

Weibull 分布引进故障的软件可靠性增长模型*

王金勇¹, 张策², 米晓萍¹, 郭新峰¹, 李济洪¹

¹(山西大学 软件学院, 山西 太原 030006)

²(哈尔滨工业大学(威海) 计算机科学与技术学院, 山东 威海 264209)

通讯作者: 王金勇, E-mail: wangjinyong818@163.com



摘要: 软件调试是复杂过程,可能会受到很多种因素的影响,例如调试资源分配、调试工具的使用情况、调试技巧等.在软件调试过程中,当检测到的故障被去除时,新的故障可能会被引进.因此,研究故障引进的现象对建立高质量的软件可靠性增长模型具有重要意义.但是到目前为止,模拟故障引进过程仍是一个复杂和困难的问题.虽然有许多研究者开发了一些不完美调试的软件可靠性增长模型,但是一般都是假设故障内容(总数)函数为线性、指数分布或者是与故障去除的数量成正比.这个假设与实际的软件调试过程中故障引进情况并不完全一致.提出一种基于 Weibull 分布引进故障的软件可靠性增长模型,考虑故障内容(总数)函数服从 Weibull 分布,并用相关的实验验证了提出的模型的拟合和预测性能.在用两个故障数据集进行的模拟实验中,实验结果指出:提出的模型和其他模型相比,有更好的拟合和预测性能以及更好的鲁棒性.

关键词: 软件可靠性;软件可靠性增长模型;非齐次泊松过程;不完美调试;Weibull 分布的故障内容函数

中图法分类号: TP311

中文引用格式: 王金勇,张策,米晓萍,郭新峰,李济洪. Weibull 分布引进故障的软件可靠性增长模型. 软件学报, 2019, 30(6): 1759-1777. <http://www.jos.org.cn/1000-9825/5427.htm>

英文引用格式: Wang JY, Zhang C, Mi XP, Guo XF, Li JH. A software reliability growth model based on Weibull distribution introduced faults. Ruan Jian Xue Bao/Journal of Software, 2019, 30(6): 1759-1777 (in Chinese). <http://www.jos.org.cn/1000-9825/5427.htm>

Software Reliability Growth Model Based on Weibull Distribution Introduced Faults

WANG Jin-Yong¹, ZHANG Ce², MI Xiao-Ping¹, GUO Xin-Feng¹, LI Ji-Hong¹

¹(School of Software Engineering, Shanxi University, Taiyuan 030006, China)

²(School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai 264209, China)

Abstract: Software debugging is a complex process and affected by many factors, such as debugging resources, debugging tools, debugging skills, etc. When detected faults were removed, new faults may be introduced. Therefore, it plays an important role to research an imperfect debugging phenomenon in the software debugging process. How to model fault introduction in building an imperfect debugging model is still an unresolved issue. So far, numerous software debugging models are developed by researchers, for example, assuming the fault content function is a linear, exponential distribution or proportional to the number of removed faults, etc. However, they can not entirely satisfy the realistic needs due to fault introduction complicated changes over time. In this study, an NHPP software reliability model is proposed based on Weibull distribution introduced faults and the fault content function following Weibull distribution is considered. The related experiment is carried out which validates the fitting and predictive power of the proposed model. The experimental results also show the proposed model has much better fitting and predictive performance than other models using two fault data sets, as well as better robustness.

* 基金项目: 山西省自然科学基金(201801D121120); NSFC-广东联合基金(U1501501); 山西省软科学研究项目(2017041039-6)

Foundation item: Natural Science Foundation of Shanxi Province of China (201801D121120); NSFC-Guangdong Joint Fund Key Support Program of China (U1501501); Soft Science Research Projects of Shanxi Province of China (2017041039-6)

收稿时间: 2017-04-17; 修改时间: 2017-07-17, 2017-09-12, 2017-09-30; 采用时间: 2017-10-13

Key words: software reliability; software reliability growth model (SRGM); non-homogeneous Poisson process (NHPP); imperfect debugging; Weibull-distribution fault content function

软件测试和调试是一个复杂的过程,其中,测试者和调试者的能力和技术对软件故障的检测和排错有着重要影响.当一个故障被检测出来时,调试人员需要用相关的知识去考虑怎样完全去除和修复故障,而且不要引入新的故障.但是,调试过程会被许多因素影响,例如调试者的技术、调试用的工具、调试环境和调试人员在调试过程中的心理变化等.这些因素都会影响调试人员在去除故障时是否会引入新的故障.人们一般称在软件调试过程中,当检测出的故障被去除时,引进新的故障现象为不完美调试.

在过去的 40 年里,研究者对不完美调试现象进行相关的研究,而且也提出了许多不完美调试的软件可靠性增长模型.由于不完美调试情况的复杂性,人们一般把不完美调试过程看做是具有不确定性的随机过程.另外,在实际的软件测试过程中,软件检测出故障的数量不一定和故障去除的故障数量相等,例如,去除一个已检测出的故障可能会引入新的故障或者原来的故障没有被完全去除.也就是说,在不完美调试过程中,故障总个数可能会增加.G-O^[1]首先提出了不完美调试的概念,Obha 等人^[2]在 G-O 模型的基础上提出了故障引进的不完美调试软件可靠性模型.Kapur 等人^[3]提出了故障检测率随测试时间下降的不完美调试软件可靠性增长模型.Pham 等人^[4,5]通过考虑线性引进故障现象和合并多失效类型的故障生成过程,提出了相应的不完美调试的软件可靠性模型.Zhang 等人^[6]和 Kapur 等人^[7]分别就软件调试过程中故障引进的不同形式提出了相应的软件可靠性模型.另外,Kapur 等人^[8]通过研究故障检测和故障去除之间的不同分布情况,提出了两个一般的框架(framework),并得出了几个具有 NHPP 类的不完美调试的软件可靠性增长模型.Wang 等人^[9-11]通过对故障引进现象的研究,分别提出了故障引进非线性变化和故障引进了先增后减的不完美调试的软件可靠性增长模型.另外,王金勇等人^[12]还提出了在软件测试过程中,故障引进率不规则变化的完美调试的软件可靠性增长模型.谢景燕等人^[13]通过对不完美调试过程中引进故障和不完全去除故障的研究,提出了故障排错率随测试时间变化的不完美调试的软件可靠性模型.

虽然上面提到的关于不完美调试软件可靠性模型在实际测试过程中能够有效地评估软件可靠性和预测软件中存在的剩余故障的数量,但是没有哪一种模型能够应用到软件测试的所有环境下^[14].因此,还需要对软件调试过程中故障引进现象进行深入研究.实际上,故障引进不但可能是线性^[4]、指数分布^[15,16]或者与故障排错的数量成正比^[7],而且还可能是更复杂的变化,可能受到调试过程中的环境、工具以及调试者的能力和技术等影响.因此,故障引进在软件调试过程中是一个需要重点考虑的因素.

在本文中,我们用 Weibull 分布来模型化故障引进过程.Weibull 分布有许多优点:第 1,与指数分布具有无记忆性相比,它具有记忆性;第 2,它可以模型化多种故障数据集,例如模型化左倾(left-skewed)、右倾(right-skewed)或者对称数据(symmetric data);第 3,当形状参数变化时,它可以有不同的形状;第 4,它可以模型化故障引进强度函数随时间的变化,例如故障引进的强调函数随时间具有先增后减的变化或者下降的变化等.因此,考虑在实际的软件调试过程中,用 Weibull 分布来模型化故障引进随测试时间变化的过程,是一种有效的和可行的方法.

另外,用 Weibull 分布来模型化故障引进现象,与其他方法相比还有其他优势.例如,用 Weibull 分布模型化故障引进的现象,可以考虑故障引进率随测试时间先增后减和逐渐减少的变化趋势.但是其他一般只能模型化故障引进的一种变化.因此,用其他方法建立的软件可靠性模型对软件测试和调试的环境适应性具有一定的限制.而用 Weibull 模型化故障引进过程建立的软件可靠性模型则能更好地适应软件测试和调试的复杂变化.本文中的实验结果也证明了提出的模型和其他不完美调试的软件可靠性模型相比,具有更准确的故障拟合和故障预测性能,而且提出的模型能更好地适应不同软件测试和调试环境下.

1 故障引进服从 Weibull 分布的原因和表示

(1) 缩略语

- NHPP:非齐次泊松过程;

- SRGM:软件可靠性增长模型;
- MLE:最大似然估计;
- MVF:均值函数;
- SSE:误差平方和;
- AIC:Akaike 信息标注.

(2) 符号说明

- a :期望引进故障的总数量;
- b :故障检测率;
- c :故障检测率的上边界;
- d :形状参数;
- p :故障去除效率;
- α :率参数;
- β :拐点因子;
- r :常量故障检测率;
- α_1 :比例参数;
- β_1 :形状参数;
- n :实际观测到的故障数量;
- C :期望检测出最初故障总数量;
- $N(t)$:随机变量,即,表示到 t 时刻为止,检测出故障的数量;
- $a(t)$:故障内容(总数)函数;
- $b(t)$:故障检测率函数;
- o_i :表示到 t_i 时刻为止,实际观测到故障的数量;
- $m(t_i)$:均值函数,表示到 t_i 时刻为止,期望检测出故障的数量.

1.1 Weibull分布相关概念和表示方法

Weibull 分布是一个连续的概率分布,以瑞典工程师、数学家 Waloddi Weibull 的名字命名.

(1) Weibull 概率密度函数可以表示为

$$f(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} \exp\left[-\left(\frac{x}{\lambda}\right)^k\right], & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (1)$$

其中, $f(x; \lambda, k)$ 是概率密度函数, x 是随机变量, λ 是比例参数, k 是形状参数.

Weibull 概率密度函数有下列特点.

- ① 当 $k < 1$ 时,Weibull 概率密度函数曲线在随机变量 x 逐渐增加时,有急剧下降的趋势;
- ② 当 $k = 1$ 时,Weibull 概率密度函数曲线在随机变量 x 逐渐增加时,有逐渐缓慢下降的趋势;
- ③ 当 $k > 1$ 时,Weibull 概率密度函数曲线在随机变量 x 逐渐增加时,有先增后降的趋势.

从图 1 能够清晰地看到这种变化.考虑到 Weibull 概率密度函数的这些特点,可以用来模拟故障引进率先增后减和逐渐下降的过程.

(2) Weibull 累计分布函数可以表示为

$$F(x; \lambda, k) = \begin{cases} 1 - \exp\left[-\left(\frac{x}{\lambda}\right)^k\right], & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

其中, $F(x; \lambda, k)$ 为 Weibull 累计分布函数.

Weibull 累计分布函数有下列特点.

- ① 当 $k=1$ 时,它为指数分布(exponential distribution);
- ② 当 $k=2, \lambda = \sqrt{2}\lambda$ 时,它为瑞利分布(rayleigh distribution).

从图 2 能够清晰地看到 Weibull 累计分布函数在随机变量 x 逐渐增加时发生的相应变化.考虑到 Weibull 累计分布函数的特点,因此可以用来模拟故障引进过程.

