doi:10.11918/j.issn.0367-6234.2016.08.029

SRGM 建模类别与性能分析

张 策[1,2],孟凡超[2],万 锟[2],陈智朋[2],刘宏伟[1],崔 刚[1]

(1.哈尔滨工业大学 计算机科学与技术学院,哈尔滨 150001; 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)

摘 要: 针对软件可靠性增长模型 SRGM(software reliability growth model)在可靠性评估与保障中的重要作用,为全面掌握 SRGM 的建模与工作机理,对 SRGM 的典型建模过程以及不同模型间的性能差异进行深入研究.首先剖析了 SRGM 建模的基础假设和含义,梳理了 SRGM 的发展演化历程;然后分析了两类基本 SRGM 建模流程与关联,针对考虑更多真实测试情况的建模趋势,对不完美排错相关与考虑测试工作量 TE (Testing-Effort)相关的 SRGM 建模过程进行了剖析;最后选取 8 个典型的模型在 4 个失效数据集上进行实验,依据度量与拟合结果进行了模型差异化的深入分析.研究分析表明,客观上不同失效数据集间的差异以及主观上研究人员对测试过程认知的差异是造成 SRGM 性能差异的主要根源.进一步建立更为准确与全面的 SRGM,在有限的数据集上选取出优秀的 SRGM 已成为当前研究中亟待解决的难题.

关键词: 软件可靠性增长模型;不完美排错;测试工作量;度量;预测

中图分类号: TP311 **文献标志码**: A **文章编号**: 0367-6234(2016)08-0171-08

Analysison SRGM modeling categories and performances

ZHANG Ce[1,2], MENG Fanchao[2], WAN Kun[2], CHEN Zhipeng[2], LIU Hongwei[1], CUI Gang[1]

(1.School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China;
2.School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai 264209, Shandong, China)

Abstract: In terms of the importance of SRGM (Software Reliability Growth Model) in evaluating and ensuring reliability, in order to grasp the modeling and working mechanism of SRGM, the typical process of SRGM modeling and the differences of performance in different models are studied in this article. First, the fundamental assumptions and the meaning of SRGM modeling are illustrated, and the development of SRGM is summarized. Second, the modeling processes and the relationship of two basic types of SRGM are analyzed. For the tendency of considering more real testing factors into SRGM, the SRGM modeling process relative to the imperfect debugging and TE (Testing-Effort) are discussed. Finally, the performances of 8 typical models selected are compared using 4 published failure data sets, and analyses on the differences are illustrated. The results indicate that the objective differences in different failure data sets and subjective differences in cognition of testing process by different researchers are the main causes that account for the different performances of SRGMs. Further establishing a more accurate and comprehensive SRGM and selecting excellent ones on finite failure data sets are the problems that must be solved in the future.

Keywords: software reliability growth model (SRGM); imperfect debugging; testing effort (TE); measurement; prediction

在软件发布前的测试阶段以及之后的运行阶段,软件可靠性增长模型 SRGM(software reliability growth model)是用来定量建模可靠性增长过程的重要数学工具[1-3],已获得了广泛应用.SRGM 采用基于随机过程理论的数学微分方程(组)来描述故障检测与移除过程[4-6],使得与可靠性 $R(t)$ 直接依赖的参变量被求解出来,进而为可靠性的提高与评测,测试资源分配以及最优发布提供决策依据.目前,已有数百个 SRGMs 被提出,其建模过程类别存有较大差异,且均在有限个数的失效数据集上表现出良好的性能[7-9],但这些模型隐藏的内在本质差异在当前研究中却鲜有提及.本文探讨与挖掘差异的根源有助于洞悉 SRGM 在建模描述测试过程的本质,也为提出性能更优的模型和在实际工程中选择 SRGMs 提供必要的科学依据.

收稿日期: 2014-08-28
基金项目: 国家科技支撑计划(2014BAF07B02); 山东省科技攻关项目(2011GGX10108, 2010GGX10104)
作者简介: 张 策(1978—),男,讲师;博士研究生;
刘宏伟(1971—),男,教授,博士生导师;
崔 刚(1949—),男,教授,博士生导师
通信作者: 张 策, zhangce@hitwh.edu.cn

1 NHPP 类 SRGM 建模基础与发展历程

1.1 NHPP 类 SRGM 建模基础假设及可靠性增长含义

现有的上百个 SRGM 建立了失效检测/改正个

数与执行时间之间的对应关系，可以分为[10]：指数型和 S 型模型.指数型模型以 Geol-Okumoto 模型（记为 G-O）[9] 及其改进为代表；S 型模型以 Yamada 模型[11] 为代表. 这些非齐次泊松过程 NHPP（non-homogeneous poisson process）类 SRGMs 的建立均是基于以下公共基础假设[1, 6, 9, 11-15]，并在此基础上进行更为靠近实际测试过程的假设.

1）软件失效随机发生，失效观察与故障移除满足非齐次泊松过程 NHPP.

2）定义随机计数过程$[N(t),\ t\geqslant 0]$. 其中 $N(t)$ 表示到$[0,\ t]$内累计检测到故障数量，其均值函数为 $m(t)$，$m(0)=0$，基于 NHPP 基本性质可得

$$\Pr[N(t)=\ n]=\frac{[m(t)^n]'\times \mathrm{e}^{-m(t)}}{n!},(n=\ 0,1,2,\cdots).$$

$$\lambda(t)=\frac{\mathrm{d}m(t)}{\mathrm{d}t}.$$

这里，假定上次失效的时间为 $t(t\geqslant 0,\ x>0)$，则在$(t,\ t+x]$内可靠性如下.显然，$R(x\mid t)$ 与 $m(t)$ 紧密关联，因而现有 SRGM 研究的核心是如何找到最适宜的 $m(t)$ 函数，使得 $R(x\mid t)$ 能够得到有效提高.

$$R(x\mid t)=\mathrm{e}^{-(m(t+x)-m(t))}.$$

设 T 为软件发布时间，运行阶段的可靠性表示为

$$R(x\mid T)=\mathrm{e}^{-l(T)x}.$$

基于上述分析，可以认为，SRGM 基于失效数据描述软件测试过程中累积检测的故障数量，TE 与测试时间等的数学关系[14, 16]，是实现建模软件可靠性提高过程的数学工具.

