文章编号:1001 - 9081(2011)07 - 1762 - 03
doi:10.3724/SP.J.1087.2011.01762

基于优化 PSO-SVM 模型的软件可靠性预测

张晓南,刘安心,刘　斌,张宏梅,青　星
(解放军理工大学 工程兵工程学院,南京 210007)
(zxn8206@163.com)

摘　要:讨论了传统软件可靠性预测模型的主要弱点;在分析传统 PSO-SVM 模型和软件可靠性预测特点的基础上,对传统 PSO-SVM 模型进行改进,建立了优化 PSO-SVM 软件可靠性预测模型。最后通过仿真结果表明,该优化预测模型具有更好的小样本适应性,训练速度快,预测精度高,能够更好地适用于软件可靠性预测。

关键词:粒子群优化算法;支持向量机;软件可靠性;预测
中图分类号:TP18;TP311　　**文献标志码**:A

Software reliability prediction based on improved PSO-SVM model

ZHANG Xiao-nan, LIU An-xin, LIU Bin, ZHANG Hong-mei, QING Xing
(*Engineering Institute of Engineer Corps, PLA University of Science and Technology, Nanjing Jiangsu* 210007, *China*)

Abstract: The major disadvantages of the current software reliability models were discussed. And then based on analyzing classic PSO-SVM model and the characteristics of software reliability prediction, some measures of the improved PSO-SVM model were proposed and an improved model was established. Lastly, the simulation results show that compared with classic models, the improved model has better prediction precision, better generalization ability and lower dependence on the number of sample, which is more applicable for software reliability prediction.

Key words: Particle Swarm Optimization (PSO); Support Vector Machine (SVM); software reliability; prediction

0　引言

软件可靠性定量评估与预测是软件可靠性工程的重要组成部分。根据软件可靠性测试阶段得到的失效数据进行软件可靠性预测,是软件可靠性定量评估和预测的主要方法之一。目前利用失效数据进行软件可靠性分析和预测的方法主要有:

1)利用基于概率与数理统计的软件可靠性增长模型(Software Reliability Growth Model,SRGM)进行软件可靠性评估与预测。该方法将软件可靠性测试过程中出现的故障现象看做随机过程,将失效数据看做是随机分布的数据,利用失效数据对函数模型进行参数估计得到模型的具体表达式,从而计算描述软件可靠性的主要参数取值。此类模型最早应用于软件可靠性工程领域,具有坚实的理论基础。但在工程实践中发现,当模型假设条件与软件实际情况不一致时,同一软件应用不同的模型得出的评价结果截然不同,从而使软件可靠性评价变得毫无意义。

2)利用时间序列分析方法进行软件可靠性预测。该方法将失效数据看做时间序列,根据现在和过去的时间序列值来预测未来的数据。此方法主要考虑故障发生过程的动态特性,与软件可靠性增长模型相比具有一定的优势。此类方法目前还处于尝试阶段,国内外学者将多元线性回归或者人工神经网络的方法[1-2]应用于该领域,并取得了阶段性的研究成果。

但是多元线性回归方法只能处理线性关系或者某种变形的线性关系,对自变量和样本的要求比较严格,并且必须保证较大的样本数量。而人工神经网络本身充满着浓厚的经验色彩,理论基础不严格,结构难于确定,并且存在拟合不足或过拟合及容易陷入局部极小点等问题。在小样本条件下,人工神经网络的效果也不够理想。

鉴于此,本文将支持向量机(Support Vector Machine,SVM)[3]和粒子群优化(Particle Swarm Optimization,PSO)算法[4]应用到软件可靠性预测的研究中,针对传统 PSO 和 SVM 算法本身固有的缺点和不足,进行了相应的优化对策分析,并构建了优化 PSO-SVM 软件可靠性预测模型,通过实例证明了优化模型对于软件早期样本数据较少情况下,具有较高的预测精度和较好的适用性。

1　传统 PSO-SVM 特点分析

传统 PSO-SVM 模型采用 PSO 算法优化 SVM 的模型参数和核参数,通过搜索并使用参数的最优值来提高 SVM 的预测精度。SVM 最早是针对二分类问题提出的最大间隔算法,后来逐步推广到非线性回归预测领域。支持向量机非线性回归预测问题与分类问题类似,都是根据给定数据计算出决策函数,然后由决策函数进行分类和预测。回归问题保留了最大间隔算法的主要特征,要最小化一个凸函数,而且非线性函数也可以通过核特征空间的线性学习器得到,不同之处主要体现在给定的数据集合。假设给定的数据集合为 $\{(x_1,y_1),(x_2,y_2),\cdots,(x_i,y_i),\cdots,(x_l,y_l)\}$,其中 $x_i \in \mathbf{R}^n, y_i \in \mathbf{R}, i = 1,2,\cdots,l$。分类问题的 y_i 取值为对应的数据种类数,如二分类

