

$$AIC = -2 \times MLL + 2 \times N_p \quad (13)$$

其中, MLL 表示最大似然函数值的 Log 值, N_p 表示 NHPP 模型的模型参数个数.当参数估计方法为 MLE 时,对应的拟合性能比较准则为 AIC.AIC 信息准则的基本思想是:对参数越多的模型,在计算其拟合能力时给予越高的惩罚值,以降低参数自由度对于模型拟合能力的影响效果.因此,在所建立的模型参数较多时,应采用 MLE 法和 AIC 准则来增强模型拟合性能对比结果的可信性.同样地,AIC 值越小,模型拟合效果越好.

(6) 模型预计性能比较准则:

选择相对误差值(RE 值,见公式(14))来评价 NHPP 模型对未来软件失效行为(即缺陷数均值)的预计效果.

$$RE = (m(t_i) - m_i) / m_i \quad (14)$$

假设测试结束的时刻为 t_n ,此时共观测到 m_n 个失效.使用截止到时刻 $t_e (t_e \leq t_n)$ 的失效数据来估计模型 $m_e(t)$ 的参数值.将参数估计值代入模型 $m_e(t)$,从而获得 t_n 时刻的失效数预计值为 $m_e(t_n)$.然后将预计值 $m_e(t_n)$ 与 t_n 时刻的真实失效数 m_n 进行对比,即计算公式(14)中的 RE 值.取不同的预计点 $t_e (t_e \leq t_n)$,并重复上述过程,即可获得不同的 RE 值.如果 RE 值为正数,则意味着模型的预计值偏激进;反之,则意味着预计值偏保守.通常情况下,我们会将不同的 RE 值制成 RE 图,以直观地检验模型的预计能力.RE 图中接近于横轴(即 RE 值接近于 0)的点越多,表明模型预计能力越好.Musa 明确指出^[7],如果一个模型在失效数据集上根据 RE 值证明其对缺陷数均值具有最优的预计性能,那么它对其他软件可靠性度量也将具有最优的预计性能.

3.2 步骤1:建立适于DS1和DS2的考虑测试工作量和测试覆盖率的NHPP模型

1. 确定适用于 DS1 和 DS2 的测试工作量函数

将 Weibull-TEF、指数 Weibull-TEF(EW-TEF)、Logistic-TEF(LO-TEF)、广义 Logistic1 和广义 Logistic2 函数(GL-1&2-TEFs)以及 IS-TEF 这 6 种典型的 TEFs 分别应用于 DS1 和 DS2,每个 TEF 在两组数据集上的参

数估计值(LSE 法)以及拟合结果(MSE 值)见表 2,加粗数字表示该拟合结果是该数据集上的最佳结果.

Table 2 Estimated parameters and fitting results of six TEFs for DS1 and DS2
表 2 6 种 TEFs 的参数估计值与函数拟合值列表

TEFs	Cumulative testing-effort ($W(t)$)	DS1		DS2	
		Parameters (LSE)	MSE	Parameters (LSE)	MSE
Weibull-TEF	$W_{\max}(1-\exp(-bt^m))$	$W_{\max}=792.5$ $b=0.002347$ $m=1.115$	0.86	$W_{\max}=120.8$ $b=0.004098$ $m=1.906$	5.38
EW-TEF	$W_{\max}(1-\exp(-bt^m))^\theta$	$W_{\max}=886$ $b=0.1111$ $\theta=2.795$ $m=0.4619$	0.91	$W_{\max}=132.7$ $b=0.01427$ $\theta=1.342$ $m=1.501$	5.61
LO-TEF	$\frac{W_{\max}}{1+A\exp(-at)}$	$W_{\max}=54.84$ $A=13.03$ $a=0.2263$	1.63	$W_{\max}=99.9$ $A=28.01$ $a=0.2574$	5.90
GL-1-TEF	$\frac{W_{\max}}{\sqrt[3]{1+A\exp(-akt)}}$	$W_{\max}=59.73$ $A=0.224$ $a=0.00845$ $k=16.19$	1.34	$W_{\max}=117$ $A=1.003$ $a=0.02417$ $k=6.127$	4.90
GL-2-TEF	$W_{\max} \left(\frac{(k+1)/\beta}{1+A\exp(-akt)} \right)^{1/k}$	$W_{\max}=354.6$ $A=0.4669$ $a=1.029$ $k=0.1216$ $\beta=1.374$	1.06	$W_{\max}=98.68$ $A=0.811$ $a=1.05$ $k=0.1379$ $\beta=1.11$	4.90
IS-TEF	$W_{\max} \frac{1-e^{-bt}}{1+\psi e^{-bt}}$	$W_{\max}=91.58$ $b=0.09013$ $\psi=3.164$	0.83	$W_{\max}=107.3$ $b=0.2091$ $\psi=14.97$	4.55