1.2 SRGM 发展演变概要分析

不同 SRGMs 的建模假设在上述基础上，依据对测试过程的认知，又进行了不同的假设，使得建立的 SRGMs 差异较大.以 G-O 模型为起点，SRGM 发展经历了如图 1 所示的历程.

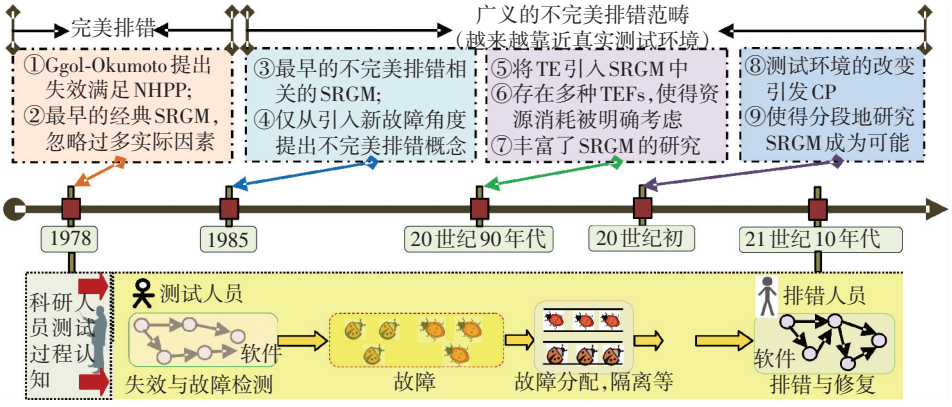


图 1 近 40 年来 SRGM 发展历程

Fig.1 Development history of SRGM in the past 40 years

从 1970 年开始，SRGMs 逐渐得到了广泛的研究.早在 1978 年，文献[9]使用一个 NHPP 作为随机过程来描述失效过程，后来文献[11]修改了 G-O 模型并将 TE 引入到 NHPP 模型中获得了更好的效果. SRGMs 通常基于失效数据[12]（失效个数\失效时间\TE\失效严重程度或连续失效的时间间隔）来建立软件测试过程的数学关系描述模型.因而，随着研究人员对测试过程认知的深入，SRGM 正在将真实测试环境（例如，不完美排错[17-20]）融入建模中.

2 两类基本 SRGM：指数型与 S 型

本文从相对统一的建模角度[10] 来分析两类 SRGMs 建模流程与关联.在建模基础上，增加如下假设：1）引起失效发生的故障立即被排除，无新故障引入；2）$(t,\ t+\Delta t)$内累积检测到的故障数量与当前软件中剩余的故障数量成正比；3）故障检测率与当前剩余故障数量成正比，且该比例随着故障排除而线性增加；4）软件中存在相互独立与依赖的两种类型故障.

基于上述假设，可以得到以下微分方程

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t}=f(t)(a-m(t)),$$

其中

$$\varphi(t)=b\left(r+(1-r)\frac{m(t)}{a}\right).$$

式中：a 为测试之初故障总数；b 为常量；$\varphi(t)$ 为时间相关的故障检测率，由于这里认为被检测到的故障立即被排除，因此也被称为故障移除率；r 为弯曲参数，用以描述独立不相关故障比例. 在 $t=0$，$m(t)=0$ 的初始条件下求解可得到

$$m(t)=a\left(\frac{1-\mathrm{e}^{-bt}}{1+((1-r)/r)\,\mathrm{e}^{-bt}}\right). \tag{1}$$

当 $r=1$ 时，式(1)可变为

$$m(t)=a(1-\mathrm{e}^{-bt}), \tag{2}$$

此为文献[9]提出的经典的 G-O 模型.因其 $m(t)$ 曲线呈现指数性增长趋势，被称为指数性模型，同时也被叫做 Musa 模型.

当令 $\beta=\frac{(1-r)}{r}$ 时，式(1)可变为

$$m(t)=a\left(\frac{1-\mathrm{e}^{-bt}}{1+\beta\mathrm{e}^{-bt}}\right),$$

此为 Inflection-S 型 SRGM[21]．显然若 $\beta=0$，则该 S 型模型又转化为指数型的 G-O 模型．另一种典型的 S 型模型是延迟型，是 G-O 模型的一种改进，其形式为

$$m(t)=a(1-(1+bt)\mathrm{e}^{-bt}).$$

图 2 给出了两类 SRGM 的基本形状，其中设定 $a=45$，$r=0.04$，$b=0.35$，$t\in[0,20]$ 仅用以展示曲线形状．

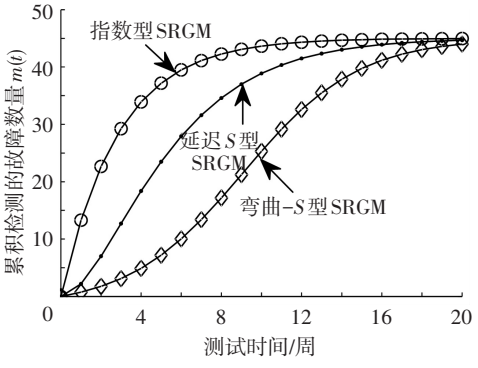


图 2　两类基本的 SRGMs：指数型与 S 型

Fig.2　Two typical SRGMS: exponential and S type

3　不完美排错与考虑 TE 的 SRGM 建模分析

实际的测试过程中含有大量的随机因素，因而从获取更为准确的 SRGM 角度出发，试图兼顾到更多真实情况的不完美排错相关的 SRGM 一直是研究人员关注的重点．此外，软件测试与成本紧密相关，易知融入测试工作量 TE[22] 到 SRGM 中是一个重要研究分支．

