

基于分阶段增长测试信息的软件可靠性贝叶斯验证方案

王玉琢, 刘海涛, 袁昊劼, 翟亚利, 张志华

引用本文

王玉琢, 刘海涛, 袁昊劼, 翟亚利, 张志华. 基于分阶段增长测试信息的软件可靠性贝叶斯验证方案[J]. 计算机科学, 2025, 52(9): 360-367.

WANG Yuzhuo, LIU Haitao, YUAN Haojie, ZHAI Yali, ZHANG Zhihua. [Bayesian Verification Scheme for Software Reliability Based on Staged Growth Test Information](#) [J]. Computer Science, 2025, 52(9): 360-367.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于函数调用序列模式挖掘的程序缺陷检测](#)

Defects Detection Based on Mining Function Call Sequence Patterns

计算机科学, 2017, 44(11): 226-231. <https://doi.org/10.11896/j.issn.1002-137X.2017.11.034>

[大粒度软件复用](#)

计算机科学, 1997, 24(2): 69-73.

[基于本体的视频语义内容分析](#)

Video Semantic Content Analysis Based on Ontology

计算机科学, 2009, 36(7): 170-174. <https://doi.org/10.11896/j.issn.1002-137X.2009.07.040>

[基于信息熵与SIFT算法的天文图像配准](#)

Astronomical Image Registration Combining Information Entropy and SIFT Algorithm

计算机科学, 2015, 42(6): 57-60. <https://doi.org/10.11896/j.issn.1002-137X.2015.06.013>

基于分阶段增长测试信息的软件可靠性贝叶斯验证方案

王玉琢^{1,2} 刘海涛¹ 袁昊劼¹ 翟亚利¹ 张志华³

1 海军工程大学基础部 武汉 430033

2 海军工程大学兵器工程学院 武汉 430033

3 海军工程大学舰船与海洋学院 武汉 430033

(0907032003@nue.edu.cn)

摘要 针对现有贝叶斯方案的先验分布确定方法对软件可靠性分阶段增长测试信息加工偏保守、假设条件偏理想的问题,利用正则不完全贝塔函数与二项分布累积和之间的恒等关系式,给出增长测试最后一阶段软件成功率的边缘概率分布模型。在此基础上,提出了顺序约束条件下的软件成功率先验分布确定方法,并从保护使用方利益的角度设计了基于平均后验风险的软件可靠性贝叶斯验证方案。经实例和仿真验证,所提先验分布确定方法对软件可靠性分阶段增长测试信息的加工更合理,设计的贝叶斯方案能在保证方案可信度的前提下明显减少可靠性验证测试用例数量,减轻测试负担,具有一定的经济价值。

关键词 软件可靠性验证;可靠性增长测试;软件成功率

中图分类号 E917

Bayesian Verification Scheme for Software Reliability Based on Staged Growth Test Information

WANG Yuzhuo^{1,2}, LIU Haitao¹, YUAN Haojie¹, ZHAI Yali¹ and ZHANG Zhihua³

1 Department of Foundation, Naval University of Engineering, Wuhan 430033, China

2 College of Weaponry Engineering, Naval University of Engineering, Wuhan 430033, China

3 College of Naval Architecture and Ocean Engineering, Naval University of Engineering, Wuhan 430033, China

Abstract The prior distribution determination methods of existing Bayesian schemes are relatively conservative and ideal for processing software reliability phased growth testing information. In this article, an edge distribution model for the software success probability at the final stage of growth testing is constructed using the identity relationship between the regular incomplete beta function and the cumulative sum of binomial distributions. On this basis, a prior distribution determination method of software success probability under sequential constraints is proposed, and a Bayesian verification scheme based on average a posteriori risk is designed to protect the interests of users. The example and simulation show that the proposed prior distribution determination method is more reasonable for processing software reliability phased growth information, the designed Bayesian scheme can significantly reduce the number of reliability verification test cases and reduce testing burden while ensuring the reliability of the schemes and has certain economic value.

Keywords Software reliability verification, Reliability growth testing, Software success rate

随着现代产品智能化水平的不断提升,人们对产品功能迭代的需求与日俱增。为了提高产品的可扩展性和可维护性,许多行业都采取软硬件分离的研发模式,将软件和硬件独立地设计、开发、测试、部署,最终推向市场^[1-2]。可靠性是衡量软件质量的一项重要指标。为确保投入使用的软件产品质量,软件订购方特别是安全关键领域的订购方,往往会规定软件产品的可靠性指标。因此,不仅需要在软件研发阶段开展模拟用户使用环境下的软件测试,尽可能多地排除软件缺陷,提升软件可靠性,还需要在最后的验收阶段验证软件可靠性是否满足规定的指标要求^[3]。

目前,软件可靠性验证主要依托软件可靠性验证测试(Software Reliability Demonstration Testing, SDRT)开展。受时间、经济、人力成本等客观因素的制约,如果可靠性验证所

需的测试用例数量过大,验收工作将难以顺利开展。基于贝叶斯理论的可靠性验证方法利用产品先验信息和现场测试信息给出验证结论,能缩短可靠性验证期,还可以改善可靠性验证工作实施困难的问题。因此,该方法越来越受到工程界和学术界的重视。

贝叶斯方案的制定步骤为:首先确定待验证可靠性参数的先验分布,然后确定后验分布,最后制定验收方案。可见,先验分布的确定是贝叶斯方案的关键。尽管目前关于可靠性贝叶斯验证方法的研究已有不少,但软件在交付验收前经历的通常是可靠性增长测试,现有的贝叶斯方案的先验分布确定方法对软件先验信息的加工仍不够合理,假设条件比较理想、主观。例如,无先验信息的贝叶斯验证方案采用无先验信息先验分布确定方法^[4],假设产品失败率服从均匀分布,然而

可靠性增长测试末期的软件失败概率已降低到一定值,不会服从0与1之间的均匀分布。基于减函数的可靠性贝叶斯验证方法将软件失败率的先验分布构造为减函数^[5],符合可靠性增长测试末期软件失败率取较大值的可能性较小、取较小值的可能性较大的特点。但在计算时,为方便处理,依主观经验将先验分布中的待估参数简化为1个,对先验信息的利用不够充分。基于先验矩的先验分布确定方法^[6]将软件成功率的先验分布确定为贝塔分布,用可靠性增长测试数据的一阶矩和二阶矩估计贝塔分布的两个参数值。然而,以成功率为可靠性指标的软件,其可靠性增长测试数据是来自不同母体的分组测试数据,用分组数据的样本矩来估计参数的先验分布,虽然充分利用了先验信息,但对分阶段增长信息的加工方法不够合理。还有一些学者构建了融合各结构层次先验信息的故障样本量确定方法、基于多寿命分布模型的贝叶斯后验概率改进方法等^[7-9],但这些方法因为加工的先验信息结构各不相同,不能完全适用于软件可靠性增长先验信息的加工。