由表 2 可知,对于 DS1 和 DS2 两组测试工作量数据集,IS-TEF 的拟合结果均是最优的(MSE 值最小,分别为 0.83 和 4.55).由此可表明,IS-TEF 是最适于 DS1 和 DS2 的测试工作量函数.

2. 确定适用于 DS1 和 DS2 的测试覆盖率函数

选择表 1 中列举的 3 种典型 TCF-SRGMs(即 LL-TCFM,LO-TCFM,PZ-TCFM)来确定适用于 DS1 和 DS2 的 TCF.将这 3 种 TCF-SRGMs 分别应用于 DS1 和 DS2,每个模型在两组失效数据集上的参数估计值(LSE 法)以及拟合结果(MSE 值)见表 3,加粗数字表示该拟合结果是该失效数据集上的最佳结果.

Table 3 Estimated parameters and fitting results of three TCF-SRGMs for DS1 and DS2
表 3 3 种 TCF-SRGMs 的参数估计值与模型拟合值列表

SRGMs	DS1		DS2	
	Parameters (LSE)	MSE	Parameters (LSE)	MSE
LL-TCFM	$N=631.8, \lambda=0.06, k=1.315$	111.79	$N=104.6, \lambda=0.1044, k=1.956$	7.44
PZ-TCFM	$N=328, \beta=0.01013, b=0.2554$	136.74	$N=103.3, \beta=6.07e^{-10}, b=0.2146$	8.85
LO-TCFM	$N=394.6, \alpha=0.1202, A=1.766$	86.05	$N=88.0, \alpha=0.1741, A=4.699$	5.17

由表 3 可知,对于 DS1 和 DS2 这两组失效数据集,LO-TCFM 模型的拟合结果均是最优的(MSE 值最小,分别为 86.05 和 5.17).由此可表明,该模型所对应的 LO-TCF 是最适于 DS1 和 DS2 的测试覆盖率函数.

3. 建立适于 DS1 和 DS2 的考虑 TEF 和 TCF 的 NHPP 模型

综上所述,适用于 DS1 和 DS2 的 TEF 和 TCF 分别为 IS-TEF 和 LO-TCF,将这两种函数同时代入公式(8),即可获得适于 DS1 和 DS2 的考虑测试工作量和测试覆盖率的 NHPP 模型,如公式(11)所示,即 IS-LO-SRGM.

3.3 步骤2:NHPP模型拟合和预计性能比较

本节将选取 IS-LO-SRGM 与表 1 中的 8 种 NHPP 模型分别在 DS1 和 DS2 上进行模型拟合与预计性能比较,以验证本文提出的 NHPP 建模框架的有效性.模型参数估计方法分别选用 MLE 和 LSE,对应的拟合性能比较准则分别选取 AIC 和 MSE.模型预计性能比较准则选取 RE 图法,预计点 t_e 从第 6 个时间点开始选取直至最后一个时间点.

1. 在两组失效数据集上的拟合结果

表 1 中的 8 个典型 NHPP 模型以及 IS-LO-SRGM 模型的参数估计值以及在 DS1 和 DS2 上的模型拟合值 (AIC 值和 MSE 值) 分别列举于表 4. 其中, 加粗数字表示该模型拟合结果是该失效数据集上的最佳结果.