3.1　不完美排错相关的 SRGM

本文以经典的不完美排错 SRGM 模型：Pham 模型[8] 为示例进行阐释．Pham 模型建立的微分方程为

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t}=b(t)\times[a(t)-m(t)],\qquad(3)$$

该微分方程所做的假设为：$[0,t]$ 内累积检测到的故障个数 $m(t)$ 与剩余的故障个数 $[a(t)-m(t)]$ 成比例，比例函数为 $b(t)$．同时，文献[7]认为，$a(t)=a(1+rt)$，即 $a(t)$ 是测试时间 t 的线性增函数，这表明测试中会引入新故障，因而被称为不完美排错模型；$b(t)$ 是故障检测率，设定 $b(t)=\frac{b(1+\sigma)}{1+\sigma\mathrm{e}^{-b(1+\sigma)t}}$，$\sigma$ 是反映测试人员的学习因子．在 $m(0)=0$，$a(0)=0$ 的初始条件下，可求解得到

$$m(t)=\frac{a}{1+\sigma\mathrm{e}^{-b(1+\sigma)t}}\left[(1-\mathrm{e}^{-b(1+\sigma)t})\left(1-\frac{r}{b(1+\sigma)}\right)+rt\right].$$

验证的核心是将求解得到的 $m(t)$ 与真实的失效个数进行比较[23-26]．在比较内容上包括：拟合与预测．拟合是基于拟合标准计算 $m(t)$ 与真实失效数据集之间的偏差，通常偏差值越小越好；预测是求解并画出预测曲线，曲线越快趋于 0 效果越好．其具体步骤为：1）基于失效数据集中不同时间点的累积故障数量对 $m(t)$ 表达式中的参数进行估计：可以采用 LSE(least squares method)，MLE(maximum likelihood method) 或者采用数值求解软件(例如 Matlab，Labview，SPSS 等)给出数值解；2）将参数估计结果代入 $m(t)$ 中，求解出 $m(t)$ 的表达式；3）将 $m(t)$ 与真实的失效个数进行比较：拟合与预测；4）基于评价结果给出模型的优劣评价．

3.2　考虑 TE 时的 SRGM 建模

故障检测与修复是在 TE 的消耗下进行的，因而 SRGM 中应考虑到 TE 的存在．这样，在式(3)的基础上，当考虑 TE 时，可以得到以下文献所采用的模型[1, 6, 10, 12-15]

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t}\times\frac{1}{w(t)}=b(t)\times[a-m(t)],\qquad(4)$$

式中 $w(t)$ 为测试工作量消耗率函数，满足 $w(t)=\frac{\mathrm{d}W(t)}{\mathrm{d}t}$，通常 $W(0)=0$，但也存在 $W(0)\neq0$ 的情况[15]，代表测试之初需要消耗部分测试资源做准备．这里主要关注 $w(t)$ 的参与，这样在 $b(t)=b$，$m(0)=0$ 的初始条件下，可求解得到

$$m(t)=a[1-\mathrm{e}^{-b(W(t)-W(0))}],\qquad(5)$$

鉴于式(4)中考虑到了 TE，因而使其能够将测试资源的影响纳入到 SRGM 中．当 SRGM 中考虑到 TE 时，其验证步骤为：1）基于 TE 数据，对 $W(t)$ 中参数进行估计，本步骤估计中采用 LSE[1] 的方法较多；2）再对 $m(t)$ 中剩余的参数进行估计；3）后续步骤与前述相一致．

4　SRGMs 性能验证与分析

4.1　参与比较的模型、数据集及评价标准

本文精心选择 8 个典型的不完美排错相关以及考虑到 TE 的 SRGMs 在数据集 $DS_1\sim DS_4$[23-26] 上来进行验证，见表 1．这些公开发表的 4 个数据集已被广泛用来作为分析 SRGMs 的性能．其中 DS_1 和 DS_2 记录有失效发生的时间和累计检测到的故障数量，用以验证模型①～⑤的性能；DS_3 和 DS_4 还包括描述测试资源消耗的 TE 数据，用以验证模型⑥～⑧的性能．

目前，对 SRGM 性能的验证，主要采用以下拟合与预测标准：MSE，MEOP，Variation，RMS-PE，TS 和 BMMRE 越小，R-square 越接近于 1，这表明拟合效果越好；RE 越快趋近于 0 表明预测效果越好．

表 1 参与比较的模型及建模过程

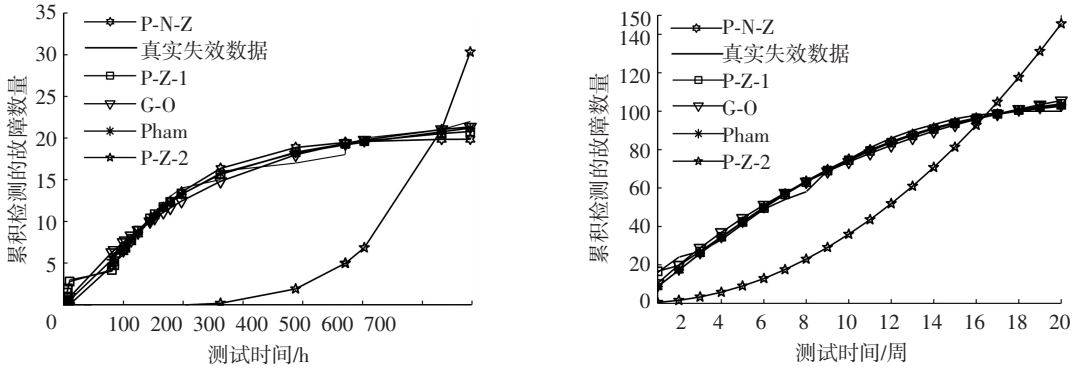
Tab.1 The selected models for comparison and modeling process