本文以可靠性指标为成功率的软件为研究对象,针对其分阶段增长的先验信息特征,利用正则不完全贝塔函数与二项分布部分和之间的恒等关系式,研究增长测试最后一阶段软件成功率的边缘概率分布;在此基础上,给出软件成功率的先验分布确定方法,并设计软件可靠性贝叶斯验证方案。最后,通过实例和仿真对所提方法的有效性进行验证。

1 问题的提出

软件在研发阶段通常会经历一系列常规测试,如功能测试、性能测试、接口测试、安全性测试、边界测试、回归测试等。如果还有明确的可靠性指标要求,常规测试活动完成之后,研发方还会安排可靠性增长测试,通过不断查找和移除缺陷来提高软件可靠性水平,使其达到规定的可靠性指标要求^[10]。当可靠性指标为成功率/失败率时,软件可靠性增长测试通常以“测试—缺陷暴露—集中排除—再测试”的方式进行,对于两次集中排错之间检测出的缺陷,只统计数量,不调试缺陷。因此,增长测试阶段的软件可靠性水平总体上以一种分阶段增长模式逐步提高。记测试期间的软件累计被调试次数为 l ,测试失效数据集为 $\Omega = \{(n_1, x_1), (n_2, x_2), \dots, (n_l, x_l)\}$,其中 (n_i, x_i) 表示第 i 阶段总共执行了 n_i 个测试用例,有 x_i 条用例执行成功, $n_i - x_i$ 条用例执行失败。再记 $R_i (i=1, 2, \dots, l)$ 为第 i 阶段的软件可靠度真实水平,基于可靠性增长测试环境,可以假设 R_i 满足如下顺序约束条件:

$$0 < R_1 < R_2 < \dots < R_l < 1 \quad (1)$$

其关系如图1所示。

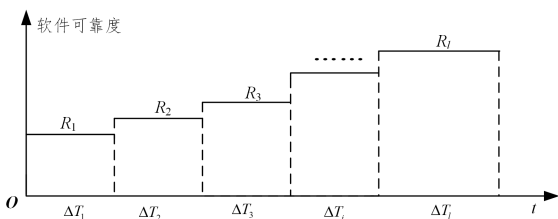


图1 软件可靠性增长测试

Fig. 1 Software reliability growth test

可靠性增长测试通常实施于研发阶段末期,增长测试结束之后,软件转入验收阶段。软件可靠性验证通常依托可靠性验证测试实现,由第三方机构组织开展,验证测试按照预定的验证方案执行。测试目的是最终检验而不是调试,测试期间所有被检出缺陷不进行调试,测试结束后统计失效数据,评估软件可靠性,并给出统计学意义上的接收/拒绝结论。因此,验证测试阶段的软件可靠性水平不增长。不妨记验证测试阶段的软件可靠性为 R ,本文研究的主要问题就是如何利用交付验收前积累的 l 组可靠性增长测试数据 $\Omega = \{(n_1, x_1), (n_2, x_2), \dots, (n_l, x_l)\}$ 来确定验收阶段软件成功率 R 的先验分布,并设计软件可靠性贝叶斯验证方案。

通过对软件可靠性工程过程的分析可以看出,研发期最后一阶段的软件可靠性水平 R_l 对验收期软件可靠性 R 先验分布的确定非常重要,且 R_l 的概率分布可以理解为研发期分阶段增长测试样本的边缘分布。因此从研究 R_l 的边缘概率分布入手,确定研究思路与流程,如图2所示。

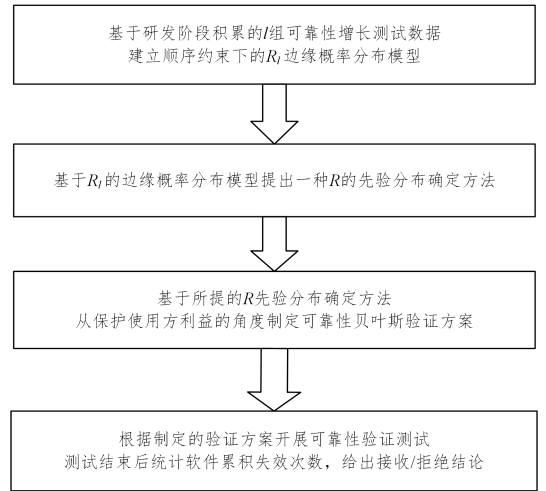


图2 研究流程图

Fig. 2 Research flow chart

2 R_l 的边缘概率分布模型

2.1 可靠性增长测试样本的似然函数

对于一个研发期经历了 l 次可靠性增长、增长测试失效数据集为 $\Omega = \{(n_1, x_1), (n_2, x_2), \dots, (n_l, x_l)\}$ 的软件,引入随机变量 X_i 表示第 i 阶段测试执行成功的用例个数,则 x_i 为 X_i 的样本观测值, $i=1, 2, \dots, l$ 。由于增长测试的任意两次集中纠错之间不调试缺陷,因此,将第 i 阶段 n_i 个测试用例的独立执行看作 n_i 重伯努利实验, X_i 就服从参数为 n_i 和 R_i 的二项分布,即:

$$X_i \sim B(n_i, R_i)$$

在参数 R_i 取定的情况下, X_i 的似然函数为:

$$L_{X_i}(n_i, x_i | R_i) = C_{n_i}^{x_i} \cdot R_i^{x_i} (1 - R_i)^{n_i - x_i}$$

再假设各阶段增长测试相互独立, X_1, X_2, \dots, X_l 的联合似然函数为:

$$L_{X_1, X_2, \dots, X_l}(\Omega | R_1, \dots, R_l) = \prod_{i=1}^l C_{n_i}^{x_i} R_i^{x_i} (1 - R_i)^{n_i - x_i} \quad (2)$$