Table 4 Estimated parameters and fitting results of the comparison NHPP SRGMs

表 4 对比 NHPP 软件可靠性模型的参数估计值与模型拟合值列表

SRGMs	DS1				DS2			
	Parameters (MLE)	AIC	Parameters (LSE)	MSE	Parameters (MLE)	AIC	Parameters (LSE)	MSE
GO	$N=513.15$ $r=0.0537$	220.76	$N=760.53$ $r=0.0323$	139.82	$N=133.43$ $r=0.045$	120.93	$N=153.1$ $r=0.04144$	22.9
IS	$N=355.06$ $r=0.222$ $\psi=3.27$	207.72	$N=374.39$ $r=0.179$ $\psi=3.01$	127.31	$N=88.606$ $r=0.249$ $\psi=6.186$	111.06	$N=87.21$ $r=0.2631$ $\psi=6.899$	5.10
LL-TCFM	$N=515.07$ $\lambda=0.079$ $k=1.377$	214.76	$N=631.8$ $\lambda=0.06$ $k=1.315$	111.79	$N=103.39$ $\lambda=0.106$ $k=1.8896$	110.79	$N=104.6$ $\lambda=0.1044$ $k=1.956$	7.44
LO-TCFM	$N=361.715$ $A=2.3199$ $\alpha=0.144$	206.26	$N=394.6$ $A=1.766$ $\alpha=0.120$	86.05	$N=89.28$ $A=4.229$ $\alpha=0.166$	110.86	$N=88$ $A=4.699$ $\alpha=0.174$	5.17
PZ-TCFM	$N=519.61$ $\beta=0.0097$ $b=0.1982$	218.44	$N=380$ $\beta=0.004$ $b=0.231$	138.69	$N=161.06$ $\beta=0.0127$ $b=0.158$	109.970	$N=103.3$ $\beta=2 \times 10^{-12}$ $b=0.2146$	8.85
LO-TEFM	$N=351.09$ $r=0.0344$	213.73	$N=395.6$ $r=0.04164$	114.05	$N=92.7942$ $r=0.0298$	110.581	$N=88.31$ $r=0.0338$	5.88
EW-TEFM	$N=568.54$ $r=0.0194$	216.85	$N=575.7$ $r=0.01916$	113.68	$N=94.7240$ $r=0.0254$	108.741	$N=94.63$ $r=0.0254$	7.05
IS-TEFM	$N=554.05$ $r=0.0201$	216.16	$N=564.2$ $r=0.01972$	107.05	$N=110.49$ $r=0.0274$	116.461	$N=94.79$ $r=0.0253$	5.28
IS-LO-SRGM	$N=378.759$ $A=1.21$ $\alpha=0.051$	205.60	$N=401.3$ $A=0.9388$ $\alpha=0.043$	83.05	$N=94.69$ $A=0.081$ $\alpha=0.0261$	111.122	$N=91.19$ $A=0.2255$ $\alpha=0.0295$	5.01

2. 在两组失效数据集上的预计结果

图 1 和图 2 中分别绘制了表 4 中 9 个 NHPP 模型在 DS1 和 DS2 上的预计 RE 图. 需要说明的是, 限于篇幅, 这里仅利用 LSE 法来计算这 9 个模型参数的估计值.

3. 拟合与预计结果的对比与分析

- 1) 由表 4 可知: IS-LO-SRGM 模型在两组失效数据集上均获得不错的模型拟合结果. 在 DS1 上, IS-LO-SRGM 的 MSE 值和 AIC 值均小于其他模型. 在 DS2 上, IS-LO-SRGM 的 MSE 值小于其他模型. 而 AIC 值虽然不是最小, 但与其他模型相差不多. 从而表明 IS-LO-SRGM 具有非常显著的模型拟合能力;
- 2) 由图 1 和图 2 可以看出, 与其他 NHPP 模型相比, IS-LO-SRGM 模型的 RE 曲线向横轴(图中连接零点的横线)接近的速度是最快的, 也即表明, IS-LO-SRGM 模型在两组失效数据集上具有最为优秀的预计效果;
- 3) 值得注意的是, 在两组失效数据集上, IS-LO-SRGM 模型的拟合和预计结果均优于只考虑测试工作量或者只考虑测试覆盖率的 NHPP 模型(如 LO-TCFM 或 IS-TEFM 等)的拟合效果;

上述对比结果表明: 同时考虑两种测试过程因素的 NHPP 建模方法比考虑单个测试过程因素的 NHPP 建模方法更为有效, 可更为显著地改进传统 NHPP 模型的拟合和预计性能.

- 4) 从第 3.2 节的建模过程可以看出, IS-LO-SRGM 中的测试工作量函数 IS-TEF 在 DS1 和 DS2 上均具有最为优秀的拟合结果, 也即 IS-TEF 最为准确地描述 DS1 和 DS2 中测试工作量的增长趋势. 并且由测试覆盖率函数 LO-TCF 建立的 TCF-SRGM 模型在 DS1 和 DS2 上也具有最为优秀的模型拟合能力.