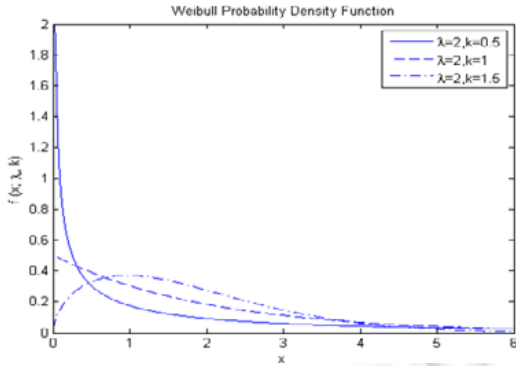


Fig.1 Graph of Weibull probability density function

图 1 Weibull 概率密度函数图

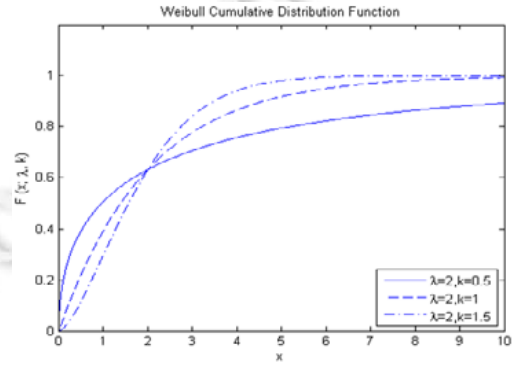


Fig.2 Graph of Weibull cumulative distribution function

图 2 Weibull 累计分布函数图

1.2 故障引进服从Weibull分布的原因

为了简化已建立不完美调试软件可靠性增长模型的复杂性,一般都是假设软件调试过程中,故障引入率或者为常数,或者故障引入服从指数分布.但在实际软件调试中,故障引入率有可能不为常数,故障引入可能会受到许多因素的影响,例如调试者的技术和调试的环境等.软件测试是一个复杂的过程,受到很多外部或内部的因素的影响,不可能建立一种模型把所有的故障引入的因素都考虑进去.因此,综合考虑故障引入的变化,在实际的软件调试过程中更有一定的实际意义.

另一方面,Weibull 分布有许多优点,例如可以模拟多种其他分布函数形状,通过 Weibull 参数的变化,有效地表示实际软件调试过程中故障引入的复杂变化.所以,用 Weibull 分布函数来建立不完美调试条件下的软件可靠性模型还是有一定的实际意义.并且随着人们对软件可靠性建模的不断深入了解和认识,综合考虑故障引入条件,并通过 Weibull 分布函数来建立相应的不完美软件可靠性模型,则更符合软件可靠性调试的实际情况.

在实际软件调试过程中,故障引入会发生复杂变化,考虑用其他分布函数来模拟故障引入的情况,不能完全反映故障引入的随机变化.虽然有的不完美调试软件可靠性增长模型考虑了指数分布的故障引入情况,但是由于指数分布具有无记忆性,即 $P(s+t|t)=P(t)$,和发生的时间 s 无关.这一性质对于故障引入情况来说,是不符合故障引入的实际变化.也就是说,在实际的调试过程中,故障引入发生的概率会随着测试时间逐渐变小,而不会与以前的故障引入发生概率相等.由于 Weibull 分布函数没有无记忆性的性质,因此,用 Weibull 分布函数来模拟故障引入的过程会更有效地反映实际的软件调试过程中故障引入情况.另外,它还可以模拟故障引进率随测试时间下降、不变和先增后减的变化.

因此,考虑用 Weibull 分布来模拟故障引进过程是可行、合理的.本文的实验结果也证明:用 Weibull 分布来模拟故障引进随测试时间变化的情况,是建立高质量的软件可靠性增长模型最好的方法之一.

1.3 用Weibull分布模拟故障引进过程

从以上的分析可以得出,故障引进情况可以用 Weibull 分布函数来表示.因此,故障内容(总数)函数可以被表示为

$$a(t)=aF(t)+C \quad (3)$$

其中, a 表示最终期望引进故障的总数量, $F(t)$ 是故障引进的分布函数, C 是最初期望的软件内存在的故障数量, $a(t)$ 是故障内容(总数)函数.

故障引进强度函数可以表示为

$$\eta(t)=aF'(t) \quad (4)$$

假设故障内容(总数)函数服从 Weibull 分布,即故障内容(总数)函数 $a(t)$ 与故障引进强度函数可分别表示为

$$a(t)=aF(t)+C=a[1-\exp(-\alpha t^d)]+C \quad (5)$$

$$\eta(t)=aF'(t)=\alpha a d t^{d-1} \exp(-\alpha t^d) \quad (6)$$

其中, α 为率参数, d 为形状参数.

考虑到故障引进过程主要是考察故障引进率随测试时间发生变化的情况,因此,故障引进强度函数可以改写为以下形式:

$$\eta(t)=[a+C-a(t)]\frac{F'(t)}{1-F(t)}=[a+C-a(t)]h(t) \quad (7)$$

$$h(t)=\frac{F'(t)}{1-F(t)}=\alpha d t^{d-1} \quad (8)$$

其中, $h(t)$ 表示故障引进率函数; $[a+C-a(t)]$ 表示到 t 时刻为止,期望剩余故障引进故障的数量.

在公式(7)中, $[a+C-a(t)]$ 是一个随时间增长而非增长的函数.因此,故障引进强度函数的变化趋势是由故障引进率的变化决定的.例如在公式(8)中:当 $d=1$ 时,公式(7)意味着在 $(t,t+\Delta t)$ 时间内引进故障的数量和剩余故障引进故障的数量成正比,即故障引进率为常数;当 $d<1$ 时,故障引进率函数 $h(t)$ 随测试时间逐渐下降;当 $d>1$ 时,故障引进率函数 $h(t)$ 随测试时间有逐渐增加的变化趋势.

2 提出的不完美调试模型的建立

2.1 提出模型的基本假设

提出模型的假设条件如下.

- ① 软件失效和故障去除过程遵循非齐次泊松过程(NHPP);
- ② 软件失效发生是随机发生的,发生的原因是由软件中剩余故障造成的;
- ③ 每次检测到一个软件故障,立即被去除,并且可能引进新的故障;
- ④ 故障内容(总数)函数服从 Weibull 分布.

从上面的假设可以看出,假设③和假设④捕获了在软件调试过程中,故障引进过程所表现出的变化.

2.2 提出模型建立过程

一般来说,基于 NHPP 的模型都是假设软件失效是随机发生的,在 $(t,t+\Delta t)$ 时间内,期望软件发生失效的数量和期望软件剩余故障发生的数量成正比^[17].这意味着软件故障检测率为常量,它可以用以下微分方程来表示:

$$\frac{dm(t)}{dt}=b(t)[a(t)-m(t)] \quad (9)$$

其中, $m(t)$ 表示均值函数, $b(t)$ 表示故障检测率, $a(t)$ 表示故障内容(总数)函数.

基于假设③和假设④,故障内容(总数)函数可以表示为

$$a(t)=a[1-\exp(-\alpha t^d)]+C \quad (10)$$

这里的 a 为期望最终引进故障的总的数量, α 为率参数, d 为形状参数, C 为期望最初软件中存在的故障的数量.当 $t=0$, $a(0)=C$;当 $t \rightarrow \infty$, $a(\infty)=a+C$.公式(10)两边求导,得:

$$\frac{da(t)}{dt}=\alpha a d t^{d-1} \exp(-\alpha t^d) \quad (11)$$

从公式(11)可以得到:

- ① 当 $d \leq 1$ 时,故障引进随测试时间逐渐减少;

② 当 $d > 1$ 时,故障引进随测试时间有先增后减的变化.

因此,从公式(11)可以得出,故障引进随测试时间发生变化的情况.

当 $b(t)=b$,把公式(10)带入公式(9)可得到:

$$\frac{dm(t)}{dt} = b[a(1 - \exp(-\alpha t^d)) + C - m(t)] \tag{12}$$

因为公式(12)是一个复杂的微分方程,可以得到近似解为

$$m(t) = a[1 - \exp(-\alpha t^d)] - a\alpha d \sum_{i=0}^k \frac{b^i t^{d+i}}{d+i} \exp(-bt) + C[1 - \exp(-bt)] \tag{13}$$

当 $k=5$ 时:

$$m(t) = a[1 - \exp(-\alpha t^d)] - a\alpha d \sum_{i=0}^5 \frac{b^i t^{d+i}}{d+i} \exp(-bt) + C[1 - \exp(-bt)] \tag{14}$$

公式(13)就是提出的模型的表达式,公式(14)是所用到的提出模型的表达式.从公式(13)和公式(14)可以得到:当 $t=0, m(0)=0$;当 $t \rightarrow \infty, m(\infty)=a+C$.

2.3 提出的模型证明

证明:假设 $D(t) = \int_0^t b(x)dx$ 并且 $b(x)=b$,那么 $\exp[D(t)]=\exp(bt)$.公式(9)两边乘以 $\exp[D(t)]$,可得:

$$\exp[D(t)]dm(t) = \exp[D(t)]b(t)[a(t) - m(t)]dt \tag{15}$$

$$\exp[D(t)]dm(t) + \exp[D(t)]b(t)m(t)dt = \exp[D(t)]b(t)a(t)dt \tag{16}$$

公式(15)两边积分得:

$$\int \exp[D(t)]dm(t) + \int \exp[D(t)]b(t)m(t)dt = \int \exp[D(t)]b(t)a(t)dt \tag{17}$$

$$\int d(\exp(D(t))m(t)) = \int \exp[D(t)]b(t)a(t)dt \tag{18}$$

当 $b(t)=b$ 和 $a(t)=a[1 - \exp(-\alpha t^d)] + C$,公式(18)可以转换成以下形式:

$$\left. \begin{aligned} \exp(bt)m(t) &= \int a(t)d(\exp(bt)) \\ m(t) &= \{a(t)\exp(bt) - \int \exp(bt)d(a(t))\} / \exp(bt) \\ m(t) &= a(t) - \int \exp(bt)d(a(t))\exp(-bt) \\ m(t) &= a[1 - \exp(-\alpha t^d)] + C - a\alpha d \exp(-bt) \int \exp(bt)\exp(-\alpha t^d)t^{d-1}dt \\ m(t) &= a[1 - \exp(-\alpha t^d)] + C - a\alpha d \exp(-bt) \int \exp(bt - \alpha t^d)t^{d-1}dt \end{aligned} \right\} \tag{19}$$

因为麦克劳林公式为

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \dots + \frac{f^{(n)}(0)}{n!}x^n + o(x^n) \tag{20}$$

所以可以用公式(20)简化公式(19),可得:

$$\exp(bt - \alpha t^d) = \sum_{i=0}^k b^i t^i \tag{21}$$

$$m(t) = a[1 - \exp(-\alpha t^d)] + C - a\alpha d \exp(-bt) \int t^{d-1} \sum_{i=0}^k b^i t^i dt \tag{22}$$

$$m(t) = a[1 - \exp(-\alpha t^d)] + C - a\alpha d \exp(-bt) \int \sum_{i=0}^k b^i t^{d+i-1} dt \tag{23}$$

$$m(t) = a[1 - \exp(-\alpha t^d)] + C - a\alpha d \exp(-bt) \left(\sum_{i=0}^k \frac{b^i t^{d+i}}{d+i} + C_1 \right) \tag{24}$$

其中, C_1 为一个常数.当 $t=0, m(0)=0$ 时,把它代入公式(24),可得:

$$m(t) = a[1 - \exp(-\alpha t^d)] - a\alpha d \sum_{i=0}^k \frac{b^i t^{d+i}}{d+i} \exp(-bt) + C[1 - \exp(-bt)] \tag{25}$$

取 $k=5$ 时,公式(25)可得:

$$m(t) = a[1 - \exp(-\alpha t^d)] - a\alpha d \sum_{i=0}^5 \frac{b^i t^{d+i}}{d+i} \exp(-bt) + C[1 - \exp(-bt)] \tag{26}$$

3 模拟实验

为了充分、有效地评估提出模型的拟合和预测性能,本节给出模型比较标准和模型参数的估计方法.我们还用两个故障数据集进行相应的仿真实验和模型的置信区间分析.表 1 列出了在本文中进行比较的所有模型.