2.2 R_l 的边缘概率分布

将参数 R_i 看作随机变量,根据贝叶斯理论, R_i 存在先验

分布和后验分布。采取共轭分布法将 R_i 的先验分布确定为贝塔分布,即:

$$\pi(R_i) = \text{Be}(x_{i0}, n_{i0} - x_{i0}) = \frac{1}{B(x_{i0}, n_{i0} - x_{i0})} R_i^{x_{i0}-1} (1-R_i)^{n_{i0}-x_{i0}-1}$$

n 维随机变量 R_1, R_2, \dots, R_l 的先验分布为:

$$\pi(R_1, \dots, R_l) = \prod_{i=1}^l \pi(R_i) \propto \prod_{i=1}^l R_i^{x_{i0}-1} (1-R_i)^{n_{i0}-x_{i0}-1} \quad (3)$$

其中, (n_{i0}, x_{i0}) 为 $\pi(R_i)$ 的超参数,可根据第 i 阶段增长测试信息的特点来选择合适的估计方法。由于软件测试基本不存在硬件实验的物理损耗问题,温度、湿度等环境因素对软件测试的影响也很小,因此在各阶段可靠性增长测试相互独立的

$$h(R_l, \Omega) = \int_0^{R_l} \int_0^{R_{l-1}} \dots \int_0^{R_2} h(R_1, \dots, R_l, \Omega) dR_1 dR_2 \dots dR_{l-1} = R_l^{x_l-1} (1-R_l)^{n_l-x_l-1} \times \int_0^{R_l} R_{l-1}^{x_{l-1}-1} (1-R_{l-1})^{n_{l-1}-x_{l-1}-1} \times \dots \times \int_0^{R_2} R_1^{x_1-1} (1-R_1)^{n_1-x_1-1} dR_1 dR_2 \dots dR_{l-1} \quad (5)$$

2.3 边缘概率分布的解析解

$h(R_l, \Omega)$ 的计算涉及复杂高维积分,直接求解比较困难。一些学者通过数值算法如分层抽样算法^[12]和 Gibbs 抽样法^[13]给出了软件成功率边缘分布的数值解。然而数值计算存在不可避免的随机误差,初值的选取、收敛的精度等主观因素均会影响验证方案的可信性。本小节利用不完全贝塔函数与二项分布累积和的恒等关系式,结合数学归纳法给出 R_l 边缘分布的解析解。

引理 1 已知完全贝塔函数的定义为 $B(s, n-s) = \int_0^1 \theta^{s-1} (1-\theta)^{n-s-1} d\theta$, 不完全贝塔函数的定义为 $B(R; s, n-s) = \int_0^R \theta^{s-1} (1-\theta)^{n-s-1} d\theta$, 正则不完全贝塔函数的定义为 $I_R(s,$

$$\left\{ \begin{aligned} B(R; s+1, n-s-1) &= \frac{1}{s+1} (1-R)^{n-s-2} R^{s+1} + \frac{n-s-2}{s+1} B(R; s+2, n-s-2) \\ B(R; s+2, n-s-2) &= \frac{1}{s+2} (1-R)^{n-s-3} R^{s+2} + \frac{n-s-3}{s+2} B(R; s+3, n-s-3) \\ &\dots \\ B(R; n-2, 2) &= \frac{1}{n-2} (1-R) R^{n-2} + \frac{1}{n-2} B(R; n-1, 1) \\ B(R; n-1, 1) &= \frac{1}{n-1} R^{n-1} \end{aligned} \right.$$

于是,不完全贝塔函数 $B(R; s, n-s)$ 可由如下有限项累积和表示:

$$B(R; s, n-s) = \sum_{k=0}^{n-s-1} \frac{(s-1)!}{(s+k)!} \cdot \frac{(n-s-1)!}{(n-s-k-1)!} (1-R)^{n-s-k-1} R^{k+s}$$

令 $x = k + s$, 上式可等价表示为:

$$B(R; s, n-s) = \frac{\sum_{x=s}^{n-1} (s-1)! \cdot (n-s-1)!}{(n-1)!} \cdot \frac{(n-1)!}{x! \cdot (n-x-1)!} (1-R)^{n-x-1} R^x = B(s, n-s) \sum_{x=s}^{n-1} C_{n-1}^x (1-R)^{n-x-1} R^x$$

因此有 $I_R(s, n-s) = \sum_{x=s}^{n-1} C_{n-1}^x (1-R)^{n-x-1} R^x$, 其中 $I_R(s, n-s)$ 为正则不完全贝塔函数。注意到 $\sum_{x=s}^{n-1} C_{n-1}^x (1-R)^{n-x-1} R^x$

情况下,可以采取无先验信息法来估计超参数 (n_{i0}, x_{i0}) 的值。常见的无先验信息超参数估计法有贝叶斯法、Novick-hall 法、Jeffrey 法^[11]。研究表明,在实验数据量较大的情况下,无先验信息超参数估计法的选择对贝叶斯统计推断结果的影响都很小。为此,本文采用贝叶斯法取 (n_{i0}, x_{i0}) 为 $(2, 1)$, 可得 R_1, R_2, \dots, R_l 与样本集 Ω 的联合密度函数为:

$$h(R_1, \dots, R_l, \Omega) = L(\Omega | R_1, \dots, R_l) \cdot \pi(R_1, \dots, R_l) \propto \prod_{i=1}^l R_i^{x_i-1} (1-R_i)^{n_i-x_i-1} \quad (4)$$

利用 R_1, R_2, \dots, R_l 的联合分布与 R_l 的边缘分布之间的关系以及 $0 < R_1 < R_2 < \dots < R_l < 1$ 的顺序约束条件, R_l 的边缘分布可表示为:

$n-s) = \frac{B(R; s, n-s)}{B(s, n-s)}$, 其中 $s \in \mathbb{Z}^+$, 则正则不完全贝塔函数

与二项分布部分和之间满足如下恒等关系式:

$$I_R(s, n-s) = \sum_{x=s}^{n-1} C_{n-1}^x (1-R)^{n-x-1} R^x$$

证明:利用分部积分法计算不完全贝塔函数,可得如下递推关系式:

$$B(R; s, n-s) = \int_0^R \theta^{s-1} (1-\theta)^{n-s-1} d\theta = \frac{1}{s} (1-R)^{n-s-1} R^s + \frac{n-s-1}{s} B(R; s+1, n-s-1)$$

根据该递推关系式进一步可得:

即为二项分布 $B(n, R)$ 的分布函数部分和,故定理得证。

接下来,利用引理 1 首先将式(5)的一重定积分转化为:

$$\int_0^{R_2} R_1^{q_1-1} (1-R_1)^{m_1-s_1-1} dR_1 = B(s_1, n_1-s_1) \sum_{k_1=s_1}^{q_1} C_{q_1}^{k_1} R_2^{k_1} (1-R_2)^{q_1-k_1}$$

其中, $q_1 = n_1 - 1$ 。然后将式(5)中的二重定积分转化为:

$$\int_0^{R_3} \int_0^{R_2} R_2^{s_2-1} (1-R_2)^{n_2-s_2-1} R_1^{s_1-1} (1-R_1)^{m_1-s_1-1} dR_1 dR_2 = B(s_1, n_1-s_1) \sum_{k_1=s_1}^{q_1} C_{q_1}^{k_1} \int_0^{R_3} R_2^{s_2+k_1-1} (1-R_2)^{n_2-s_2+q_1-k_1-1} dR_2 = B(s_1, n_1-s_1) \times \sum_{k_1=s_1}^{q_1} \sum_{k_2=s_2+k_1}^{q_2} C_{q_1}^{k_1} C_{q_2}^{k_2} B(s_2+k_1, n_2-s_2+q_1-k_1) R_3^{k_2} (1-R_3)^{q_2-k_2}$$

其中, $q_2 = n_1 + n_2 - 2$ 。根据数学归纳法将 $l-1$ 重积分逐层转化为有限项求和式,可计算出 R_l 的边缘分布:

$$\begin{aligned} h(R_l, \Omega) &= B(x_1, n_1 - x_1) \times B(x_2 + k_1, n_2 - x_2 + q_1 - k_1) \times \cdots \times B(x_{l-1} + k_{l-2}, n_{l-1} - x_{l-1} + q_{l-2} - k_{l-2}) \times \\ &C_{q_1}^{k_1} C_{q_2}^{k_2} \cdots C_{q_{l-1}}^{k_{l-1}} \times \sum_{k_1=x_1}^{q_1} \sum_{k_2=x_2+k_1}^{q_2} \cdots \sum_{k_{l-1}=x_{l-1}+k_{l-2}}^{q_{l-1}} (R_l)^{k_{l-1}} (1 - R_l)^{q_{l-1}-k_{l-1}} \end{aligned}$$

其中, $q_i = \sum_{j=1}^i (n_j - 1)$, $i=1, 2, \dots, l-1$ 。

注意到 $(R_l)^{k_{l-1}} (1 - R_l)^{q_{l-1}-k_{l-1}}$ 为贝塔分布核函数,满足 $(R_l)^{k_{l-1}} (1 - R_l)^{q_{l-1}-k_{l-1}} = B(x_l + k_{l-1}, q_{l-1} - k_{l-1} + n_l - x_l) \cdot \text{Be}(x_l + k_{l-1}, q_{l-1} - k_{l-1} + n_l - x_l)$

于是 $h(R_l, \Omega)$ 可等价表示为:

$$h(R_l, \Omega) = B(x_1, n_1 - x_1) \cdot \sum_{k_1=x_1}^{q_1} \sum_{k_2=x_2+k_1}^{q_2} \cdots \sum_{k_{l-1}=x_{l-1}+k_{l-2}}^{q_{l-1}} \rho(k_1, \dots, k_{l-1}) \cdot \text{Be}(x_l + k_{l-1}, q_{l-1} - k_{l-1} + n_l - x_l) \quad (6)$$

$$\pi(R_l | \Omega) = \frac{\sum_{k_1=x_1}^{q_1} \sum_{k_2=x_2+k_1}^{q_2} \cdots \sum_{k_{l-1}=x_{l-1}+k_{l-2}}^{q_{l-1}} \rho(k_1, \dots, k_{l-1}) \text{Be}(x_l + k_{l-1}, q_{l-1} - k_{l-1} + n_l - x_l)}{\sum_{k_1=x_1}^{q_1} \sum_{k_2=x_2+k_1}^{q_2} \cdots \sum_{k_{l-1}=x_{l-1}+k_{l-2}}^{q_{l-1}} \rho(k_1, \dots, k_{l-1}) \int_0^1 \text{Be}(x_l + k_{l-1}, q_{l-1} - k_{l-1} + n_l - x_l) dR_l} \quad (8)$$

$$= \sum_{k_{l-1}=x_{l-1}+k_{l-2}}^{q_{l-1}} [C_{k_{l-1}} \text{Be}(x_l + k_{l-1}, q_{l-1} + n_l - x_l - k_{l-1})]$$

$$\text{其中, } C_{k_{l-1}} = \frac{\sum_{k_1=x_1}^{q_1} \sum_{k_2=x_2+k_1}^{q_2} \cdots \sum_{k_{l-2}=x_{l-2}+k_{l-3}}^{q_{l-2}} \rho(k_1, \dots, k_{l-1})}{\sum_{k_1=x_1}^{q_1} \sum_{k_2=x_2+k_1}^{q_2} \cdots \sum_{k_{l-1}=x_{l-1}+k_{l-2}}^{q_{l-1}} \rho(k_1, \dots, k_{l-1})}$$

式(8)为增长测试最后一阶段软件成功率 R_l 的后验分布概率模型。从式(8)可以看出,该模型以 l 组可靠性增长测试数据作为模型输入,描述了最后一阶段软件成功率 R_l 的概率分布规律,对研发阶段末期软件可靠性水平的认识更加全面和客观。

接下来,记 R 为验收阶段的软件可靠性真实水平,设增长测试结束之后没有再获得任何信息,则可以假设 $R = R_l$, R_l

$$\begin{cases} E(R_l | \Omega) = \int_0^1 R_l \cdot \pi(R_l | \Omega) dR_l = \sum_{k_{l-1}=x_{l-1}+k_{l-2}}^{q_{l-1}} \left[C_{k_{l-1}} \cdot \frac{x_l + k_{l-1}}{q_{l-1} + n_l} \right] \\ E[R_l^2 | \Omega] = \int_0^1 R_l^2 \cdot \pi(R_l | \Omega) dR_l = \sum_{k_{l-1}=x_{l-1}+k_{l-2}}^{q_{l-1}} \left[C_{k_{l-1}} \cdot \frac{x_l + k_{l-1}}{q_{l-1} + n_l} \cdot \frac{x_l + k_{l-1} + 1}{q_l + n_l + 1} \right] \end{cases} \quad (10)$$