这表明, 第 2.1.3 节中关于如何根据失效数据集以及公式(8)来建立最为适合的 NHPP 模型的过程是合理且可行的. 由此可以得到一个初步推论: 选择合适的测试工作量函数以及测试覆盖率函数, 对于所建立的 NHPP 模型的拟合和预计能力有着十分重要的影响. 限于篇幅, 关于此推论的深入验证将留在后续工作中进行.

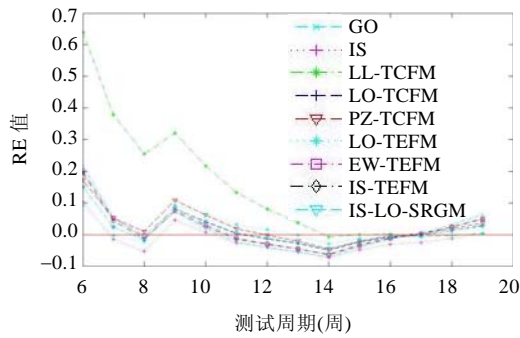


Fig.1 RE curves of NHPP SRGMs for DS1

图 1 NHPP 模型在 DS1 数据集上的预计 RE 图

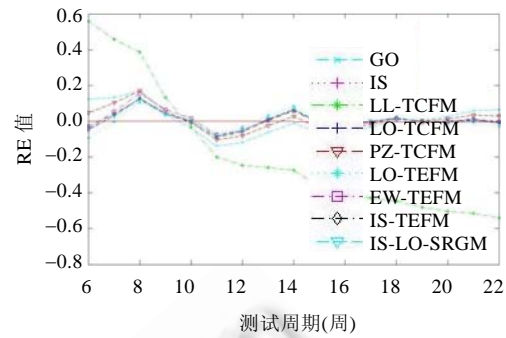


Fig.2 RE Curves of NHPP SRGMs for DS2

图 2 NHPP 模型在 DS2 数据集上的预计 RE 图

4 考虑不完美排错的初步讨论

在实际的测试过程中,由于软件复杂性与开发人员水平等因素的影响,排错的同时往往会引入新的缺陷,即所谓的不完美排错现象.因此,在软件可靠性建模过程中应对这种不完美排错现象进行准确描述^[41,42].同时考虑测试工作量、测试覆盖率以及不完美排错的 NHPP 软件可靠性建模框架的建模假设如下:

- 1) 软件失效的探测与排除过程随时间服从 NHPP 过程;
- 2) 当前失效是由软件中残存的缺陷引发的;
- 3) 在时间间隔 $[t, t+\Delta t]$ 内,所探测到的缺陷数均值与当前的测试工作量消耗率的比值,与软件残存缺陷数均值成正比,这个比值即为当前的缺陷探测率函数 $r(t)$,可由 t 时刻的测试覆盖增长速率函数 $C'(t)$ 与当前未覆盖代码比例 $1-C(t)$ 的比值来表示;
- 4) 排除缺陷的过程可能会引入新缺陷,也即软件的缺陷总数 N 是一个随时间变化的函数 $N(t)$.

根据上述假设,即可获得如下建模框架:

$$\frac{dm(t)}{dt} \frac{1}{w(t)} = \frac{C'(t)}{1-C(t)} [N(t) - m(t)] \quad (15)$$

限于篇幅,本节仅给出一个初步的研究成果,关于该成果的分析、验证等工作将在后续研究中进行.

5 结论

测试工作量与测试覆盖率是两种重要的软件测试过程因素.为定量描述这两种测试因素对于 NHPP 软件可靠性建模效果的综合影响效果,本文首先提出一个同时考虑测试工作量和测试覆盖率的 NHPP 类软件可靠性建模框架.在此基础上,结合已有的 IS-TEF 以及 LO-TCF,提出新的 NHPP 软件可靠性模型,即 IS-LO-SRGM.然后,对如何利用新建模框架,根据失效数据集进行建模过程进行了讨论与分析.在两组真实的失效数据集上,利用新的建模框架进行建模,并将所建立的 IS-LO-SRGM 模型分别与 8 个典型的 NHPP 类模型进行了拟合与预计性能对比.由对比结果可知,新模型 IS-LO-SRGM 的拟合与预计结果在两组数据集上均显著优于其他对比模型.最后,还对建模框架中的不完美排错情况进行了初步的讨论与分析.