Table 1 Summary of some SRGMs and their $m(t)$

表 1 软件可靠性增长模型和 $m(t)$

| 模型名称 | 模型的均值函数($m(t)$) |
|---|--|
| Goel-Okumoto(G-O)模型 ^[17] | $m(t)=a[1-\exp(-bt)],a(t)=a,b(t)=b$ |
| Delayed S-shape(DSS)模型 ^[18] | $m(t) = a[1 - (1 + bt)\exp(-bt)], a(t) = a, b(t) = \frac{b^2 t}{1 + bt}$ |
| Inflection S-shape(ISS)模型 ^[19] | $m(t) = \frac{a[1 - \exp(-bt)]}{1 + \beta \exp(-bt)}, a(t) = a, b(t) = \frac{b}{1 + \beta \exp(-bt)}$ |
| Yamada 不完美调试模型-1 ^[15] | $m(t) = \frac{ab}{\alpha + b} [\exp(\alpha t) - \exp(-bt)], a(t) = a \exp(\alpha t), b(t) = b$ |
| Yamada 不完美调试模型-2 ^[15] | $m(t) = a[1 - \exp(-bt)] \frac{b - \alpha}{b} + a\alpha t, a(t) = a(1 + \alpha t), b(t) = b$ |
| P-N-Z 模型 ^[4] | $m(t) = \frac{a \left[(1 - \exp(-bt)) \left(1 - \frac{\alpha}{b} \right) + \alpha t \right]}{1 + \beta \exp(-bt)}, a(t) = a(1 + \alpha t), b(t) = \frac{b}{1 + \beta \exp(-bt)}$ |
| P-Z 模型 ^[16] | $m(t) = \frac{1}{(1 + \beta \exp(-bt))} \left[(C + a)(1 - \exp(-bt)) - \frac{ab}{b - \alpha} (\exp(-\alpha t) - \exp(-bt)) \right],$ $a(t) = C + a(1 - \exp(-\alpha t)), b(t) = \frac{b}{1 + \beta \exp(-bt)}$ |
| Zhang-Teng Pham 模型 ^[6] | $m(t) = \frac{a}{p - \alpha} \left[1 - \left(\frac{(1 + \beta) \exp(-bt)}{1 + \beta \exp(-bt)} \right)^{\frac{c}{p - \alpha}} \right], a(t) = a + \alpha m(t), b(t) = \frac{c}{1 + \beta \exp(-bt)}$ |
| Pham Zhang IFD 模型 ^[20] | $m(t) = a - a \exp(-bt) (1 + (b + \alpha)t + b\alpha t^2), b(t) = \left(\frac{b^2 t}{1 + bt} - \frac{\alpha}{1 + \alpha t} \right)$ |
| Kapur 模型-1 ^[7] | $m(t) = \frac{a}{1 - \alpha} [1 - \exp(-bp(1 - \alpha)t)]$ |
| Kapur 模型-2 ^[8] | $m(t) = \frac{a}{1 - \alpha} \left[1 - \left(\frac{(1 + \beta) \exp(-bt)}{1 + \beta \exp(-bt)} \right)^{p(1 - \alpha)} \right]$ |
| 提出的模型 | $m(t) = a[1 - \exp(-\alpha t^d)] - a\alpha d \sum_{i=0}^5 \frac{b^i t^{d+i}}{d+i} \exp(-bt) + C[1 - \exp(-bt)],$ $a(t) = a[1 - \exp(-\alpha t^d)] + C, b(t) = b$ |

3.1 模型的性能比较标准

除了以前使用过的 AIC^[12]模型评价标准外,在这里还给出了其他模型比较评价标准.

(1) 误差平方和

$$SSE = \sum_{i=1}^n [m(t_i) - o_i]^2 \tag{27}$$

其中, n 表示故障数据集的样本大小. SSE 值越小,说明模型的性能越好.

(2) Bias 评价标准

Bias 表示实际观察到故障值和估计的故障值之差的绝对值之和的平均数,它的定义可以表示为

$$Bias = \frac{\sum_{i=1}^n |m(t_i) - o_{t_i}|}{n} \quad (28)$$

其中, $m(t_i)$ 表示到 t_i 时刻为止,期望估计检测出故障的数量; o_{t_i} 表示实际观察到故障发生的数量; n 为故障数据集样本大小.Bias 值越小,说明模型的性能越好.

(3) Variance 评价标准

Variance 表示估计偏差的标准差^[21],它可以表示为

$$Variance = \sqrt{\frac{1}{n-1} \sum_{i=1}^n [o_{t_i} - m(t_i) - Bias]^2} \quad (29)$$

Variance 值越小,说明模型的性能越好.

(4) RMSPE(root mean square prediction error)评价标准

RMSPE 表示模型预测出故障发生数量偏差性^[21],它可以定义为

$$RMSPE = \sqrt{Variance^2 + Bias^2} \quad (30)$$

RMSPE 值越小,说明模型的性能越好.

3.2 故障数据集

第 1 组故障数据集来自文献[22],总共用时 22 天,86 个软件故障被检测出来.第 2 组故障数据集来自文献[23],它是一个支持航空飞行器(space shuttle flights)的软件系统,测试用时 38 周,231 个软件故障被检测出来;另外,这个故障数据集包括重要故障(critical errors)、主要故障(major errors)和次要故障(minor errors).第 3 组故障数据集和第 4 组故障数据集是 Musa^[24]收集和整理的,这两个故障数据集是实际测试(real-world)中得到的,它们的系统代码分别为 4 和 27,被研究者广泛用于新建的软件可靠性模型性能评测中.第 3 组故障数据集包括测试用时 72 天,54 个故障被检测出来.第 4 组故障数据集包括测试用时 79 天,41 个故障被检测出来.

为了充分地验证和评估提出模型的拟合和预测能力,我们把故障数据集 1 和故障数据集 2 划分为 63%和 80%的子故障数据集.也就是说,分别用 63%和 80%的故障数据集来拟合和估计模型的参数,剩余 37%和 20%的故障数据集用来预测和评估模型的性能.另外,我们把故障数据集 3 和故障数据集 4 划分为 60%和 80%的子故障数据集.即,分别用 60%和 80%的故障数据集来拟合和估计模型的参数,剩余 40%和 20%的故障数据集用来预测和评估模型的性能.

3.3 模型的参数估计方法

本文用的参数估计方法是用最大似然估计方法,它可以表示为

$$L = \Pr\{N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n\} = \prod_{i=1}^n \frac{[m(t_i) - m(t_{i-1})]^{y_i - y_{i-1}}}{(y_i - y_{i-1})!} \exp[-(m(t_i) - m(t_{i-1}))].$$

为了方便计算,两边取对数为

$$l = \log L.$$

通过对 l 取导数,联立解方程组,则提出模型的参数(a, b, d, α 和 C)可以被估计出来(a^*, b^*, d^*, α^* 和 C^*):

$$\frac{\partial l}{\partial a} = \frac{\partial l}{\partial b} = \frac{\partial l}{\partial d} = \frac{\partial l}{\partial \alpha} = \frac{\partial l}{\partial C} = 0.$$

3.4 模型的性能比较

1) 数据集 1(DS1)

从表 2 可以看到,在用 63%的故障数据集时,与其他模型相比,我们能看到:

- 有关 $SSE_{predict}$ 的值方面,提出模型的 $SSE_{predict}$ 的值最小,是 124.0;其次是 DSS 模型,为 241.6;最差是 P-Z 模型,为 851.6;

- 有关 $Bias_{predict}$ 的值方面,提出模型的 $Bias_{predict}$ 值最小,仅为 3.5;其次为 DSS 模型,为 5.5;最差为 ISS 模型和 P-Z 模型,为 10.1;
- 有关 $Variance_{predict}$ 的值方面,提出模型的 $Variance_{predict}$ 的值最小,为 5.1;其次为 G-O 模型、Yamada 不完美调试模型-1、Yamada 不完美调试模型-2、Zhang-Teng Pham 模型、Kapur 模型-1 和 Kapur 模型-2,都是 6.5;最差为 ISS 模型和 P-Z 模型,都为 21.7;
- 有关 $RMSPE_{predict}$ 的值方面,提出模型的 $RMSPE_{predict}$ 的值最小,仅为 6.2;其次为 G-O 模型、Yamada 不完美调试模型-1、Yamada 不完美调试模型-2、Zhang-Teng Pham 模型、Kapur 模型-1 和 kapur 模型-2,都为 8.9;最差是 ISS 模型和 P-Z 模型,都为 23.9;
- 有关 AIC 值方面,提出模型的 AIC 值也是最小,仅为 76.1;其次为 DSS 模型,是 76.2;最差是 Zhang-Teng Pham 模型,为 93.1.虽然 DSS 模型拟合得很好,它的 AIC 值和提出模型的 AIC 值很接近,但是 DSS 模型的 $SSE_{predict}$ 、 $Bias_{predict}$ 、 $Variance_{predict}$ 和 $RMSPE_{predict}$ 的值却远大于提出模型相应的值.因此,提出的模型在同其他的软件可靠性模型相比,有更好的故障拟合效果和预测能力.