又因为 $\text{Be}(R; u, v)$ 的一阶矩与二阶矩为:

$$\begin{cases} ER = \frac{u}{u+v} \\ ER^2 = \frac{u(u+1)}{(u+v)(u+v+1)} \end{cases} \quad (11)$$

联立式(10)和式(11)可解得:

$$\begin{cases} u_0 = \frac{\omega_1 - \omega_2}{(\omega_2 - \omega_1^2)} \omega_1 \\ v_0 = (1 - \omega_1) \frac{\omega_1 - \omega_2}{(\omega_2 - \omega_1^2)} \end{cases} \quad (12)$$

其中, $\omega_1 = E(R_l | \Omega)$, $\omega_2 = E[R_l^2 | \Omega]$ 。故验收阶段软件成功率 R 的先验分布 $p(R)$ 就被确定为 $\text{Be}(u_0, v_0)$, 其超参数 u_0, v_0 满足式(12)。

由于两个具有相等一阶矩与二阶矩的概率分布等价于两个分布具有相同的期望和方差,因此采用本节的先验分布近

其中:

$$\begin{cases} \rho(k_1, \dots, k_{l-1}) = \prod_{i=1}^{l-1} C_{q_i}^{k_i} B(x_{i+1} + k_i, n_{i+1} - x_{i+1} + q_i - k_i) \\ q_i = \sum_{j=1}^i (n_j - 1), i=1, 2, \dots, l-1 \\ B(x, y) = \int_0^1 \theta^{x-1} (1-\theta)^{y-1} d\theta \end{cases}$$

式(6)即为 R_l 的边缘分布解析解,不难看出该表达式为有限项贝塔分布的线性组合。

3 软件成功率先验分布的确定

在计算出 R_l 边缘分布的情况下,根据贝叶斯定理可得 R_l 的后验分布为:

$$\pi(R_l | \Omega) = \frac{h(R_l, \Omega)}{\int_0^1 [h(R_l, \Omega)] dR_l} \quad (7)$$

将式(6)代入式(7)可得:

的后验分布就是 R 的先验分布,即:

$$p(R) = \pi(R_l | \Omega) \quad (9)$$

其中, $p(R)$ 为验收阶段软件成功率 R 的先验分布。

然而, $\pi(R_l | \Omega)$ 的表达形式复杂,实际应用困难。本章采用第一工程近似法^[14]给出一种 R 先验分布的近似确定方法。首先采用共轭分布法将 R 的先验分布 $p(R)$ 确定为贝塔分布 $\text{Be}(u, v)$, 其中参数 (u, v) 待估计;然后用 $\text{Be}(u, v)$ 近似拟合 $\pi(R_l | \Omega)$, 要求 $\text{Be}(u, v)$ 与 $\pi(R_l | \Omega)$ 具有相同的一阶矩和二阶矩。根据式(8)容易求得 R_l 的后验一阶矩和二阶矩分别为:

似确定方法,能确保给出的近似分布 $\text{Be}(u_0, v_0)$ 与原始分布在统计形态上具有相同的中心位置和离散程度。

最后,应用给出的近似先验分布 $\text{Be}(u_0, v_0)$ 对 R 进行点估计。在贝叶斯学派的理论中,参数估计被认为是一个决策问题,需要引入损失函数和贝叶斯风险进行参数估计。估计(决策)的优良性评价标准是决策风险,其目标是依据现有信息将所做决策的风险最小化。可以证明,在给定平方损失函数 $L(R, \hat{R}) = (\hat{R} - R)^2$ 下 R 的 Bayes 估计为 $\hat{R} = E(R | \Omega)$ ^[15], 因此,将近似分布 $\text{Be}(u_0, v_0)$ 代入公式可得 R 的 Bayes 点估计为:

$$\hat{R} = \frac{u_0}{u_0 + v_0} \quad (13)$$

式(13)表明,在平方损失函数下,基于近似先验分布

$Be(u_0, v_0)$ 给出的 Bayes 点估计, 能确保其对验收阶段软件成功率 R 的估计的平均损失最小。

4 软件成功率的贝叶斯验证方案

当获得了软件成功率 R 的先验分布之后, 根据贝叶斯公式可得 R 的后验分布为:

$$\pi(R|N, S) = Be(u_0 + S, v_0 + N - S) \quad (14)$$

其中, N 为可靠性验证测试阶段执行的测试用例个数, S 为成功执行的用例个数。

下面考虑可靠性贝叶斯验证方案的制定问题。软件可靠性贝叶斯验证测试方案已有不少研究成果, 如序贯贝叶斯可靠性验证测试方案^[16]、多层贝叶斯可靠性验证测试方案^[17]、基于斐波那契迭代算法的贝叶斯软件可靠性验证测试方案^[18]、基于先验信息动态整合的贝叶斯软件可靠性验证测试方法^[19]等。根据保护的利益方的不同, 验证方案可分为单风险控制方案与双风险控制方案。基于不同的测试的运行方式, 验证方案又可分为固定期方案(测试前规定好所需的测试用例数或测试时间)与非固定期方案(所需测试工作量根据测试情况动态调整), 其中非固定方案实施起来更加灵活, 原理与制定步骤也更加复杂。本节从保护使用方利益的角度制定简单且易理解的固定期验证方案, 便于后续实验结果的直观对比。

假设给定的软件可靠性指标为 (R_0, γ) , R_0 为使用方可接受的最低可靠度下限, γ 为使用方可承受的最大风险, 则一个通过了可靠性验证的软件在使用过程中因实际质量水平不合格而发生失效的概率为 $P\{R < R_0 | N - S \leq c\}$ 。要控制该事件发生的风险, 即要求:

$$P\{R < R_0 | N - S \leq c, N\} \leq \gamma \quad (15)$$

其中, c 为使用方可容忍的最大失效次数。

利用贝叶斯公式, 式(15)可等价变形为:

$$P\{R \geq R_0 | N - S \leq c, N\} = \frac{P\{R \geq R_0, S \geq N - c | N\}}{P\{S \geq N - c | N\}} = \frac{\sum_{S=N-c}^N P\{R \geq R_0 | N, S\} \cdot P\{S | N\}}{\sum_{S=N-c}^N P\{S | N\}} \geq 1 - \gamma \quad (16)$$