综上所述,本文的研究成果具有如下两个方面的意义:

- 1) 理论意义:据我们所知,这是首次将测试工作量与测试覆盖率引入 NHPP 类软件可靠性模型的研究工作.如何在建模过程中考虑尽可能多的测试过程因素,是当前软件可靠性建模研究领域的一个难点问题.文献[6,8]中均对此问题进行了讨论研究,但这两篇文献的建模过程非常繁复,不利于实例化.而本文提出的建模框架则可以很直观且方便地引入两种甚至 3 种重要的测试过程因素;
- 2) 工程意义:与以往大部分的软件可靠性建模文献不同,本文研究成果的重点不在于提出一个新的在

某些失效数据集上具有较为优秀的拟合和预计性能的软件可靠性模型,而在于提出新的建模框架和结合失效数据集的动态建模思想(即根据实际的失效数据集,利用新的建模框架来确定具体的 TCF 和 TEF 形式,进而建立最适合该组失效数据集的考虑多种测试过程因素的软件可靠性模型),这使得本文的研究成果可以显著提升现有 NHPP 类软件可靠性模型的拟合与预计性能,并且具有优秀的实用性与适用性.

References:

- [1] Lyu MR. Handbook of Software Reliability Engineering. New York: McGraw Hill, 1996.
- [2] Kapur PK, Ompal S, Aggarwal AG, Kumar R. Unified framework for developing testing effort dependent software reliability growth models. WSEAS Trans. on Systems, 2009,4(8):521-531.
- [3] Huang CY, Lyu MR. Estimation and analysis of some generalized multiple change-point software reliability models. IEEE Trans. on Reliability, 2011,60(2):498-514. [doi: 10.1109/TR.2011.2134350]
- [4] Xu GC, Liu XZ, Hu L, Fu XD, Dong YS. Software reliability assessment models incorporating software defect correlation. Ruanjian Xuebao/Journal of Software, 2011,22(3):439-450 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3739.htm> [doi: 10.3724/SP.J.1001.2011.03739]
- [5] Okamura H, Etani Y, Dohi T. A multi-factors software reliability model based on Logistic regression. In: Marathe M, ed. Proc. of the IEEE 21st Int'l Symp. on Software Reliability Engineering (short for ISSRE). San Jose: IEEE Computer Society, 2010. 31-40.
- [6] Kapur PK, Phan H, Anand S, Yadav K. A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. IEEE Trans. on Reliability, 2011,60(1):331-340. [doi: 10.1109/TR.2010.2103590]
- [7] Huang CY, Kuo SY, Lyu MR. An assessment of testing-effort dependent software reliability growth models. IEEE Trans. on Reliability, 2007,56(2):198-211. [doi: 10.1109/TR.2007.895301]
- [8] Shibata K, Rinsaka K, Dohi T. Metrics-Based software reliability models using non-homogeneous Poisson processes. In: Vouk MA, ed. Proc. of the IEEE Int'l Symp. on Software Reliability Engineering (ISSRE). Raleigh: IEEE Computer Society, 2006. 52-61. [doi: 10.1109/ISSRE.2006.28]
- [9] Cai X, Lyu MR. Software reliability modeling with test coverage experimentation and measurement with a fault-tolerant software project. In: Goseva-Popstojanova K, ed. Proc. of the 18th IEEE Int'l Symp. on Software Reliability Engineering. Trollhattan: IEEE Computer Society, 2007. 17-26. [doi: 10.1109/ISSRE.2007.17]
- [10] Kuo SY, Huang CY, Lyu MR. Framework for modeling software reliability, using various testing-efforts and fault-detection rates. IEEE Trans. on Reliability, 2001,50(3):310-320. [doi: 10.1109/24.974129]
- [11] Huang CY, Kuo SY. Analysis of incorporating logistic testing-effort function into software reliability modeling. IEEE Trans. on Reliability, 2002,51(3):261-270. [doi: 10.1109/TR.2002.801847]
- [12] Ahmad N, Khan MGM, Rafi LS. A study of testing-effort dependent inflection S-shaped software reliability growth models with imperfect debugging. Int'l Journal of Quality & Reliability Management, 2010,27(1):89-110. [doi: 10.1108/02656711011009335]
- [13] Norman EF, Martin N. A critique of software defect prediction models. IEEE Trans. on Software Engineering, 1999,25(5):675-688. [doi: 10.1109/32.815326]
- [14] Yamada S, Hishitani J, Osaki S. Software reliability growth model with a Weibull testing effort: A model and application. IEEE Trans. on Reliability, 1993,42(1):100-106. [doi: 10.1109/24.210278]
- [15] Ahmad N, Bokhari MU, Quadri SMK, Khan MGM. The exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy. Int'l Journal of Quality & Reliability Management, 2008,25(2):211-235. [doi: 10.1108/02656710810846952]
- [16] Lin CT, Huang CY. Enhancing and measuring the predictive capabilities of testing-effort dependent software reliability models. The Journal of Systems and Software, 2008,81:1025-1038. [doi: 10.1016/j.jss.2007.10.002]
- [17] Ahmad N, Khan MGM, Quadri SMK, Kumar M. Modeling and analysis of software reliability with Burr type X testing-effort and release-time determination. Journal of Modeling in Management, 2009,4(1):28-54. [doi: 10.1108/17465660910943748]
- [18] Huang CY. Cost-Reliability-Optimal release policy for software reliability models incorporating improvements in testing efficiency. The Journal of Systems and Software, 2005,77:139-155. [doi: 10.1016/j.jss.2004.10.014]