Table 2 Comparison results of different SRGMs for the first data set (63% of DS1)

表 2 第 1 组数据集上不同软件可靠增长模型比较结果(63%故障数据)

| 模型 | 估计的参数值 | $SSE_{predict}$ | $Bias_{predict}$ | $Variance_{predict}$ | $RMSPE_{predict}$ | AIC |
|--------------------|--|-----------------|------------------|----------------------|-------------------|------|
| G-O 模型 | $a=279.2228, b=0.0199$ | 431.4 | 6.1 | 6.5 | 8.9 | 85.1 |
| DSS 模型 | $a=84.1429, b=0.2178$ | 241.6 | 5.5 | 11.7 | 12.9 | 76.2 |
| ISS 模型 | $a=72.6964, b=0.3739, \beta=11.8833$ | 849.5 | 10.1 | 21.7 | 23.9 | 76.6 |
| Yamada 不完美调试模型-1 | $a=279.2228, b=0.0199, \alpha=0$ | 429.2 | 6.1 | 6.5 | 8.9 | 87.1 |
| Yamada 不完美调试模型-2 | $a=279.2228, b=0.0199, \alpha=0$ | 429.2 | 6.1 | 6.5 | 8.9 | 87.1 |
| P-N-Z 模型 | $a=71.2595, b=0.3706, \beta=11.2637, \alpha=0.0018$ | 758.4 | 9.6 | 20.5 | 22.6 | 78.8 |
| P-Z 模型 | $a=0, b=0.3739, \beta=11.8833, \alpha=0, C=72.6964$ | 851.6 | 10.1 | 21.7 | 23.9 | 80.6 |
| Zhang-Teng Pham 模型 | $a=86.9153, b=0.0198, \beta=0.0097, \alpha=0.0777, c=0.0647, p=0.3915$ | 432.8 | 6.1 | 6.5 | 8.9 | 93.1 |
| Pham Zhang IFD 模型 | $a=82.1168, b=0.2423, \alpha=0.0117$ | 304.2 | 6.1 | 13.1 | 14.5 | 77.9 |
| Kapur 模型-1 | $a=241.8748, b=0.1823, \alpha=0.1338, p=0.1262$ | 431.4 | 6.1 | 6.5 | 8.9 | 89.1 |
| Kapur 模型-2 | $a=278.4365, b=0.4822, \beta=0, \alpha=0.0028, p=0.0415$ | 437.5 | 6.1 | 6.5 | 8.9 | 91.1 |
| 提出的模型 | $a=2.0628, b=0.0135, \alpha=0.0227, d=1.994, C=431.056$ | 124.0 | 3.5 | 5.1 | 6.2 | 76.1 |

图 3(a)给出了提出模型使用故障数据集 1 的 63%进行故障拟合和预测的 95%的置信区间,我们也能看出:提出的模型很好地拟合故障数据,并准确地预测了在实际的软件测试当中,软件故障发生的数量.另外,在图 3(a)中,实际观察到的故障数量较好地落在提出模型的 95%上下界中.同时我们也能看到,提出模型有很好的拟合效果和准确的软件故障预测行为.

图 3(b)给出了提出模型使用故障数据集 1 的 80%进行故障拟合和预测的 95%的置信区间,我们也能看出:提出的模型很好地拟合故障数据,并精准地预测在实际的软件测试当中,软件故障发生的数量.另外,在图 3(b)中,实际观察到的故障数量很好地落在提出模型的 95%上下界中.同时我们也能看到,提出模型有很好的拟合效果和精确地预测软件故障发生.

从图 3(a)和图 3(b)可以看出:提出的模型不论是用 63%的故障数据,还是用 80%的故障数据时,都能很好地拟合软件故障数据和准确地预测软件故障发生.因此,可以合理地得出,提出的模型有很好的拟合能力和软件故障预测能力.

图 3(c)给出了提出模型使用故障数据集 2 的 63%进行故障拟合和预测的 95%的置信区间,我们能看到:提出的模型很好地拟合故障数据,并较好地预测软件发生故障的行为.另外,在图 3(c)中,实际观察到的故障数量很好地落在提出模型的 95%上下界中.同时我们也能看到,提出模型有很好的拟合效果和准确地预测软件故障发

生行为.

图 3(d)给出提出模型使用故障数据集 2 的 80%进行故障拟合和预测的 95%的置信区间,我们也能看到:提出的模型很好地拟合故障数据,并精确地预测软件故障发生的数量.另外,在图 3(d)中,实际观察到的故障数量很好地落在提出模型的 95%上下界中.同时我们也能看到,提出模型有很好的拟合效果和精确地预测软件故障发生数量.

从图 3(c)和图 3(d)可以看出:提出的模型不论是用 63%的故障数据,还是用 80%的故障数据时,提出的模型很好地拟合故障数据,并且准确地预测软件发生故障的行为.因此,在软件故障拟合和软件故障预测方面,提出的模型都有很好性能.

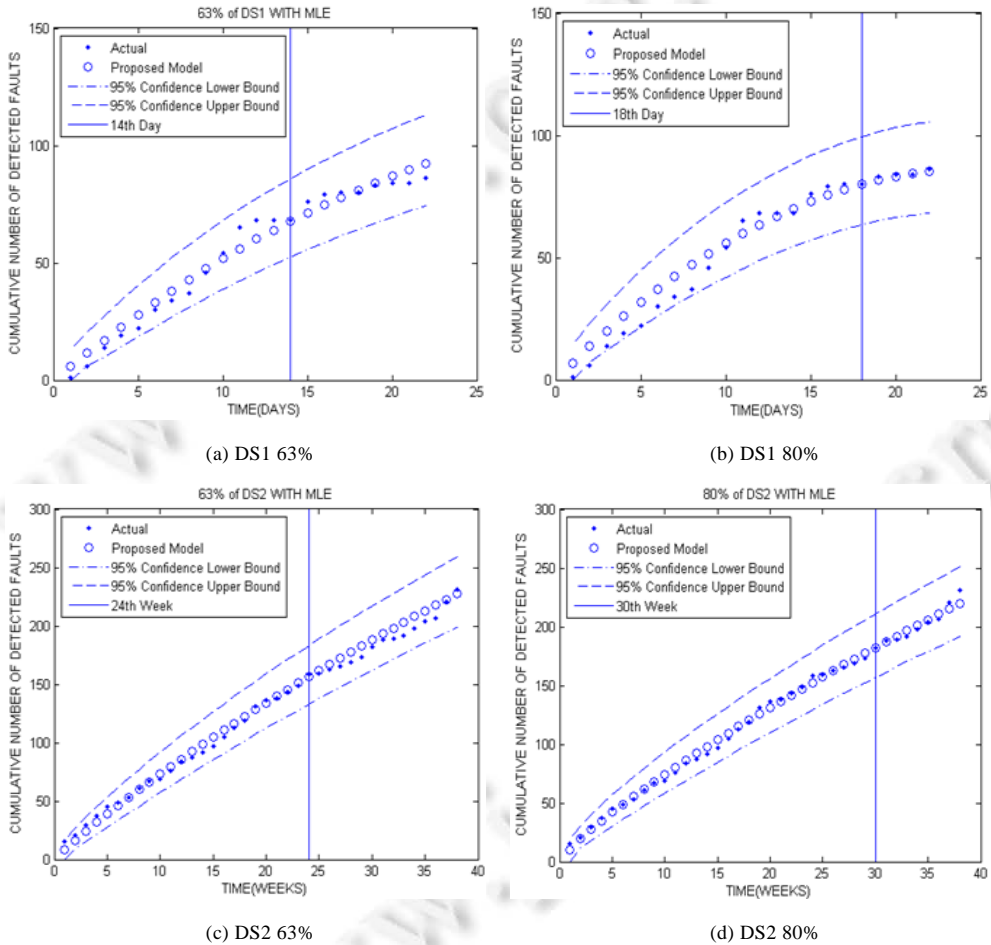


Fig.3 Plots of mean value functions and 95% s-confidence bounds for the actual data

图 3 用实际故障数据得出提出模型均值函数和 95% 置信区间

从表 3 可以看到,在用 80%的故障数据集时,与其他模型相比,我们能看到:

- 有关 $SSE_{predict}$ 的值方面,提出模型的 $SSE_{predict}$ 的值最小,是 2.9;其次是 DSS 模型,为 3.6;最差是 G-O 模型、Yamada 不完美调试模型-1 和 Kapur 模型-1,为 49.5;
- 有关 $Bias_{predict}$ 的值方面,提出模型的 $Bias_{predict}$ 的值最小,仅为 0.8;其次为 DSS 模型和 Pham Zhang IFD 模型,为 0.9;最差为 G-O 模型、Yamada 不完美调试模型-1 和 Kapur 模型-1 模型,为 2.9;
- 有关 $Variance_{predict}$ 的值方面,提出模型的 $Variance_{predict}$ 的值最小,为 1.7;其次为 DSS 模型,是 2.0;最差为

ISS 模型、P-N-Z 模型和 P-Z 模型,都为 5.4;

- 有关 $RMSPE_{predict}$ 的值方面,提出模型的 $RMSPE_{predict}$ 的值最小,仅为 1.9;其次为 DSS 模型,为 2.1;最差是 ISS 模型和 P-Z 模型,都为 5.9;
- 有关 AIC 值方面,提出模型的 AIC 值也是最小,仅为 95.7;其次为 DSS 模型,是 96.2;最差是 Zhang-Teng Pham 模型,为 115.1.虽然 DSS 模型拟合得很好,它的 AIC 值和提出模型的 AIC 值很接近,但是 DSS 模型的 $SSE_{predict}$, $Bias_{predict}$, $Variance_{predict}$ 和 $RMSPE_{predict}$ 的值却大于提出模型相应的值.因此,提出的模型在同其他的软件可靠性模型相比,有更好的故障拟合能力和预测软件发生故障的能力.