其中:

$$\begin{cases} P\{S | N\} = \frac{B(u_0 + S, v_0 + N - S)}{B(u_0, v_0)} \\ P\{R \geq R_0 | N, S\} = \int_{R_0}^1 \pi(R | N, S) dR \\ = \int_{R_0}^1 Be(u_0 + S, v_0 + N - S) dR \end{cases}$$

在可容忍失效次数 c 给定的情况下, 使式(16)成立的最小 N , 即为可靠性验证测试所需的测试用例总数。当软件采取此验证方案通过可靠性验证测试时, 可以保证软件可靠性达到使用方要求的平均概率为 $1 - \gamma$ 。

5 实验

本章分别通过实例和仿真对所提先验分布确定方法和贝叶斯验证方案的有效性进行检验。

5.1 实例验证

某被测软件由 PL/I 和汇编语言编写, 包含约 50 000 行代码, 研发阶段以每日集中排除缺陷的方式开展可靠性增长测试, 记录到的测试数据如表 1 所列^[20]。

表 1 可靠性增长测试数据

Table 1 Reliability growth test data

每日执行用例个数 n_i	成功个数 x_i	失败个数 $(n_i - x_i)$
14	9	5
14	11	3
14	12	2
29	27	2
114	111	3

由表 1 提供的数据知 $\Omega = \{(9, 5), (11, 3), (12, 2), (27, 2), (111, 3)\}$, 软件经历了五阶段可靠性增长, 式(8)计算出最后一阶段软件可靠性水平 R_5 的概率分布为:

$$\begin{aligned} \pi(R_5 | \Omega) &= C_1 Be(R_5; 170, 11) + C_2 Be(R_5; 171, 10) + C_3 Be(R_5; 172, 9) + C_4 Be(R_5; 173, 8) + C_5 Be(R_5; 174, 7) + C_6 Be(R_5; 175, 6) + C_7 Be(R_5; 176, 5) + C_8 Be(R_5; 177, 4) + C_9 Be(R_5; 178, 3) \end{aligned}$$

其中, $C_1 = 0.079, C_2 = 0.097, C_3 = 0.118, C_4 = 0.119, C_5 = 0.153, C_6 = 0.118, C_7 = 0.153, C_8 = 0.074, C_9 = 0.088$ 。构造验收阶段软件成功率 R 的先验分布为 $Be(u, v)$, 用 $\pi(R_5 | \Omega)$ 近似拟合 $Be(u, v)$, 由式(10)计算出样本矩 $w_1 = 0.939, w_2 = 0.882$, 代入式(12)可得 $u = 170, v = 11$, 从而确定 R 的先验分布为 $Be(170, 11)$; 记软件成功率 R 的贝叶斯点估计为 $\hat{R}_{(1)}$, 由式(13)可得 $\hat{R}_{(1)} = 0.939$; 假设规定的可靠性指标 (R_0, γ) 为 $(0.9, 0.01)$, 根据式(16)可给出基于平均后验风险的软件可靠性贝叶斯验证方案, 计算结果如表 2 所列。

表 2 本文所提贝叶斯验证方案

Table 2 Bayesian scheme proposed in this paper

c	$N_{(1)}$
0	17
1	30
2	43
3	56
4	69

接下来, 对所提先验分布确定方法(简称方法 1)进行有效性检验; 以表 1 数据为先验信息, 分别采用基于先验矩的先验分布确定方法(简称方法 2)和无先验信息先验分布确定方法(简称方法 3)给出 R 的先验分布, 记方法 2 和方法 3 给出的成功率点估计为 $\hat{R}_{(2)}, \hat{R}_{(3)}$; 基于表 1 最后一行数据, 计算出增长测试最后一阶段软件成功率的经验样本值 $x_5/n_5 = 0.973$; 考查 $\hat{R}_{(1)}, \hat{R}_{(2)}, \hat{R}_{(3)}$ 与经验样本值的偏差。计算结果如表 3 所列。

从表 3 和图 3 中可以看出 $\hat{R}_{(3)} < \hat{R}_{(2)} < \hat{R}_{(1)}$, 且 $\hat{R}_{(1)}$ 与增长测试最后一阶段的经验样本值最接近, $\hat{R}_{(3)}$ 偏差最大。可

见,所提先验分布确定方法对交付验收前的软件可靠性水平估计最准确。

表3 不同先验分布确定方法的比较
Table 3 Comparison of different prior distribution determination methods

先验分布超参数计算方法	先验分布 $Be(u, v)$	$\hat{R}_{(i)}$	$ \hat{R}_{(i)} - \frac{x_5}{n_5} $
方法1 $\begin{cases} \frac{u}{u+v} = E(R_i \Omega) \\ \frac{u(u+1)}{(u+v)(u+v+1)} = E(R_i^2 \Omega) \end{cases}$	$Be(170, 11)$	0.939	0.034
方法2 $\begin{cases} \frac{u}{u+v} = \frac{1}{l} \sum_{i=1}^l R_i \\ \frac{uv}{(u+v)^2(u+v+1)} = \frac{1}{l-1} \sum_{i=1}^l (R_i - \bar{R})^2 \end{cases}$	$Be(5, 828, 1, 126)$	0.838	0.135
方法3 $u=v=1$	$Be(1, 1)$	0.500	0.473

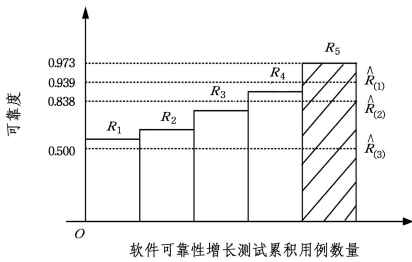


图3 基于不同先验分布确定方法的 R_5 点估计比较
Fig. 3 Comparison of R_5 estimation based on different prior distribution determination methods

最后,对本文所提贝叶斯验证方案($N_{(1)}$)的有效性进行检验:分别采取先验分布确定方法2和方法3同样给出基于平均后验风险的贝叶斯验证方案($N_{(2)}$ 和 $N_{(3)}$),此外还给出经典验证方案($N_{(4)}$)作为对比方案,计算结果如表4所列。

表4 不同可靠性验证方案所需测试用例数量的比较

Table 4 Comparison of test case quantity for different reliability verification schemes

methods	c				
	0	1	2	3	4
$N_{(1)}$	17	30	43	56	69
$N_{(2)}$	41	60	77	93	109
$N_{(3)}$	43	63	80	96	112
$N_{(4)}$	44	64	81	97	113