- [19] Huang CY, Lyu MR. Optimal release time for software systems considering cost, testing-effort, and test efficiency. *IEEE Trans. on Reliability*, 2005,54(4):583–591. [doi: 10.1109/TR.2005.859230]
- [20] Ahmad N, Quadri SMK, Mohd R. Comparison of predictive capability of software reliability growth models with exponentiated Weibull distribution. *Int'l Journal of Computer Applications*, 2011,15(6):40–43. [doi: 10.5120/1949-2607]
- [21] Chatterjee S, Misra RB, Alam SS. Joint effect of test effort and learning factor on software reliability and optimal release policy. *Int'l Journal of System Science*, 1997,28(4):391–396. <http://dx.doi.org/10.1080/00207729708929399>
- [22] Yamada S, Ohtera H, Narihisa H. Software reliability growth models with testing-effort. *IEEE Trans. on Reliability*, 1986,R-35(1):19–23. [doi: 10.1109/TR.1986.4335332]
- [23] Kapur PK, Goswami DN, Bardhan A, Singh O. Flexible software reliability growth model with testing effort dependent learning process. *Applied Mathematical Modeling*, 2008,32:1298–1307. [doi: 10.1016/j.apm.2007.04.002]
- [24] Kapur PK, Shatnawi O, Aggarwal AG, Ravi K. Unified framework for developing testing effort dependent software reliability growth models. *WSEAS Trans. on Systems*, 2009,4(8):521–531.
- [25] Jha PC, Gupta D, Yang B, Kapur PK. Optimal testing resource allocation during module testing considering cost, testing effort and reliability. *Computers & Industrial Engineering*, 2009,57:1122–1130. [doi: 10.1016/j.cie.2009.05.001]
- [26] Huang CY. Performance analysis of software reliability growth models with testing-effort and change-point. *The Journal of Systems and Software*, 2005,76:181–194. [doi: 10.1016/j.jss.2004.04.024]
- [27] Kapur PK, Gupta A. Testing effort control using flexible software reliability growth model with change point. *Int'l Journal of Performability Engineering*, 2006,2(3):245–262.
- [28] Kapur PK, Grover PS, Younes S. Modeling an imperfect debugging phenomenon with testing effort. In: Ferraz P, ed. *Proc. of the ISSRE*. Monterey: IEEE Computer Society, 1994. 178–183. [doi: 10.1109/ISSRE.1994.341371]
- [29] Lin CT, Huang CY, Chang JR. Integrating generalized Weibull-type testing-effort function and multiple change-points into software reliability growth models. In: Huang WT, ed. *Proc. of the 12th Asia-Pacific Software Engineering Conf*. Taipei: IEEE Computer Society, 2005. 431–438.
- [30] Li QY, Li HF, Lu MY, Wang XC. Software reliability growth model with S-shaped testing effort function. *Journal of Beijing University of Aeronautics and Astronautics*, 2011,37(2):149–155 (in Chinese with English abstract).
- [31] Malaiya YK, Li MN. Software reliability growth with test coverage. *IEEE Trans. on Reliability*, 2002,51(4):420–426. [doi: 10.1109/TR.2002.804489]
- [32] Del Frate F, Garg P, Mathur AP, Pasquini A. On the correlation between code coverage and software reliability. In: Joseph H, ed. *Proc. of the 6th IEEE Int'l Symp. on Software Reliability Engineering*. Toulouse: IEEE Computer Society, 1995. 124–132. [doi: 10.1109/ISSRE.1995.497650]
- [33] Briand LC, Phahl D. Using simulation for assessing the real impact of test-coverage on defect-coverage. *IEEE Trans. on Reliability*, 2000,49(1):60–70.
- [34] Cai X. Coverage-Based testing strategies and reliability modeling for fault-tolerant software systems [Ph.D. Thesis]. Hong Kong: The Chinese University of Hong Kong, 2006.
- [35] Lyu MR, Huang Z, Sze SKS, Cai X. An empirical study on testing and fault tolerance for software reliability engineering. In: Malaiya Y, ed. *Proc. of the 14th IEEE Int'l Symp. on Software Reliability Engineering*. Denver: IEEE Computer Society, 2003. 119–130. [doi: 10.1109/ISSRE.2003.1251036]
- [36] Gokhale SS, Trivedi KS. A time/structure based software reliability model. *Annals of Software Engineering*, 1999,8(1):85–121. [doi: 10.1023/A:1018923329647]
- [37] Pham H, Zhang XM. NHPP software reliability and cost models with testing coverage. *European Journal of Operational Research*, 2003,145(2):443–454. [doi: 10.1016/S0377-2217(02)00181-9]
- [38] Li HF, Li QY, Lu MY. Software reliability modeling with logistic test coverage function. *Journal of Computer Research and Development*, 2011,48(2):232–240 (in Chinese with English abstract).
- [39] Huang CY, Lin CT. Software reliability analysis by considering fault dependency and debugging time lag. *IEEE Trans. on Reliability*, 2006,55(3):436–450. [doi: 10.1109/TR.2006.879607]