Table 3 Comparison results of different SRGMs for the first data set (80% of DS1)

表 3 第 1 组数据集上不同软件可靠增长模型比较结果(80%故障数据)

| 模型 | 估计的参数值 | $SSE_{predict}$ | $Bias_{predict}$ | $Variance_{predict}$ | $RMSPE_{predict}$ | AIC |
|--------------------|--|-----------------|------------------|----------------------|-------------------|-------|
| G-O 模型 | $a=153.875, b=0.0408$ | 49.5 | 2.9 | 2.4 | 3.7 | 107.1 |
| DSS 模型 | $a=90.9335, b=0.2031$ | 3.6 | 0.9 | 2.0 | 2.1 | 96.2 |
| ISS 模型 | $a=83.9267, b=0.2919, \beta=8.3426$ | 23.1 | 2.3 | 5.4 | 5.9 | 98 |
| Yamada 不完美调试模型-1 | $a=153.8759, b=0.0408, \alpha=0$ | 49.5 | 2.9 | 2.4 | 3.7 | 109.1 |
| Yamada 不完美调试模型-2 | $a=174.6693, b=0.0331, \alpha=0.0002$ | 37.7 | 2.6 | 2.8 | 3.8 | 109.3 |
| P-N-Z 模型 | $a=83.9879, b=0.2918, \beta=8.3473, \alpha=0$ | 22.6 | 2.3 | 5.4 | 5.8 | 99.9 |
| P-Z 模型 | $a=0, b=0.2919, \beta=8.3427, \alpha=0, C=83.9267$ | 23.1 | 2.3 | 5.4 | 5.9 | 102 |
| Zhang-Teng Pham 模型 | $a=12.2235, b=0.0448, \beta=0.0133, \alpha=0.0012, c=0.5169, p=0.0806$ | 48.3 | 2.8 | 2.4 | 3.7 | 115.1 |
| Pham Zhang IFD 模型 | $a=90.0669, b=0.2169, \alpha=0.0072$ | 4.2 | 0.9 | 2.1 | 2.3 | 98.1 |
| Kapur 模型-1 | $a=152.8639, b=0.8999, \alpha=0.0066, p=0.0456$ | 49.5 | 2.9 | 2.4 | 3.7 | 111.1 |
| Kapur 模型-2 | $a=152.7261, b=0.31, \beta=0.0002, \alpha=0.0037, p=0.1326$ | 47.3 | 2.8 | 2.4 | 3.7 | 113.1 |
| 提出的模型 | $a=0.2972, b=0.0273, \alpha=0.2286, d=2.0058, C=259.0116$ | 2.9 | 0.8 | 1.7 | 1.9 | 95.7 |

2) 数据集 2(DS2)

从表 4 可以看到,在用 63%的故障数据集时,与其他软件可靠性模型相比,我们能看到:

- 有关 $SSE_{predict}$ 的值方面,提出模型的 $SSE_{predict}$ 的值最小,是 848.6;其次是 Kapur 模型-1,为 1328.5;最差是 Yamada 不完美调试模型-1,为 38647.3;
- 有关 $Bias_{predict}$ 的值方面,提出模型的 $Bias_{predict}$ 的值最小,仅为 7.2;其次为 G-O 模型、ISS 模型、P-Z 模型、Zhang-Teng Pham 模型、kapur 模型-1 和 kapur 模型-2,都为 9.0;最差为 Yamada 不完美调试模型-1,为 45.7;
- 有关 $Variance_{predict}$ 的值方面,提出模型的 $Variance_{predict}$ 的值最小,为 4.1;Zhang-Teng Pham 模型也为 4.1;其次为 G-O 模型、ISS 模型、Yamada 不完美调试模型-2、P-Z 模型、Kapur 模型-1 和 Kapur 模型-2,都是 4.3;最差为 DSS 模型和 Pham Zhang IFD 模型,都为 31.5;
- 有关 $RMSPE_{predict}$ 的值方面,提出模型的 $RMSPE_{predict}$ 的值最小,仅为 8.4;其次为 Zhang-Teng Pham,为 9.9;最差是 Yamada 不完美调试模型-1,为 53.0;
- 有关 AIC 值方面,提出模型的 AIC 值也是最小,仅为 116.9;其次为 Yamada 不完美调试模型-1,是 117.1;最差是 Pham Zhang IFD 模型,为 175.1.虽然 Yamada 不完美调试模型-1 拟合得很好,它的 AIC 值和提出模型的 AIC 值很接近,但是 Yamada 不完美调试模型-1 的 $SSE_{predict}$, $Bias_{predict}$, $Variance_{predict}$ 和 $RMSPE_{predict}$ 的值却远远大于提出模型相应的值.因此,提出的模型与其他的软件可靠性模型相比,有更好的拟合故障数据的能力和更准确地预测软件中发生故障的数量的能力.

Table 4 Comparison results of different SRGMs for the second data set (63% of DS2)**表 4** 第 2 组数据集上不同软件可靠增长模型比较结果(63%故障数据)

| 模型 | 估计的参数值 | $SSE_{predict}$ | $Bias_{predict}$ | $Variance_{predict}$ | $RMSPE_{predict}$ | AIC |
|--------------------|--|-----------------|------------------|----------------------|-------------------|-------|
| G-O 模型 | $a=585.7512, b=0.0131$ | 1333.0 | 9.0 | 4.3 | 10.0 | 120.1 |
| DSS 模型 | $a=194.1617, b=0.1287$ | 5183.8 | 13.9 | 31.5 | 34.5 | 173.1 |
| ISS 模型 | $a=585.7512, b=0.0131, \beta=0$ | 1334.5 | 9.0 | 4.3 | 10.0 | 122.1 |
| Yamada 不完美调试模型-1 | $a=59.8799, b=0.2029, \alpha=0.0496$ | 38647.3 | 45.7 | 26.9 | 53.0 | 117.1 |
| Yamada 不完美调试模型-2 | $a=378.1685, b=0.01999, \alpha=0.0075$ | 1858.5 | 10.8 | 4.3 | 11.6 | 122.0 |
| P-N-Z 模型 | $a=23.9593, b=0.4902, \beta=0, \alpha=0.2519$ | 2776.3 | 13.1 | 5.3 | 14.2 | 123.6 |
| P-Z 模型 | $a=0.2009, b=0.0131, \beta=0, \alpha=0.0038, C=585.687$ | 1332.2 | 9.0 | 4.3 | 10.0 | 126.1 |
| Zhang-Teng Pham 模型 | $a=9.6139, b=0.0000249, \beta=0.2013, \alpha=0.0017, c=0.1817, p=0.0018$ | 1361.5 | 9.0 | 4.1 | 9.9 | 123.8 |
| Pham Zhang IFD 模型 | $a=194.1617, b=0.1287, \alpha=0$ | 5183.7 | 13.9 | 31.5 | 34.4 | 175.1 |
| Kapur 模型-1 | $a=536.6977, b=0.1136, \alpha=0.0837, p=0.1259$ | 1333.0 | 9.0 | 4.3 | 10.0 | 124.1 |
| Kapur 模型-2 | $a=584.4283, b=0.04315, \beta=0, \alpha=0.0023, p=0.3042$ | 1328.5 | 9.0 | 4.3 | 10.0 | 126.1 |
| 提出的模型 | $a=1.3149, b=0.00211, \alpha=0.3558, d=1.4428, C=4051.6536$ | 848.6 | 7.2 | 4.1 | 8.4 | 116.9 |

从表 5 可以看到,在用 80%的故障数据集时,与其他模型相比,我们可以看到:

- 有关 $SSE_{predict}$ 的值方面,提出模型的 $SSE_{predict}$ 的值最小,是 220.8;其次是 Kapur 模型-2,为 364.0;最差是 DSS 模型和 Pham Zhang IFD 模型,为 2091.2;
- 有关 $Bias_{predict}$ 的值方面,提出模型的 $Bias_{predict}$ 的值最小,仅为 4.3;同时,G-O 模型、ISS 模型、Yamada 不完美调试模型-1、Yamada 不完美调试模型-2、P-Z 模型、Zhang Teng Pham 模型、Kapur 模型-1 和 Kapur 模型-2 都是 4.3;其次为 P-N-Z 模型,为 6.9;最差为 DSS 模型和 Pham Zhang IFD 模型,为 12.2;
- 有关 $Variance_{predict}$ 的值方面,提出模型的 $Variance_{predict}$ 的值最小,为 7.3;其次为 P-N-Z 模型,是 7.5;最差为 DSS 模型和 Pham Zhang IFD 模型,都为 28.4;
- 有关 $RMSPE_{predict}$ 的值方面,提出模型的 $RMSPE_{predict}$ 的值最小,仅为 8.5;P-N-Z 模型也为 8.5;其次为 Kapur 模型-2,为 10.8;最差是 DSS 模型和 Pham Zhang IFD 模型,都为 30.9;
- 有关 AIC 值方面,提出模型的 AIC 值也是最小,仅为 145.3;其次为 Yamada 不完美调试模型-2,是 147;最差是 Pham Zhang IFD 模型,为 208.1.虽然 Yamada 不完美调试模型-2 拟合得很好,它的 AIC 值和提出模型的 AIC 值很接近,但是 Yamada 不完美调试模型-2 的 $SSE_{predict}$, $Bias_{predict}$, $Variance_{predict}$ 和 $RMSPE_{predict}$ 的值却大于提出模型相应的值.因此,提出的模型在与其他的软件可靠性模型相比,故障拟合能力和预测软件发生故障的能力都要好于其他软件可靠性模型.

3) 数据集 3(DS3)

从表 6 可以看到,在用 60%的故障数据集时,同其他软件可靠性模型相比,我们可以看到:

- 有关 $SSE_{predict}$ 的值方面,提出模型的 $SSE_{predict}$ 的值最小,是 351.37;其次是 P-N-Z 模型,为 419.85;最差是 Kapur 模型-1,为 65503;
- 有关 $Bias_{predict}$ 的值方面,提出模型的 $Bias_{predict}$ 的值最小,仅为 3.0;其次为 P-N-Z 模型,为 9.0;最差为 Kapur 模型-2,为 129.67;
- 有关 $Variance_{predict}$ 的值方面,提出模型的 $Variance_{predict}$ 的值最小,为 4.66;其次为 P-N-Z 模型,为 4.73;最差为 Kapur 模型-1,为 98.36;
- 有关 $RMSPE_{predict}$ 的值方面,提出模型的 $RMSPE_{predict}$ 的值最小,仅为 5.54;其次为 P-N-Z 模型,都为 5.72;最差是 Kapur 模型-2,都为 129.89;
- 有关 AIC 值方面,提出模型的 AIC 值也是最小,仅为 148.11;其次为 G-O 模型,是 149.41;最差是 Pham Zhang IFD 模型,为 160.5.

Table 5 Comparison results of different SRGMs for the second data set (80% of DS2)

表 5 第 2 组数据集上不同软件可靠增长模型比较结果(80%故障数据)

| 模型 | 估计的参数值 | $SSE_{predict}$ | $Bias_{predict}$ | $Variance_{predict}$ | $RMSPE_{predict}$ | AIC |
|-----------------------|--|-----------------|------------------|----------------------|-------------------|-------|
| G-O 模型 | $a=411.3659, b=0.01947$ | 365.7 | 4.3 | 10.0 | 10.9 | 149.6 |
| DSS 模型 | $a=211.9166, b=0.115$ | 2091.2 | 12.2 | 28.4 | 30.9 | 206.1 |
| ISS 模型 | $a=411.3659, b=0.0195, \beta=0$ | 365.7 | 4.3 | 10.0 | 10.9 | 151.6 |
| Yamada 不完美 调试模型-1 | $a=411.3659, b=0.0195, \alpha=0$ | 365.7 | 4.3 | 10.0 | 10.9 | 150.5 |
| Yamada 不完美 调试模型-2 | $a=411.3659, b=0.0195, \alpha=0$ | 365.7 | 4.3 | 10.0 | 10.9 | 147 |
| P-N-Z 模型 | $a=29.7626, b=0.4722, \beta=0, \alpha=0.1874$ | 488.9 | 6.9 | 7.5 | 8.5 | 149.5 |
| P-Z 模型 | $a=0.0872, b=0.01947, \beta=0, \alpha=0.0014, C=411.3575$ | 365.7 | 4.3 | 10.0 | 10.9 | 155.6 |
| Zhang-Teng Pham 模型 | $a=63.9842, b=0.000058, \beta=0.0889, \alpha=0.1514, c=0.1363, p=0.3069$ | 365.7 | 4.3 | 10.0 | 10.9 | 157.6 |
| Pham Zhang IFD 模型 | $a=211.9167, b=0.1150, \alpha=0$ | 2091.2 | 12.2 | 28.4 | 30.9 | 208.1 |
| Kapur 模型-1 | $a=408.3188, b=0.13298, \alpha=0.0074, p=0.1475$ | 365.3 | 4.3 | 10.0 | 10.9 | 153.6 |
| Kapur 模型-2 | $a=384.7579, b=0.1478, \beta=0, \alpha=0.0647, p=0.1409$ | 364.0 | 4.3 | 10.0 | 10.8 | 155.6 |
| 提出的模型 | $a=5.7, b=0.002125, \alpha=0.4168, d=1.19, C=5002.5649$ | 220.8 | 4.3 | 7.3 | 8.5 | 145.3 |