模型	类型	建模过程	$m(t)$
①G-O 模型[9]	指数模型+完美排错	$\frac{\mathrm{d}m(t)}{\mathrm{d}t}=b[a-m(t)]$；$a(t)=a$；$b(t)=b$	$m(t)=a[1-\mathrm{e}^{-bt}]$
②P-N-Z 模型[8]	不完美排错+S 形状	$\frac{\mathrm{d}m(t)}{\mathrm{d}t}=b(t)[a(t)-m(t)]$；$a(t)=a(1+\alpha t)$；$b(t)=\frac{b}{1+\beta\mathrm{e}^{-bt}}$. $a(t)$是时间 t 的线性函数；$b(t)$是非降的弯曲 S 型函数	$m(t)=\frac{a[(1-\mathrm{e}^{-bt})(1-\frac{\alpha}{b})+\alpha t]}{1+\beta\mathrm{e}^{-bt}}$
③P-Z-2 模型[27]	不完美排错	$\frac{\mathrm{d}m(t)}{\mathrm{d}t}=b(t)[a(t)-m(t)]$；$a(t)=a(1+rt)^2$；$b(t)=\frac{r^2t}{rt+1}$	$m(t)=a(1+rt)(rt+\mathrm{e}^{-rt}-1)$
④P-Z-1 模型[7]	不完美排错	$\frac{\mathrm{d}m(t)}{\mathrm{d}t}=b(t)[a(t)-m(t)]$；$a(t)=c+a(1-\mathrm{e}^{-\alpha t})$；$b(t)=\frac{b}{1+\beta\mathrm{e}^{-bt}}$	$m(t)=\frac{(c+a)(1-\mathrm{e}^{-bt})-\frac{ab}{b-\alpha}(\mathrm{e}^{-\alpha t}-\mathrm{e}^{-bt})}{1+\beta\mathrm{e}^{-bt}}$
⑤Pham 模型[8]	不完美排错	$\frac{\mathrm{d}m(t)}{\mathrm{d}t}=b(t)[a(t)-m(t)]$；$a(t)=a(1+rt)$，$b(t)=\frac{b(1+\sigma)}{1+\sigma\mathrm{e}^{-b(1+\sigma)t}}$. σ 是学习因子，反映测试人员测试能力的提高程度	$m(t)=\left[\frac{a}{1+\sigma\mathrm{e}^{-b(1+\sigma)t}}\right]\times\left[(1-\mathrm{e}^{-b(1+\sigma)t})\left(1-\frac{r}{b(1+\sigma)}\right)+rt\right]$
⑥Huang[6]	TE	$\frac{\mathrm{d}m(t)}{\mathrm{d}t}\times\frac{1}{w(t)}=b[a-m(t)]$；Logistic TEF：$W(t)=\frac{W}{1+A\mathrm{e}^{-\alpha t}}$	$m(t)=\frac{a}{1-b}[1-\mathrm{e}^{-bW^*(t)}]$；$W^*(t)=\frac{W}{1+A\mathrm{e}^{-\alpha t}}-\frac{W}{1+A}$
⑦Yamada-TEF[11]	TE	$\frac{\mathrm{d}m(t)}{\mathrm{d}t}\times\frac{1}{w(t)}=b[a-m(t)]$；Weibull TEF：$W(t)=W[1-\mathrm{e}^{-\beta t^{\sigma}}]$	$m(t)=a[1-\mathrm{e}^{-bW^*(t)}]$；$W^*(t)=W[1-\mathrm{e}^{-\beta t^{\sigma}}]$
⑧SEWTEFIDM[28]	不完美排错+TE	$\frac{\mathrm{d}m(t)}{\mathrm{d}t}\times\frac{1}{w(t)}=\varphi(t)\times[a-m(t)]$；$\varphi(t)=b\left[r+(1-r)\frac{m(t)}{a}\right]$. r 是弯曲率，表示软件中不相关类型故障的比率；Exponentiated Weibull (EW)类型 TEF：$W(t)=W(1-\mathrm{e}^{-\varphi t^{\sigma}})^{\theta}$	$m(t)=\frac{a[1-\mathrm{e}^{-bW(t)}]}{1+[(1-\gamma)/\gamma]\mathrm{e}^{-bW(t)}}$

4.2 结果与分析

4.2.1 不完美排错相关的 SRGM 性能评价

首先，基于数据集 DS_1 和 DS_2 进行的参数估计，本文给出了不完美排错相关的 5 个模型，即模型①~⑤在 DS_1 和 DS_2 上的拟合效果图，如图 3 所示.



(a) 模型①~⑤在 DS_1 上的拟合效果　　(b) 模型①~⑤在 DS_2 上的拟合效果

图 3 模型①~⑤在 DS_1 与 DS_2 上的拟合结果

Fig.3 Comparison of goodness of fitting of the models ① ~ ⑤ in DS_1 and DS_2

除模型③即 P-Z-2 以外的 4 个模型与真实的失效数据曲线拟合的均较好.可以看出此 4 个模型呈现出了与真实的失效数据曲线一样的凸形状，而模型③却呈现凹形状.造成该现象的原因是，虽然模型③的建模与其余 4 个模型在本质上相一致，但故障检测率 $b(t)$ 与软件中总故障个数 $a(t)$ 自行设定的结果使得求解得到的 $m(t)$ 与真实失效数据曲线走向发生严重背离.

为了进一步区分其余 4 个模型的细微差异，本文计算了其在拟合标准上的具体数值，见表 2.