可以看出,4种验证方案所需测试用例数量大小满足 $N_{(1)} < N_{(2)} < N_{(3)} < N_{(4)}$,在c等于0的无失效验证情况下,本文所提贝叶斯方案所需用例数量比贝叶斯方案2即 $N_{(2)}$ 缩减了58%,比贝叶斯方案3即 $N_{(3)}$ 缩减了60%,比经典方案 $N_{(4)}$ 缩减了61%。

5.2 仿真实验

本节采用仿真方法生成仿真软件可靠性增长测试失效数据,对所提方法的有效性做进一步验证。

假设某软件在研发阶段可靠性增长测试期内经历了6次缺陷集中移除,各次缺陷移除相互独立。记 n_i 为第i阶段测试执行用例总数, R_i 为第i阶段软件可靠性水平, $i=1,2,\dots,6$,第i阶段成功执行测试用例个数服从二项分布 $B(n_i, R_i)$,设参数 $R_1=0.65, R_2=0.75, R_3=0.85, R_4=0.9, R_5=0.95, R_6=0.98, n_1=30, n_2=30, n_3=30, n_4=60, n_5=85, n_6=300$,采用可靠度仿真方法^[21],用matlab软件binornd(n,p)命令随

机生成第i阶段成功执行用例个数 $x_i, i=1,2,\dots,6$ 。重复仿真10000次消除随机误差,得到仿真可靠性增长测试失效数据,如表5所列。

表5 仿真可靠性增长测试失效数据

Table 5 Failure data of simulation reliability growth test

每阶段测试执行用例总数 n_i	仿真成功用例个数 x_i	仿真失效用例个数 $(n_i - x_i)$
30	20	10
30	23	7
30	26	4
60	56	4
85	81	4
300	294	6

以表5仿真失效数据为先验信息,利用式(9)可计算出 $\pi(R_6 | \Omega)$,将 $\pi(R_6 | \Omega)$ 代入式(12)解得 $u_0=410, v_0=14$,从而将R的先验分布确定为 $Be(410, 14)$,再利用式(13)可得贝叶斯点估计 $\hat{R}_{(1)}' = 0.967$;假设规定的可靠性指标(R_0, γ)为(0.95, 0.01),根据式(16)可以给出软件可靠性贝叶斯验证方案。计算结果如表6所列。

表6 基于仿真失效数据的贝叶斯验证方案

Table 6 Bayesian validation scheme based on simulation failure data

c	$N_{(1)}'$
0	55
1	81
2	106
3	132
4	157

接下来采用先验分布确定方法2和方法3给出R的点估计,并记估计值为 $\hat{R}_{(2)}', \hat{R}_{(3)}'$,通过考查 $\hat{R}_{(1)}', \hat{R}_{(2)}', \hat{R}_{(3)}'$ 与最后一阶段可靠性参数 R_6 的离差,对先验分布确定方法的有效性进行检验,计算结果如表7所列。

表7 基于仿真失效数据的先验分布确定方法的比较

Table 7 Comparison of prior distribution determination methods based on simulation failure data

先验分布 $Be(u, v)$	估计值 $\hat{R}_{(i)}'$	真值离差 $ \hat{R}_{(i)}' - R_6 $
方法1 $Be(410, 14)$	0.967	0.013
方法2 $Be(6, 049, 0, 976)$	0.861	0.119
方法3 $Be(1, 1)$	0.500	0.480

从表 7 中可以看出,方法 3 给出的估计值与最后一阶段软件可靠度参数 R_0 的偏差最大,所提先验分布确定方法给出的可靠度估计值最接近,估计效果最准确。该验证结果与 5.1 节实例验证结果一致。

最后,采取先验分布确定方法 2 和方法 3 制定基于平均后验风险的贝叶斯方案,并给出经典验证方案,通过比较方案所需测试用例数量对方案的有效性进行检验,计算结果如表 8 所列。

表 8 基于仿真失效数据的验证方案测试用例数量比较

Table 8 Comparison of the number of test cases for verification schemes based on simulation failure data

methods	c				
	0	1	2	3	4
$N_{(1)}'$	55	81	106	132	157
$N_{(2)}'$	83	124	159	191	222
$N_{(3)}'$	89	129	164	197	228
$N_{(4)}'$	90	130	165	198	229

从表 8 中可以看出,4 种验证方案所需测试用例数量满足 $N_{(1)}' < N_{(2)}' < N_{(3)}' < N_{(4)}'$,在 c 等于 0 的无失效验证情况下,本文所提贝叶斯方案所需用例数量最少,比贝叶斯方案 2 即 $N_{(2)}'$ 缩减了 34%,比贝叶斯方案 3 即 $N_{(3)}'$ 缩减了 38%,比经典方案 $N_{(4)}'$ 缩减了 39%。

5.3 实验结果分析

本节选择了两种典型的先验分布确定方法与所提方法进行对比。经实例与仿真验证,所提方法对增长测试最后一阶段软件可靠性水平的估计最准确,基于先验矩的先验分布确定方法次之,无先验信息先验分布确定方法的估计偏差最大。此外,还将基于这两种先验分布确定方法制定的贝叶斯方案以及经典方案与本文所提贝叶斯方案进行对比,实例与仿真均得出了一致的验证结论,即 3 种贝叶斯方案所需用例数量均少于不利用任何先验信息的经典验证方案,由此说明了贝叶斯方案在缩减验证试验量方面的积极意义。不仅如此,在贝叶斯方案中,无先验信息的贝叶斯方案所需的测试用例数量最多,基于先验矩法的贝叶斯方案次之,所提方案所需的测试用例数量最少,由此说明了本文所提贝叶斯方案在缩减用例数量方面的优势。

出现以上实验结果的原因是,由 Bayesian 和 Laplace 提出的无先验信息先验分布确定法^[22]假定人们对产品的历史情况没有任何了解,故将可靠性参数的先验分布确定为均匀分布。对于有着丰富可靠性增长测试信息的软件产品而言,这种方法浪费了大量的历史信息,对软件可靠性水平的评价有限,因此估计结果与软件可靠性真实水平的偏差最大。基于先验矩的先验分布确定方法是一种应用广泛的方法^[23],该方法充分利用了产品历史可靠性增长测试数据,且计算方法简单便捷,但只简单取各阶段测试样本的平均值来估计最后一阶段产品的可靠性,对于有着分阶段增长先验信息的软件产品而言,这种信息加工方法比较粗糙,会造成估计结果低于增长测试最后一阶段软件可靠性水平的真实值,估计结果偏保守。所提先验分布确定方法不仅利用了各阶段的历史可靠性增长测试数据,而且给出了在顺序约束条件下最后一阶段软件可靠性 R_i 的概率分布模型,对样本分阶段增长趋势的刻