Table 6 Comparison results of different SRGMs for the third data set (60% of DS3)

表 6 第 3 组数据集上不同软件可靠增长模型比较结果(60%故障数据)

| 模型 | 估计的参数值 | $SSE_{predict}$ | $Bias_{predict}$ | $Variance_{predict}$ | $RMSPE_{predict}$ | AIC |
|-----------------------|--|-----------------|------------------|----------------------|-------------------|--------|
| G-O 模型 | $a=207.8778, b=0.0048$ | 502.47 | 3.45 | 4.89 | 5.99 | 149.41 |
| DSS 模型 | $a=49.1034, b=0.0688$ | 872.99 | 5.32 | 10.97 | 12.20 | 158.05 |
| ISS 模型 | $a=124.5116, b=0.0127, \beta=0.5930$ | 441.38 | 3.28 | 4.78 | 5.8 | 151.40 |
| Yamada 不完美 调试模型-1 | $a=207.8776, b=0.0048, \alpha=0.0555$ | 502.46 | 3.45 | 4.89 | 5.99 | 151.41 |
| Yamada 不完美 调试模型-2 | $a=207.8771, b=0.0048, \alpha=0$ | 502.45 | 3.45 | 4.89 | 5.99 | 151.41 |
| P-N-Z 模型 | $a=114.4073, b=0.0146, \beta=0.6856, \alpha=0$ | 419.85 | 3.21 | 4.73 | 5.72 | 153.40 |
| P-Z 模型 | $a=33.3468, b=0.0069, \beta=0.0016, \alpha=0.0099, C=146.4195$ | 493.8 | 3.43 | 4.87 | 5.95 | 155.41 |
| Zhang-Teng Pham 模型 | $a=38.5485, b=0.0093, \beta=0.0009, \alpha=0.1521, c=0.0261, p=0.3378$ | 504.81 | 3.45 | 4.88 | 5.98 | 157.41 |
| Pham Zhang IFD 模型 | $a=49.1034, b=0.0688, \alpha=0$ | 872.99 | 5.32 | 10.97 | 12.20 | 160.5 |
| Kapur 模型-1 | $a=207.5796, b=0.0630, \alpha=0.0768, p=0.0014$ | 65503 | 48.27 | 98.36 | 109.57 | 153.41 |
| Kapur 模型-2 | $a=207.5699, b=0.0918, \beta=0.0015, \alpha=0.0527, p=0.3453$ | 472310 | 129.67 | 7.39 | 129.89 | 156.41 |
| 提出的模型 | $a=6.6591, b=0.0092, \alpha=0.4198, d=0.7208, C=203.7921$ | 351.37 | 3.0 | 4.66 | 5.54 | 148.11 |

从表 7 可以看到,在用 80%的故障数据集时,同其他模型相比,我们能看到:

- 有关 $SSE_{predict}$ 的值方面,提出模型的 $SSE_{predict}$ 的值最小,是 6.7;其次是 Pham Zhang IFD 模型,为 17.44;最差是 Kapur 模型-1 为 3943.9;
- 有关 $Bias_{predict}$ 的值方面,提出模型的 $Bias_{predict}$ 的值最小,仅为 0.61;其次为 Pham Zhang IFD 模型,为 0.94;最差为 Kapur 模型-1,为 16.57;
- 有关 $Variance_{predict}$ 的值方面,提出模型和 Pham Zhang IFD 模型的 $Variance_{predict}$ 的值都是 0.84,为最小;其次为 DSS 模型,是 1.14;最差为 Kapur 模型-1,为 2.80;
- 有关 $RMSPE_{predict}$ 的值方面,提出模型的 $RMSPE_{predict}$ 的值最小,仅为 1.04;其次为 Pham Zhang IFD 模型,为 1.26;最差是 Kapur 模型-1 为 16.8;

- 有关 AIC 值方面,提出模型的 AIC 值也是最小,仅为 188.67;其次为 G-O 模型,是 190.01;最差是 DSS 模型,为 198.61.

Table 7 Comparison results of different SRGMs for the third data set (80% of DS3)

表 7 第 3 组数据集上不同软件可靠增长模型的比较结果(80%故障数据)

| 模型 | 估计的参数值 | $SSE_{predict}$ | $Bias_{predict}$ | $Variance_{predict}$ | $RMSPE_{predict}$ | AIC |
|--------------------|--|-----------------|------------------|----------------------|-------------------|--------|
| G-O 模型 | $a=143.9553, b=0.0075$ | 291.13 | 3.92 | 2.49 | 4.64 | 190.01 |
| DSS 模型 | $a=59.8918, b=0.057$ | 47.01 | 1.58 | 1.14 | 1.95 | 198.61 |
| ISS 模型 | $a=62.8386, b=0.0484, \beta=2.8007$ | 92.15 | 2.21 | 1.47 | 2.65 | 191.21 |
| Yamada 不完美调试模型-1 | $a=143.9553, b=0.0075, \alpha=0.000001$ | 291.13 | 3.92 | 2.49 | 4.64 | 192.01 |
| Yamada 不完美调试模型-2 | $a=143.9553, b=0.0075, \alpha=0.000001$ | 291.13 | 3.62 | 2.49 | 4.64 | 192.01 |
| P-N-Z 模型 | $a=62.8386, b=0.0484, \beta=2.8007, \alpha=0$ | 92.15 | 2.21 | 1.47 | 2.65 | 193.21 |
| P-Z 模型 | $a=14.3849, b=0.0211, \beta=0.6313, \alpha=0.0452, C=72.5585$ | 131.09 | 2.59 | 2.24 | 3.43 | 195.75 |
| Zhang-Teng Pham 模型 | $a=76.6018, b=0.0454, \beta=0.2255, \alpha=0.0024, c=0.0166, p=0.7184$ | 253.37 | 3.52 | 2.29 | 4.19 | 197.87 |
| Pham Zhang IFD 模型 | $a=55.9203, b=0.0798, \alpha=0.0141$ | 17.44 | 0.94 | 0.84 | 1.26 | 195.76 |
| Kapur 模型-1 | $a=127.4066, b=0.0984, \alpha=0.0860, p=0.1150$ | 3943.9 | 16.57 | 2.80 | 16.8 | 194.01 |
| Kapur 模型-2 | $a=132.0361, b=0.3282, \beta=0.0, \alpha=0.0828, p=0.0249$ | 296.66 | 3.96 | 2.49 | 4.68 | 196.01 |
| 提出的模型 | $a=6.8740, b=0.0140, \alpha=0.7129, d=0.7236, C=206.1978$ | 6.7 | 0.61 | 0.84 | 1.04 | 188.67 |

图 4(a)和图 4(b)给出提出模型使用故障数据集 3 的 60%和 80%进行故障拟合和预测的 95%的置信区间, 我们能看到:估计的故障数量很好地落在 95%的置信区间内,而且提出的模型很好地拟合故障数据,并很好地预测软件发生故障的行为.

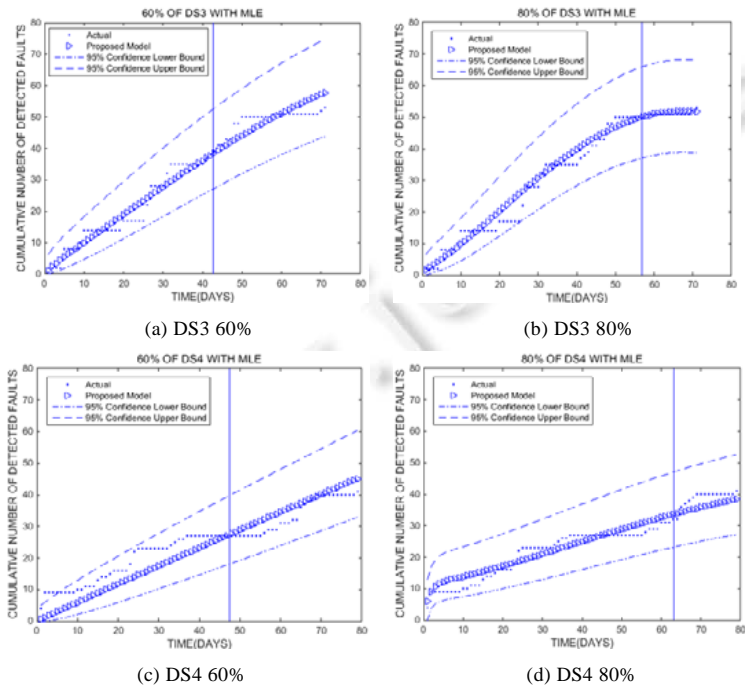


Fig.4 Plots of mean value functions and 95% s-confidence bounds for the actual data

图 4 用实际故障数据得出提出模型均值函数和 95%置信区间

4) 数据集 4(DS4)

从表 8 可以看到,在用 60%的故障数据集时,同其他软件可靠性模型相比,我们能看到:

- 有关 $SSE_{predict}$ 的值方面,提出模型的 $SSE_{predict}$ 的值最小,是 322.1;其次是 Yamada 不完美调试模型-2,为 1181.5;最差是 Kapur 模型-1,为 2878.6;
- 有关 $Bias_{predict}$ 的值方面,提出模型的 $Bias_{predict}$ 的值最小,仅为 2.74;其次为 Yamada 不完美调试模型-2,为 4.63;最差为 Kapur 模型-1,为 29.57;
- 有关 $Variance_{predict}$ 的值方面,提出模型的 $Variance_{predict}$ 的值最小,为 1.67;其次是 Yamada 不完美调试模型-2,为 10.02;最差为 Kapur 模型-1,为 60.29;
- 有关 $RMSPE_{predict}$ 的值方面,提出模型的 $RMSPE_{predict}$ 的值最小,仅为 3.21;其次为 Yamada 不完美调试模型-2,是 11.04;最差是 Kapur 模型-1 为 67.15;
- 有关 AIC 值方面,提出模型的 AIC 值也是最小,仅为 101.76;其次为 G-O 模型,是 102.58;最差是 Pham Zhang IFD 模型,为 131.24.提出的模型在与其他软件可靠性模型相比,有更好的故障拟合和预测性能.