表 2　SRGM：模型①~⑤性能比较标准结果

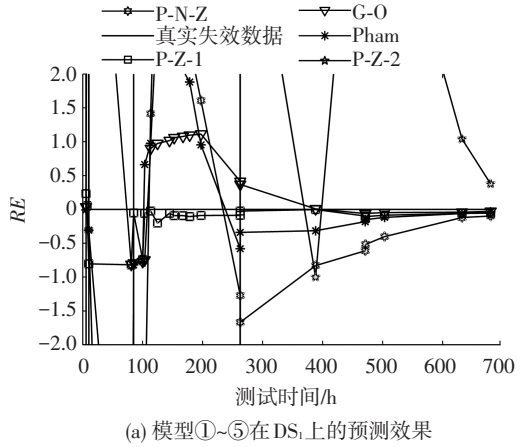
Tab.2　SRGM：performance comparison results for the models ①~⑤

标准		模型				
		①	②	③	④	⑤
MSE	DS_1	1.244 251 43	1.366 874 16	106.120 284 14	0.352 001 32	0.835 580 91
	DS_2	11.617 109 55	8.495 693 69	881.411 958 22	5.297 025 75	9.549 833 14
MEOP	DS_1	0.942 832 46	1.024 061 53	9.521 919 90	0.560 015 42	0.833 941 41
	DS_2	3.226 078 32	2.470 778 60	29.005 837 58	2.304 032 14	2.765 673 76
Variation	DS_1	1.150 279 51	1.298 439 27	18.148 231 26	0.607 363 04	0.963 642 79
	DS_2	3.500 644 04	3.108 782 71	43.590 094 57	2.361 355 53	3.286 218 91
RMS-PE	DS_1	1.152 992 85	1.329 194 01	19.969 494 03	0.607 396 19	0.972 392 14
	DS_2	3.501 818 56	3.145 324 54	46.989 249 14	2.361 367 82	3.322 004 62
R-square	DS_1	1.014 260 70	1.131 810 31	3.124 645 24	0.987 754 36	1.072 285 47
	DS_2	1.001 205 05	1.065 290 29	2.905 597 44	0.992 101 37	1.062 454 36
TS	DS_1	8.492 968 73	8.901 633 66	78.434 052 11	4.517 286 11	6.959 847 20
	DS_2	4.539 979 78	3.882 435 94	39.545 251 78	3.065 638 33	4.116 259 70
BMMRE	DS_1	0.334 718 38	112.635 879 74	643.066 534 06	0.058 066 74	0.551 701 00
	DS_2	0.077 979 29	0.084 503 90	4.411 246 80	0.035 738 05	0.088 193 28

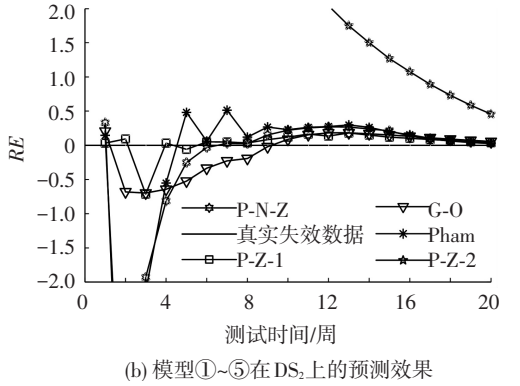
表 2 中最好的指标采用下划线来表示，其次的指标用波浪线表示，再次的指标用双下划线表示，表 3、4 中也采用同样方法.容易看出，模型④即 P-Z-1 模型在 DS_1 和 DS_2 上的拟合效果是最优的，仅在 DS_2 上的 *R*-square 指标上是次优的，且与最好的指标相差在小数点后第 3 位上.模型④最优的原因可以解释为非降的弯曲 *S* 型故障检测率函数 $b(t)$，以及软件中总故障个数 $a(t)$ 随时间基本满足指数增加的事实.次优的是模型⑤即 Pham 模型，其具有 6 个次优的指标；再次是模型①即G-O模型.而模型③即P-Z-2模型的指标是最差的，这与图 2中其曲线严重偏离真实的失效数据曲线相一致.

另外，虽然模型①G-O 模型是最早提出的 SRGM，是后续模型研究的根基，但由于其设定 $b(t)$ 和 $a(t)$ 为常量，造成一定误差出现，这使得其性能不如前者.在预测性能上，图 4 给出了 5 个模型的 *RE* 曲线.

在图 4(a)描述 DS_1 的预测曲线中，可以明显看出，模型④即 P-Z-1 最快地趋近于 0，预测效果最好，其次是模型①即 G-O，再次是模型⑤即 Pham，模型②即 P-N-Z 仅比模型③强，而模型③在有限的测试时间内已经无法准确地表现出预测能力.整体而言，5 个模型的性能依次为：④>①>⑤>②>③.在图 4(b)中，除模型③严重偏离 0 基准线外，其余 4 个模型在预测性能上基本相同，表现出了优异性能，这里不再进行区分.



(a) 模型①~⑤在 DS_1 上的预测效果



(b) 模型①~⑤在 DS_2 上的预测效果

图 4　模型①~⑤在 DS_1 与 DS_2 上的预测结果

Fig. 4 Prediction results for the models ①~ ⑤ in DS_1 and DS_2

另外，图 4 两幅图中均显示出自测试时间1/2开始，模型的预测性能开始逐渐收敛趋近于 0，呈现增强的态势.这是由于，测试半程以后，已积累了适量

的预测所需数据,测试环境与策略和技术开始变得稳定,这些使得模型的预期性能开始变好.

4.2.2 考虑 TE 的 SRGM 性能评价

通常,含有 TE 的 SRGM 需要先基于实际消耗的 TE 数据进行计算,再对累积失效的故障数量 $m(t)$ 进行评估.表 3 给出了模型⑥~⑧中所采用的 TEF 在 DS_3 和 DS_4 上的性能比较.