- [40] Lin CT. Reliability analysis and management during software testing and debugging [Ph.D. Thesis]. Hsinchu City: TsingHua University, 2009.
- [41] Xie JY, An JX, Zhu JH. NHPP software reliability growth model considering imperfect debugging. Ruanjian Xuebao/Journal of Software, 2010,21(5):942-949 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3539.htm> [doi: 10.3724/SP.J.1001.2010.03539]
- [42] Barghout M. Predicting software reliability using an imperfect debugging Jelinski-Moranda non-homogeneous Poisson process model. Model Assisted Statistics and Applications, 2010,5:31-41.

附中文参考文献:

- [4] 徐高潮,刘新忠,胡亮,付晓东,董玉双.引入关联缺陷的软件可靠性评估模型.软件学报,2011,22(3):439-450. <http://www.jos.org.cn/1000-9825/3739.htm> [doi: 10.3724/SP.J.1001.2011.03739]
- [30] 李秋英,李海峰,陆民燕,王学成.基于 S 型测试工作量函数的软件可靠性增长模型.北京航空航天大学学报,2011,37(2):149-155.
- [38] 李海峰,李秋英,陆民燕.基于 Logistic 测试覆盖率函数的软件可靠性建模研究.计算机研究与发展,2011,48(2):232-240.
- [41] 谢景燕,安金霞,朱纪洪.考虑不完美排错情况的 NHPP 类软件可靠性增长模型.软件学报,2010,21(5):942-949. <http://www.jos.org.cn/1000-9825/3539.htm> [doi: 10.3724/SP.J.1001.2010.03539]



李海峰(1981—),男,辽宁盘锦人,博士,工程师,主要研究领域为软件可靠性测试与建模,软件安全性分析.

E-mail: lihaifeng@dse.buaa.edu.cn



王栓奇(1984—),男,博士,主要研究领域为软件可靠性模型及测试,软件可信性评估.

E-mail: wangshuanqi@163.com



刘畅(1979—),男,博士,高级工程师,主要研究领域为软件仿真测试环境,软件安全性分析,软件可靠性测试.

E-mail: liuchang@dse.buaa.edu.cn



郑军(1969—),男,研究员,主要研究领域为软件测试技术,软件质量评价技术.

E-mail: zhengjun@sina.com



李震(1977—),男,博士,讲师,主要研究领域为软件测试技术,软件质量评价技术.

E-mail: justlz@163.com