Table 8 Comparison results of different SRGMs for the fourth data set (60% of DS4)

表 8 第 4 组数据集上不同软件可靠增长模型比较结果(60%故障数据)

| 模型 | 估计的参数值 | $SSE_{predict}$ | $Bias_{predict}$ | $Variance_{predict}$ | $RMSPE_{predict}$ | AIC |
|--------------------|--|-----------------|------------------|----------------------|-------------------|--------|
| G-O 模型 | $a=30.0517, b=0.0487$ | 1648.3 | 5.47 | 11.94 | 13.14 | 102.58 |
| DSS 模型 | $a=27.5453, b=0.1244$ | 2235.4 | 6.42 | 14.08 | 15.48 | 123.7 |
| ISS 模型 | $a=30.0517, b=0.0487, \beta=0.000001$ | 1648.3 | 5.47 | 11.94 | 13.14 | 104.58 |
| Yamada 不完美调试模型-1 | $a=30.05, b=0.0487, \alpha=0$ | 1648.3 | 5.47 | 11.94 | 13.14 | 104.58 |
| Yamada 不完美调试模型-2 | $a=25.7509, b=0.0571, \alpha=0.0038$ | 1181.5 | 4.63 | 10.02 | 11.04 | 104.86 |
| P-N-Z 模型 | $a=30.0510, b=0.0487, \beta=0.0002, \alpha=0$ | 1648.4 | 5.47 | 11.94 | 13.14 | 106.58 |
| P-Z 模型 | $a=3.512, b=0.0487, \beta=0, \alpha=0.000001, C=30.0517$ | 1648.3 | 5.47 | 11.94 | 13.14 | 108.58 |
| Zhang-Teng Pham 模型 | $a=27.8458, b=0.0011, \beta=0.0009, \alpha=0.000001, c=0.0528, p=0.9266$ | 1647.7 | 5.47 | 11.94 | 13.13 | 110.58 |
| Pham Zhang IFD 模型 | $a=27.3715, b=0.1500, \alpha=0.0201$ | 2296.3 | 6.51 | 14.29 | 15.7 | 131.24 |
| Kapur 模型-1 | $a=29.8428, b=0.3338, \alpha=0.1468, p=0.0070$ | 28786 | 29.57 | 60.29 | 67.15 | 106.58 |
| Kapur 模型-2 | $a=19.5822, b=0.0892, \beta=0.000001, \alpha=0.3484, p=0.8376$ | 1648.1 | 5.47 | 11.94 | 13.14 | 108.58 |
| 提出的模型 | $a=93.9382, b=0.000644, \alpha=0.0102, d=0.0027, C=908.7152$ | 322.1 | 2.74 | 1.67 | 3.21 | 101.76 |

从表 9 可以看到,在用 80%的故障数据集时,与其他模型相比,我们能够看到:

- 有关 $SSE_{predict}$ 的值方面,提出模型的 $SSE_{predict}$ 的值最小,是 59.42;其次是 Yamada 不完美调试模型-1,为 335.56;最差是 Yamada 不完美调试模型-2,为 16269;
- 有关 $Bias_{predict}$ 的值方面,提出模型的 $Bias_{predict}$ 的值最小,仅为 1.71;其次为 Yamada 不完美调试模型-1,为 4.37;最差为 Yamada 不完美调试模型-2,为 31.08;
- 有关 $Variance_{predict}$ 的值方面,提出模型的 $Variance_{predict}$ 的值最小,为 3.45;其次为 Yamada 不完美调试模型-2,是 7.37;最差为 Pham Zhang IFD 模型,为 13.72;
- 有关 $RMSPE_{predict}$ 的值方面,提出模型的 $RMSPE_{predict}$ 的值最小,仅为 3.85;其次为 Yamada 不完美调试模型-1,为 10.11;最差是 Yamada 不完美调试模型-2,都为 31.94;
- 有关 AIC 值方面,提出模型的 AIC 值也是最小,仅为 124.98;其次为 Yamada 不完美调试模型-2,是 128.07;最差是 Pham Zhang IFD 模型,为 166.81.提出的模型在同其他的软件可靠性模型相比,故障拟合能力和预测软件发生故障的能力都要好于其他软件可靠性模型.

Table 9 Comparison results of different SRGMs for the fourth data set (80% of DS4)**表 9** 第 4 组数据集上不同软件可靠增长模型的比较结果(80%故障数据)

| 模型 | 估计的参数值 | $SSE_{predict}$ | $Bias_{predict}$ | $Variance_{predict}$ | $RMSP\bar{E}_{predict}$ | AIC |
|--------------------|--|-----------------|------------------|----------------------|-------------------------|--------|
| G-O 模型 | $a=37.555, b=0.0303$ | 572.25 | 5.74 | 11.97 | 13.27 | 128.87 |
| DSS 模型 | $a=33.0402, b=0.0841$ | 741.08 | 6.5 | 13.59 | 15.06 | 157.5 |
| ISS 模型 | $a=37.5549, b=0.0303, \beta=0.000001$ | 572.26 | 5.74 | 11.97 | 13.27 | 130.87 |
| Yamada 不完美调试模型-1 | $a=20.1186, b=0.0801, \alpha=0.0091$ | 335.56 | 4.37 | 9.12 | 10.11 | 128.54 |
| Yamada 不完美调试模型-2 | $a=11.9648, b=0.1735, \alpha=0.0266$ | 16269 | 31.08 | 7.37 | 31.94 | 128.07 |
| P-N-Z 模型 | $a=21.0987, b=0.084, \beta=0.4521, \alpha=0.0103$ | 435.09 | 5.01 | 10.44 | 11.58 | 131.93 |
| P-Z 模型 | $a=1.01, b=0.0303, \beta=0.000001, \alpha=0.000001, C=37.555$ | 572.25 | 5.74 | 11.97 | 13.27 | 134.87 |
| Zhang-Teng Pham 模型 | $a=18.0465, b=0.000011, \beta=0.0002, \alpha=0.0105, c=0.0631, p=0.4910$ | 572.99 | 5.74 | 11.98 | 13.28 | 136.87 |
| Pham Zhang IFD 模型 | $a=32.8443, b=0.0958, \alpha=0.0083$ | 755.7 | 6.57 | 13.72 | 15.21 | 166.81 |
| Kapur 模型-1 | $a=30.3567, b=0.5245, \alpha=0.0716, p=0.1917$ | 707.8 | 6.36 | 13.25 | 14.7 | 132.87 |
| Kapur 模型-2 | $a=22.7409, b=0.0644, \beta=0.0014, \alpha=0.3944, p=0.7785$ | 572.29 | 5.74 | 11.97 | 13.27 | 134.88 |
| 提出的模型 | $a=12.4677, b=0.0030, \alpha=0.6746, d=0.9811, C=2730.4$ | 59.42 | 1.71 | 3.45 | 3.85 | 124.98 |

从图 4(c)和图 4(d)可以看出:提出的模型不论是用 60%的故障数据,还是用 80%的故障数据时,估计的故障数据都很好地位于 95%的置信区间内;另外,提出的模型也很好拟合故障数据,并且准确地预测软件发生故障的行为.因此,无论是在软件故障拟合还是在软件故障预测方面,提出的模型都有很好的性能.

3.5 提出模型优于其他模型机理的分析

从以上的实验结果可以看到,提出的模型性能明显优于其他模型,包括完美调试模型和不完美调试模型.提出的模型优于其他模型的机理分析如下.

① 考虑故障引进的情况,是提出的模型性能优于完美调试模型性能重要原因之一.

例如,完美调试模型包括 G-O 模型、DSS 模型和 ISS 模型.由于软件测试和调试的复杂性,例如有客观因素——测试环境的多样性和软件编写的复杂性,有主观因素——测试者或调试者的技术以及心理因素等,故障引入很大程度上是由于软件复杂性造成的(客观因素),以及调试者对软件复杂性没有深刻认识(主观因素),另外也与调试者的技术也有一定关系(主观因素).当软件发布日期临近时,调试者的心理也会产生巨大压力,调试者的心理也会发生巨大变化(主观因素),在这种压力下,当检测出的故障被去除时,很可能引进新的故障.考虑故障被完全去除,而没有引进故障的假设是完全不符合实际的故障被去除的情况.因此,用完美去除故障的假设来建立软件可靠性模型在与我们提出的不完美调试模型进行性能比较时,考虑故障引进的情况,是提出模型的性能优于完美调试模型性能重要原因之一.

② 考虑故障引进多种变化的情况,是提出模型性能优于其他不完美调试模型性能的重要原因.

例如,其他不完美调试模型包括 Yamada 不完美调试模型-1、Yamada 不完美调试模型-2、P-N-Z 模型、P-Z 模型、Zhang-Teng Pham 模型、Pham Zhang IFD 模型、Kapur 模型-1 和 Kapur 模型-2,由于故障引进的复杂性,故障引进率可能表现为随着测试时间逐渐下降,或者故障引进率随测试时间表现为先增后减的变化,或者故障引进随测试时间无规律变化等.其他不完美软件调试模型只考虑故障引进率的一种变化,不是为常数就是随测试时间下降,完全没有考虑到故障引进的其他变化.而我们提出的模型可以考虑故障引进率为多种变化,既可以为随测试时间下降,还可以随测试时间有先增后减的变化.因此,提出的模型更能适应复杂的测试和调试的情况,有更强的鲁棒性和稳定性,不易受到故障数据集噪音的干扰.而其他的不完美调试模型则受限于故障引进率的单一变化,在某些情况下会表现的很不稳定,适应性和鲁棒性也会很差.因此,考虑故障引进多种变化的情况,是提出模型性能优于其他不完美调试模型性能的重要原因.

③ 考虑故障引率非线性变化,是提出模型性能优于其他不完美调试模型性能的另一个重要原因。

为建模方便,Zhang-Teng Pham 模型、Pham Zhang IFD 模型、Kapur 模型-1 和 Kapur 模型-2 假设故障引率为常量;Yamada 不完美调试模型-2 和 P-N-Z 模型也假设故障引率为常量,并且故障内容函数(故障总个数)随测试时间线性变化.由于故障引进的复杂性,故障引进率极有可能表现为非线性变化,而不是线性变化或者常量.而我们提出的模型正是假设故障引进服从 Weibull 分布,故障引进率随测试时间表现为非线性变化.虽然 Yamada 不完美调试模型-1 和 P-Z 模型的故障引进率都随测试时间表现为非线性变化,但是由于 Yamada 不完美调试模型-1 是假设故障引进率随测试时间逐渐增长,这种情况与实际调试情况不符.因为在软件测试过程中,引进的故障会越来越少,所以故障引进率不会随测试时间逐渐增长.另外,P-Z 模型假设故障引进服从指数分布,但是由于我们提出模型假设故障引进服从 Weibull 分布包含指数分布的特殊情况,所以我们提出的模型能更好地适应软件测试和调试情况,提出模型的性能也会好于 Yamada 不完美调试模型-1 和 P-Z 模型的性能.