表 3 SRGM:⑥~⑧模型中 TE 性能比较标准结果

Tab.3 SRGM: TE performance comparison results for the models ⑥~⑧

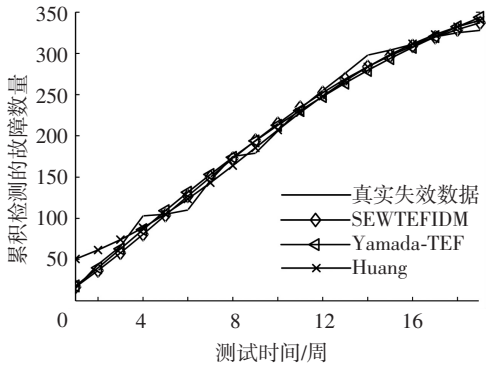
标准		模型		
		⑥	⑦	⑧
MSE	DS_3	1.627 199 73	0.902 244 91	0.858 088 03
	DS_4	37 786.668 652 51	19 335.169 645 90	17 031.677 981 16
MEOP	DS_3	1.092 982 28	0.891 650 75	0.934 991 39
	DS_4	174.631 318 55	127.795 252 91	137.373 323 27
Variation	DS_3	1.322 180 31	0.984 506 31	0.954 068 31
	DS_4	202.391 380 78	146.942 389 07	135.189 783 58
RMS-PE	DS_3	1.325 824 88	0.987 210 30	0.954 810 56
	DS_4	203.317 733 16	148.271 698 69	135.596 974 49
R-square	DS_3	0.968 030 04	1.012 630 88	1.004 884 96
	DS_4	0.972 672 96	1.020 165 62	1.010 836 74
TS	DS_3	4.542 893 45	3.382 784 55	3.298 967 53
	DS_4	3.024 515 15	2.163 517 71	2.030 556 98
BMMRE	DS_3	0.106 693 36	0.084 239 21	0.070 367 66
	DS_4	0.071 295 74	0.086 829 56	0.060 521 59

从表 3 可以看出,模型⑦和⑧中所采用的 Weibull(威布尔)和 Exponentiated Weibull(指数威布尔)类型的 TEF 在 DS_3 和 DS_4 上表现得较好,这为接下来的 $m(t)$ 性能评估贡献了积极的因素.图 5 给出了模型⑥~⑧在失效数据方面的拟合效果.

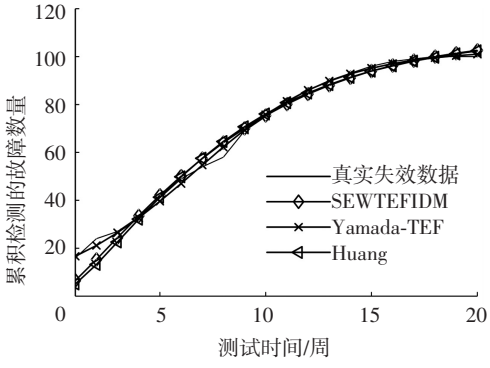
从图 5 直观上看,3 个模型的 $m(t)$ 与真实的失效数据曲线基本发生了重合,表明在 DS_3 和 DS_4 上拟合效果较好.表 4 通过计算性能比较标准值用以明确分辨 3 个模型的细微差异性.

从表 4 可以看出,在 DS_3 上,模型⑧表现最优(7 个标准中有 5 个最优,2 个次优);模型⑦次之(7 个标准中有 2 个最优,5 个次优);而模型⑥则无最优和次优值,因而最差.在 DS_4 上,出现了颠覆性的结果:模型⑥出现了 7 个最优的标准值,因而性能最优;模型⑧占据了全部 7 个标准值中的次优结果,这使得其性能为三者居中;而模型⑦则最差.

由表 1 可看出,模型⑥、⑦和⑧在建模本质上相差不大,但所采用的 TEF 存有很大差异使得 3 个模型性能各异. 另外模型⑧中的 $\varphi(t)=b\left[r+(1-r)\dfrac{m(t)}{a}\right]$ 因参数 r 的变动使得其能够适用多种情况,因而其在 DS_3 和 DS_4 上的整体表现尚可.图 6 所显示出的预测性能曲线表明 3 个模型的预测能力均较好(这里不再进行严格区分),尤其从 $t=12$ 开始,曲线开始向 0 标准收敛靠近.



(a) 模型⑥~⑧在 DS_3 上的拟合效果



(b) 模型⑥~⑧在 DS_4 上的拟合效果

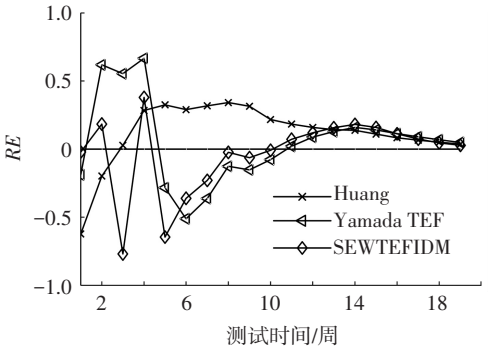
图 5 模型⑥~⑧在 DS_3 与 DS_4 上的拟合结果

Fig.5 Comparison of goodness of fitting of the models ⑥~⑧ in DS_3 and DS_4

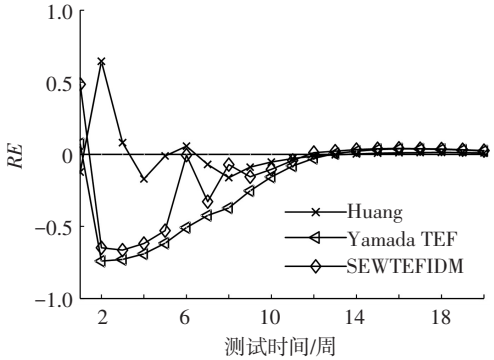
表 4 SRGM:⑥~⑧模型性能比较标准结果

Tab.4 SRGM: performance comparison results for the models ⑥~⑧

标准		模型		
		⑥	⑦	⑧
MSE	DS_3	147.120 391 25	109.739 649 04	87.056 023 03
	DS_4	1.886 788 52	17.017 938 36	12.594 651 08
MEOP	DS_3	9.671 658 45	8.864 528 42	7.978 799 49
	DS_4	0.988 650 77	2.873 292 69	2.731 786 05
Variation	DS_3	12.933 983 23	10.764 898 94	9.662 615 72
	DS_4	1.423 142 42	4.573 391 64	3.840 638 03
RMS-PE	DS_3	13.079 585 50	10.765 584 66	9.686 666 77
	DS_4	1.427 501 57	4.676 177 59	3.901 701 59
R-square	DS_3	0.910 343 72	0.984 660 23	1.017 969 15
	DS_4	1.016 725 43	1.145 663 45	1.101 967 13
TS	DS_3	5.408 170 79	4.670 848 12	4.160 192 00
	DS_4	1.829 643 55	5.494 881 73	4.727 134 36
BMMRE	DS_3	0.185 852 18	0.063 756 81	0.064 760 01
	DS_4	0.021 899 91	0.180 619 23	0.126 693 32