画更准确,因此估计结果更接近真实数据。经分析,3 种先验分布确定方法给出的 R 点估计值大小应满足 $\hat{R}_{(3)} < \hat{R}_{(2)} < \hat{R}_{(1)}$,且基于另两种先验分布确定方法制定的贝叶斯方案所需测试用例数量均会大于所提贝叶斯方案所需测试用例数量。由此可见,实验结果与理论分析结论相符。

结束语 本文以成功率为可靠性指标的软件产品为研究对象,给出了软件成功率的先验分布确定方法。在此基础上,从保护使用方利益的角度设计了基于最大后验风险的贝叶斯可靠性验证方案。

先验分布的确定是贝叶斯验证方案的关键,其本质是一种先验信息加工方法。加工方法偏保守会造成贝叶斯方案所需测试用例数量大,可靠性验证工作实施困难,而加工方法偏激进又会影响验证方案的可信性。本文针对软件历史可靠性增长测试数据呈分阶段增长的特征,利用概率统计理论给出最后一阶段软件成功率的概率分布模型,所提先验分布确定方法对先验信息的加工更合理,其能在保证验证方案可信性的前提下减轻测试负担,改善软件可靠性验证工作实施困难的问题,因此具有一定的经济价值。

参考文献

- [1] HSU J, Y, JIANG T Y, CHAO P C P. A fast FPGA hardware accelerator for remote heart rate detection based on RGB vision [J]. IEEE Transactions on Biomedical Circuits and Systems, 2024, 18: 592-607.
- [2] ZORMATI M A, LAKHLEF H, OUNI S. Review and analysis of recent advances in intelligent network softwarization for the Internet of Things [J]. Computer Networks, 2024, 241: 110215.
- [3] MA Z Y, ZHANG W, WU W, et al. Overview of software reliability demonstration testing research method [J]. Journal of Ordnance Equipment Engineering, 2019, 40(7): 118-123.
- [4] SHEN X M, WU L J, ZHAN H Y, et al. Research on reliability verification and evaluation technology of ship equipment software [J]. Computer Measurement & Control, 2020, 28(4): 232-236.
- [5] MA Z Y. Software reliability demonstration testing scheme of prior dynamic integration bayesian method based on the idea of decreasing function [J]. International Journal of Performability Engineering, 2018, 14(12): 3087-3097.
- [6] GRUNDLER A, DAZER M, HERZIG T. Statistical power analysis in reliability demonstration testing: the probability of test success [J]. Applied Sciences, 2022, 12: 6190.
- [7] AFSHARI S S, ENAYATOLLAHI F, XU X Y, et al. Machine learning-based methods in structural reliability analysis: a review [J]. Reliability Engineering & System Safety, 2022, 219: 108223.
- [8] WANG J J, YANG G K, FENG Z B. Bayesian modeling and optimization based on multiple life distribution models [J]. Journal of Systems Engineering, 2023, 38(6): 880-890.
- [9] CAI Y F, WU Y H, ZHOU J Y, et al. Quantitative software reliability assessment methodology based on Bayesian belief networks and statistical testing for safety-critical software [J]. An-

- nals of Nuclear Energy,2020,145:107593.
- [10] LIANG D J. Design and Implementation of Process Control and Management System for Military Software Testing and Measure [D]. Changsha: National University of Defense Technology, 2019.
- [11] GIANG P H. A new axiomatization for likelihood gambles[C]// Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence(UAI 2006). 2006:192-199.
- [12] WANG K SHI X J, XIAO Z C, et al. Restricted fault sample size determination method based on posterior distribution[J]. Journal of Vibration, Measurement & Diagnosis, 2020, 40(6): 1156-1164, 1234-1235.
- [13] LEI H J, QIN K Y. Bayesian method for determination of testability demonstration test scheme[J]. Systems Engineering and Electronics, 2012, 34(12): 2612-2616.
- [14] ZHOU Y Q, LI B S, ZHANG Z P, et al. Reliability Statistical Analysis [M]. Beijing: Science Press, 2017: 87-88.
- [15] MAO S S, WANG J L, PU X L. Advanced Mathematical Statistics (Third Edition) [M]. Beijing: Higher Education Press, 2021: 386-387.
- [16] STEFAN A M, SCHÖNBRODT F D, EVANS N J, et al. Efficiency in sequential testing: Comparing the sequential probability ratio test and the sequential Bayes factor test[J]. Behavior Research Methods, 2022, 54: 3100-3117.
- [17] LI J X, LI Z Y, JIANG G J. Reliability estimation of ELNs based on multi-layer Bayesian theory[C]// 12th International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering(QR2MSE 2022); Institution of Engineering and Technology. 2022: 776-781.
- [18] MA Z Y WU W, ZHANG W, et al. Bayesian software reliability demonstration testing scheme based on fibonacci iteration algorithm[J]. Journal of Academy of Armored Force Engineering, 2017, 31(4): 116-120.
- [19] JIA Y, MEI F, SUN P. Software reliability verification simulation based on dynamic integration of prior information[J]. Computer Simulation, 2023, 40(10): 377-380, 430.
- [20] CAI K Y. Towards a conceptual framework of software run reliability modeling[J]. Information Sciences, 2000, 126: 137-163.
- [21] FU H M, WEN X L, YANG H F. Test method for reliability simulation[J]. Intelligent Computer and Applications, 2019, 9(5): 7-12.
- [22] MAO S S, TANG Y C. Bayesian Statistics (Second Edition) [M]. Beijing: China Statistical Press, 2012: 52-53.
- [23] WU Y M, YANG R S, LI H F, et al. Bayesian theory based software reliability demonstration test method for safety critical software[J]. Mathematical Structures in Computer Science, 2014, 24: 1-22.



WANG Yuzhuo, born in 1984, Ph.D, associate professor. Her main research interest is software reliability evaluation.



LIU Haitao, born in 1982, Ph.D, professor. His main research interest is reliability theory and application.

(责任编辑:何杨)