综上所述,提出的模型优于其他模型的原因有:一是提出的模型考虑故障引进的情况,更符合实际的软件测试和调试过程;二是假设故障引进服从 Weibull 分布,能够包括更多在软件测试和调试过程中故障引进率变化情况;三是假设故障引进率非线性变化更符合故障引进随测试复杂变化的情况.

3.6 提出模型的参数敏感性分析

一般来说,模型的参数比较多时,需要进行模型的参数敏感性分析^[25].主要目的是考察哪些参数对模型的评估效果和预测有重要影响,也就是考察一下模型的鲁棒性(robust).而具体使用的方法就是改变模型的某个参数值,模型其他参数则保持不变.图 5(a)~图 5(e)显示出提出模型的参数 a, b, α, d 和 C 的敏感性分析的情况.

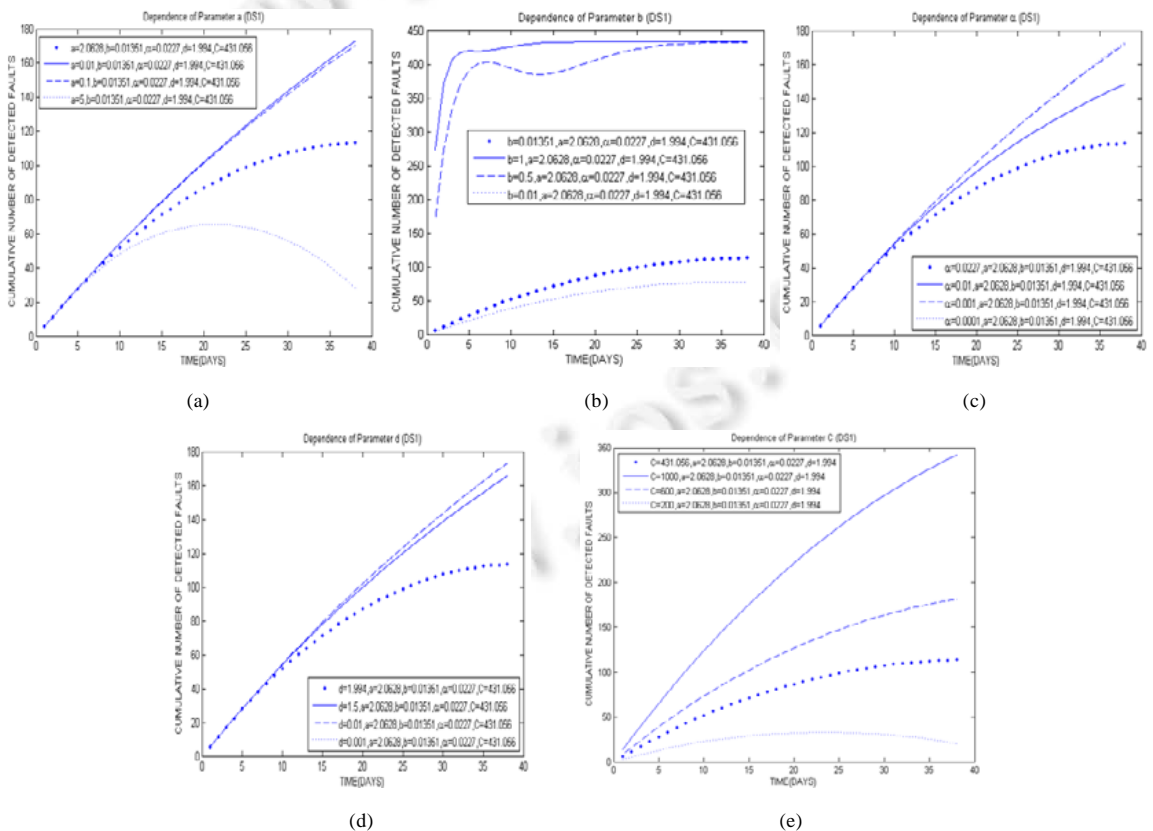


Fig.5 Denote sensitivity analysis of parameters of the proposed model for 63% of DS1, respectively

图 5 提出的模型分别用故障数据集 DS1 的 63% 的故障数据进行参数敏感分析的情况

从图 5(a)、图 5(b)和图 5(e)中可以清晰地看到,提出模型期望估计的累计检测出的故障数量随着期望最终引进故障的数量 a 、故障检测率 b 和期望最初存在软件中的故障数量 C 的变化而大幅度地发生变化.因此,提出模型的参数 a, b 和 C 都是有影响的参数.

4 结束语

本文介绍 Weibull 分布的相关概念和表示方法,然后分析在软件调试过程中,故障引进服从 Weibull 分布的原因.在此基础上,用 Weibull 分布进行故障引进过程的相应模拟,并提出一个基于 Weibull 分布引进故障的软件可靠性增长模型,同时给出相应的模型推导过程.为了验证提出模型的拟合故障数据和预测软件中故障发生数量的能力,本文给出了相应的故障数据集和有关的模型性能比较标准,并对提出的模型用最大似然估计法进行相应的模型参数估计.实验结果表明:提出的模型在与多种软件可靠性增长模型(包括完美调试软件可靠性增长模型和不完美调试软件可靠性增长模型)比较后,有更好的故障拟合效果和更准确地预测软件故障发生数量的能力.为了考察提出模型的参数变化对软件可靠性评估和软件故障预测的影响程度,我们也给出相应的参数敏感性分析.参数敏感性分析实验的结果表明,提出模型的参数 a, b 和 C 都是有影响的参数.

考虑到实际测试过程中检测到的故障不是被立即和完美地去掉,而是在检测和去除故障之间会存在时间延迟,在去除故障时可能引进新的故障,因此,未来的研究将在本文研究的基础上,就故障排错延迟问题进行深入研究.

致谢 在此,我们感谢审稿专家对本文提出的宝贵建议.

References:

- [1] Goel AL. Software reliability models: Assumptions, limitations and applicability. *IEEE Trans. on Software Engineering*, 1985, SE-11:1411–1423.
- [2] Ohba M, Chou XM. Does imperfect debugging effect software reliability growth. In: *Proc. of the 11th Int'l Conf. of Software Engineering*. 1989. 237–244.
- [3] Kapur PK, Garg RB. Optimal software release policies for software reliability growth model under imperfect debugging. *RAIRO*, 1990,24:295–305.
- [4] Pham H, Nordmann L, Zhang X. A general Imperfect software debugging model with S-shaped fault detection rate. *IEEE Trans. on Reliability*, 1999,R-48(2):169–175.
- [5] Pham H. A software cost model with imperfect debugging, random life cycle and penalty cost. *Int'l Journal of Systems Science*, 1996,27(5):455–463.
- [6] Zhang X, Teng X, Pham H. Considering fault removal efficiency in software reliability assessment. *IEEE Trans. on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 2003,33(1):2241–2252.
- [7] Kapur PK, Gupta D, Gupta A, Jha PC. Effect of introduction of faults and imperfect debugging on release time. *Ratio Mathematica*, 2008,18:62–90.
- [8] Kapur PK, Pham 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.
- [9] Wang JY, Wu ZB, Shu YJ, Zhang Z. A general imperfect software debugging model considering the nonlinear process of fault introduction. In: *Proc. of the 14th Int'l Conf. on Quality Software (QSIC)*. 2014. 222–227.
- [10] Wang JY, Wu ZB. Study of the nonlinear imperfect software debugging model. *Reliability Engineering & System Safety*, 2016,153:180–192.
- [11] Wang JY, Wu ZB, Shu YJ, Zhang Z. An imperfect software debugging model considering log-logistic distribution fault content function. *Journal of Systems and Software*, 2015,100:167–181.
- [12] Wang JY, Wu ZB, Shu YJ, Zhang Z. Software reliability model with irregular changes of fault detection rate. *Ruan Jian Xue Bao/Journal of Software*, 2015,26(10):2465–2484 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4746.htm> [doi: 10.13328/j.cnki.jos.004746]
- [13] Xie JY, An JX, Zhu JH. NHPP software reliability growth model considering imperfect debugging. *Ruan Jian Xue Bao/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]

- [14] Yin Q, Li J, Bom KH, Guo P. A new cascade software reliability model. In: Proc. of the 3rd Int'l Conf. on Natural Computation (ICNC 2007). IEEE Press, 2007. 241–245.
- [15] Yamada S, Tokuno K, Osaki S. Imperfect debugging models with fault introduction rate for software reliability assessment. Int'l Journal of System Science, 1992,23(12):2253–2264.
- [16] Pham H, Zhang X. An NHPP software reliability models and its comparison. Int'l Journal of Reliability, Quality and Safety Engineering, 1997,4(3):269–282.
- [17] Goel AL, Okumoto K. Time dependent error detection rate model for software reliability and other performance measures. IEEE Trans. on Reliability, 1979,R-28(3):206–211.
- [18] Yamada S, Ohba M, Osaki S. S-shaped reliability growth modeling for software error detection. IEEE Trans. on Reliability, 1983, 32:475–484.
- [19] Ohba M. Inflection S-shaped software reliability growth models. In: Osaki S, Hatoyama Y, ed. Proc. of the Stochastic Models in Reliability Theory. Berlin: Springer-Verlag, 1984. 144–162.
- [20] Pham H. System Software Reliability. London: Springer-Verlag, 2006.
- [21] Sharma K, Garg R, Nagpal CK, Garg RK. Selection of optimal software reliability growth models using a distance based approach. IEEE Trans. on Reliability, 2010,59(2):266–276.
- [22] Tohma Y, Jacoby R, Murata Y, Yamamoto M. Hyper-geometric distribution model to estimate the number of residual software faults. In: Proc. of the COMPSAC'89. 1989. 610–617.
- [23] Misra PN. Software reliability analysis. IBM Systems Journal, 1986,22(3):262–270.
- [24] Musa JD. Software Reliability Data, Cyber Security and Information Systems Information Analysis Center. 1980.
- [25] Li X, Xie M, Ng SH. Sensitivity analysis of release time of software reliability models incorporating testing effort with multiple change-points. Applied Mathematical Modelling, 2010,34:560–570.

附中文参考文献:

- [12] 王金勇,吴智博,舒燕君,张展.故障检测率不规则变化的软件可靠性模型.软件学报,2015,26(10):2465–2484. <http://www.jos.org.cn/1000-9825/4746.htm> [doi: 10.13328/j.cnki.jos.004746]
- [13] 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]



王金勇(1974—),男,黑龙江鸡西人,博士,讲师,CCF 专业会员,主要研究领域为软件可靠性,故障预测.



郭新峰(1974—),男,副教授,CCF 专业会员,主要研究领域为软件工程,机器学习.



张策(1978—),男,博士,讲师,CCF 专业会员,主要研究领域为软件测试,软件可靠性评估,容错计算,可信计算.



李济洪(1964—),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为机器学习,软件可靠性.



米晓萍(1976—),女,副教授,CCF 专业会员,主要研究领域为软件可靠性.