(a) 模型⑥~⑧在 DS_3 上的预测效果



(b) 模型⑥~⑧在 DS_4 上的预测效果

图 6　模型⑥~⑧在 DS_3 与 DS_4 上的预测结果

Fig.6　Prediction results for the models ⑥~⑧ in DS_3 and DS_4

4.2.3　模型性能的两面性以及需考虑的实际问题

1）特别指出，模型③即 P-Z-2 在 DS_1 和 DS_2 上表现得差强人意，但在某些数据集上却很优异，另一方面，这里表现优异的模型也存在于某些数据集上偏差较大.这说明一个模型不可能适应于所有的数据集.

2）另外，一个不容忽视的事实是，DS_1 的测试时间 t 是以数值较大的时钟时间为单位，与大部分失效数据集的以周为单位相迥异；DS_1 以累积失效个数为顺序记录，与绝大部分以失效时间为顺序记录累积失效个数方式也极为不同，这些综合因素使得其与模型③中 $m(t)$ 表达式存有很严重的不匹配现象，因而造成模型③性能表现最差.当然，这种数据集在当前可收集到的数据集中并不多见，本文这里特用以说明该事实.

3）显然，对测试过程的深入认识以及由此建立考虑更多实际因素的数学模型，能够刻画更多的测试细节，但随着引发的求解难度也急剧增加.例如，当所建立的模型考虑到更多的细节，且存有一定的限制条件时，欲获得 $m(t)$ 的表达式已经变得无法解析求解，此时只能采用非解析方法（借助于计算机利用数学求解软件来给出数值解，例如 Matlab，Labview 等），甚至在有些情况下只能给出上下界.

4）当考虑 TE 到 SRGM 中时，当前研究主要还是对 $m(t)$ 的拟合与预测结果进行最终的评估.实际上，TE 代表着测试成本的支出，虽然高可靠性是 SRGM 的本质要求，但追逐这一目标还需要兼顾到成本的花销.

4.3　SRGM 性能差异化总体分析

从上述 8 个模型在 4 个数据集上的实验结果与性能分析可以看出：

1）直观上，不同 SRGMs 的差异体现在所求解得到的 $m(t)$ 上，但这种差异的根源缘由以下两点决定：①失效数据集是决定 SRGMs 差异的客观因素：不同公司、不同测试环境与策略，直接导致失效数据的收集、记录等存有不一致；②建模假设的内容是决定 SRGMs 差异的主观因素：从上述实验可以看出，即便基于同一失效数据集，研究人员做出的假设上的差异直接导致数学模型的不同.

2）SRGMs 的性能强烈地依赖于数据集：例如，模型⑧模型在 DS_3 上表现得优异，但在 DS_4 上却不理想.

3）由于不同模型对测试认知过程中的差异，以及所建立的模型的差别，使得其在同一个数据集上的性能差异较大.

4）整体上，在数据集较大的情况下，拟合效果有效的 SRGM，其在预测性能上也相对表现优秀：这是因为，拟合和预测在时间的选取上具有很强的相对性，可能存在交叉而造成的.

5）本文给出的 8 个模型在 4 个失效数据集上的结果表明并不存在一个模型能够适应所有的数据集.对于该事实的确立，本文对 26 个模型在可搜集到的 15 个失效数据集上的实验结果已表明.

5　结　论

1）由于不同公司在测试过程中存在测试环境、策略、技术以及数据收集等因素上的差异，使得不同失效数据集的差异较大，这为不同 SRGMs 的性能差异带来了客观上的解释.

2）SRGM 是客观与主观结合下研究人员去建模描述不同失效数据集的过程，因而造成一个 SRGM 选择上的难题：能够准确拟合与预测所有失效数据集的 SRGM 并不存在，但在有限个数的数据集上，采用合适的评价标准是可以优选出相对最优的 SRGM，这是近期 SRGM 研究的一个方向.

3）SRGM 的本质即是提出能够更为准确地度量拟合过去的失效数据，同时又能准确地预测未来失效情况的数学模型.这样，考虑到更多真实的测试环境要素，建立全面的 SRGM 已成为当前研究中亟待解决的难题.

参考文献

[1] AHMAD N, BOKHARI M U, QUADRI S M K, et al. The

exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy: a performance analysis [J]. International Journal of Quality & Reliability Management, 2008, 25(2): 211-235. DOI: 10.1108/02656710810846952.

[2] DOHI T, MATSUOKA T, OSAKI S. An infinite server queuing model for assessment of the software reliability[J]. Electronics and Communications in Japan (Part III: Fundamental Electronic Science), 2002, 85(3): 43-51. DOI: 10.1002/ecjc.1078.

[3] YANNIPRINCY B, SRIDHAR S. Prediction of software reliability using cobb-douglas model in SRGM[J]. Journal of Theoretical and Applied Information Technology, 2014, 62(2): 355-363.

[4] YAMADA S, ICHIMONRI T, NISHIWAKI M. Optimal allocation policies for testing-resource based on a software reliability growth model[J]. Mathematical and Computer Modelling, 1995, 22(10): 295-301. DOI: 10.1016/0895-7177(95)00207-I.

[5] OKAMURA H, DOHI T. A novel framework of software reliability evaluation with software reliability growth models and software metrics [C]// Proceedings of the IEEE 15th International Symposium on High-Assurance Systems Engineering (HASE), Miami Beach: IEEE, 2014: 97-104. DOI: 10.1109/HASE.2014.22.

[6] HUANG C Y, LYU M R. Optimal release time for software systems considering cost, testing-effort, and test efficiency [J]. IEEE Transactions on Reliability, 2005, 54(4): 583-591. DOI: 10.1109/TR.2005.859230.

[7] PHAM H, ZHANG Xuemei. An NHPP software reliability model and its comparison[J]. International Journal of Reliability, Quality and Safety Engineering, 1997, 4(3): 269-282. DOI: 10.1142/S0218539397000199.

[8] PHAM H, NORDMANN L, ZHANG Xuemei. A general imperfect-software-debugging model with S-shaped fault-detection rate [J]. IEEE Trans on Reliability, 1999, 48(2): 169-175. DOI: 10.1109/24.784276.

[9] GOEL A L, OKUMOTO K. Time-dependent error-detection rate model for software reliability and other performance measures[J]. IEEE Transactions on Reliability, 1979, R-28(3): 206-211. DOI: 10.1109/TR.1979.5220566.

[10] JHA P C, GUPTA D, YANG Bo, et al. Optimal testing resource allocation during module testing considering cost, testing effort and reliability[J]. Computers & Industrial Engineering, 2009, 57(3): 1122-1130. DOI: 10.1109/PRDC.2004.1276561.

[11] YAMADA S, HISHITANI J, OSAKI S. Software-reliability growth with a Weibull test-effort: a model and application[J]. Reliability, IEEE Transactions on Reliability, 1993, 42(1): 100-106. DOI: 10.1109/24.210278.

[12] HUANG C Y, LO J H. Optimal resource allocation for cost and reliability of modular software systems in the testing phase [J]. Journal of Systems and Software, 2006, 79(5): 653-664. DOI: 10.1016/j.jss.2005.06.039.

[13] 张 策, 孟凡超, 崔刚, 等. SRGM 中 TE 建模机制与模型比较分析[J]. 哈尔滨工业大学学报, 2015, 47(5): 32-39. DOI: 10.11918/j.issn.0367-6234.2015.05.006.
ZHANGCe, MENG Fanchao, CUI Gang, et al. Overview of modeling of TE in SRGM and comparisons for models[J]. Journal of Harbin Institute of Technology, 2015, 47(5): 32-39. DOI: 10.11918/j.issn.0367-6234.2015.05.006.

[14] HUANG C Y. Cost-reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency [J]. Journal of Systems and Software, 2005, 77(2): 139-155.

[15] AHMAD N, KHAN M G M, QUADRI S M K, et al. Modelling and analysis of software reliability with Burr type X testing-effort and release-time determination[J]. Journal of Modelling in Management, 2009, 4(1): 28-54. DOI: 10.1108/17465660910943748.

[16] 张 策, 崔刚, 刘宏伟, 等. 软件测试资源与成本管控和最优发布策略[J]. 哈尔滨工业大学学报, 2014, 46(5): 51-58. DOI: 10.11918/j.issn.0367-6234.2014.05.009.
ZHANGCe, CUI Gang, LIU Hongwei, et al. Software test resources and cost control and optimal release policy[J]. Journal of Harbin Institute of Technology, 2014, 46(5): 51-58. DOI: 10.11918/j.issn.0367-6234.2014.05.009.

[17] LIN C T. Analyzing the effect of imperfect debugging on software fault detection and correction process via a simulation framework [J]. Mathematical and ComputerModelling, 2011, 54(11/12): 3046-3064. DOI: 10.1016/j.mcm.2011.07.033.

[18] MANJULA T, JAIN M, GULATI T R. Cost optimization of a software reliability growth model with imperfect debugging and a fault reduction factor[J].Anzam Journal, 2014, 55: 182-196.

[19] JAIN M, MANJULA T, GULATI T R. Imperfect debugging study of SRGM with fault reduction factor and multiple change point[J]. International Journal of Mathematics in Operational Research, 2014, 6(2): 155-175. DOI: 10.1504/IJMOR.2014.059526.

[20] PENG R, LI Y F, ZHANG W J, et al. Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction [J]. Reliability Engineering & System Safety, 2014, 126: 37-43. DOI: 10.1016/j.ress.2014.01.004.

[21] YAMADA S, OHBA M, OSAKI S. S-shaped reliability growth modeling for software error detection [J]. Reliability, IEEE Transactions on Reliability, 1983, R-32(5): 475-484. DOI: 10.1109/TR.1983.5221735.

[22] BOKHARI M U, AHMAD N. Incorporating burr type XII testing-efforts into software reliability growth modeling and actual data analysis with applications[J]. Journal of Software, 2014, 9(6): 1389-1400. DOI: 10.4304/jsw.9.6.1389-1400.

[23] OHBA M. Software reliability analysis models[J]. IBM Journal of research and Development, 1984, 28(4): 428-443. DOI: 10.1147/rd.284.0428.

[24] WOOD A. Predicting software reliability[J]. Computer, 1996, 29(11):69-77. DOI: 10.1109/2.544240.

[25] EHRLICH W, PRASANNA B, STAMPFEL J, et al. Determining the cost of a stop-test decision (software reliability)[J]. IEEE Software, 1993, 10(2): 33-42. DOI: 10.1109/52.199726.

[26] GOPALAKRISHNA R, SPAFFORD E, VITEK J. Vulnerability likelihood: a probabilistic approach to software assurance. Technical Report 2005-06[R]. West Lafayette: Purdue University, 2005.

[27] PHAM H. An imperfect-debugging fault-detection dependent-parameter software [J]. International Journal of Automation and Computing, 2007, 4(4): 325-328. DOI: 10.1007/s11633-007-0325-8.

[28] AHMAD N, KHAN M G M, RAFI L S. A study of testing-effort dependent inflection S-shaped software reliability growth models with imperfect debugging[J]. International Journal of Quality & Reliability Management, 2010, 27(1): 89-110.DOI: 10.1108/02656711011009335.

(编辑 张 红)