收稿日期:2010 - 12 - 24;**修回日期**:2011 - 02 - 18。

作者简介:张晓南(1982 -),男,辽宁丹东人,博士研究生,主要研究方向:军用装备保障、军用软件可靠性设计;　刘安心(1967 -),男,安徽宣城人,教授,博士生导师,主要研究方向:军事装备保障管理;　刘斌(1964 -),男,江苏丹阳人,副教授,硕士,主要研究方向:机械设计与自动化;　张宏梅(1974 -),女,安徽天长人,副教授,博士,主要研究方向:机械设计与自动化;　青星(1985 -),男,四川成都人,硕士研究生,主要研究方向:可靠性分析与设计。

问题可取 -1 和 1,而回归问题则可取任意实数。SVM 原问题可表示为

$$\min_{\omega\in\mathbf{R}^n,b\in\mathbf{R}} \quad J(\boldsymbol{\omega},\xi) = \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega} + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \qquad (1)$$

$$\text{s.t. } (\boldsymbol{\omega}\cdot\boldsymbol{\varphi}(x_i) + b) - y_i \leqslant \xi_i^* + \varepsilon;\ i = 1,2,\cdots,l$$

$$y_i - (\boldsymbol{\omega}\cdot\boldsymbol{\varphi}(x_i) + b) \leqslant \xi_i + \varepsilon;;i = 1,2,\cdots,l$$

$$\xi_i,\xi_i^* \geqslant 0;\ i = 1,\cdots,l$$

由于特征空间的维数很高,且目标函数不可微,为了方便计算,引入点积核函数技术和 Wolf 对偶理论,将问题转化为对偶问题。对偶问题是将原问题变为二次规划问题,使求解成为可能[4]。

采用 PSO 对 SVM 的模型 C、ε 和核参数 σ^2 进行优化时,种群在迭代更新过程中不断向本代最有位置和全局最有位置学习。假设种群大小为 m,第 i 个粒子的 d 维空间位置为:$x_i = \{x_{i1},x_{i2},\cdots,x_{id}\}$,速度为:$v_i = \{v_{i1},v_{i2},\cdots,v_{id}\}$,本代最优位置为:$pbest_i = \{p_{i1},p_{i2},\cdots,p_{id}\}$,全局最优位置为:$gbest = \{g_1,g_2,\cdots,g_d\}$,$d$ 由目标问题的特征属性维数决定,适应度函数根据目标问题被优化的函数而设定[9]。

2　模型适用性及优化对策分析

传统的 PSO-SVM 预测模型通过 PSO 优化 SVM 的模型参数和核参数使得最后的预测结果更好,该模型具有许多突出优点,与软件可靠性的预测特点相适应,该预测模型的优点与软件可靠性预测的特点对应关系如下:

1)SVM 通过模型参数的作用,调整结构风险与经验风险的比例关系,避免了过学习问题,提高了预测模型的泛化能力。与软件可靠性样本获取困难,样本少的特点相适应。

2)SVM 在预测模型中引入核函数,将多维输入空间转化为高维空间进行预测,解决了输入空间维数多的问题。与软件可靠性特征参数多的特点相适应。

3)传统 PSO-SVM 在高维空间中进行预测,将原本非线性的问题转化为线性问题预测,得到结果后再还原成非线性问题的解。与软件可靠性预测非线性的特点相适应。

传统 PSO-SVM 预测模型的优点虽然很多,但由于 PSO 和 SVM 算法本身固有的缺点和不足,使得该预测模型还有许多不尽如人意的地方,本文为了得到性能更加优越的软件可靠性预测模型,必须针对传统模型的缺点进行改进。该模型的缺点及改进策略的对应关系如下:

1)预测模型中对 SVM 的模型参数和核参数的编码初始化具有很强的随机性,在一定程度上限制了模型全局搜索的能力,不利于搜索到最优参数,从而会降低预测精度和效率。采用分块种群初始化策略,将参数的区间范围分为若干个小区间进行初始化,可以提高初始种群的多样性,提高预测效率和精度。

2)预测模型中 PSO 的惯性因子固定,调节全局搜索和局部搜索的能力有限,限制了模型获得最优解的能力。采用自适应惯性因子,随着迭代的进行适时修改惯性因子,延长全局搜索和局部搜索的时间,保证获得参数的最优解。

3)模型涉及低维输入空间转化为高维空间和求解二次规划问题,当输入空间维数较高时,计算量急剧增加,计算效率成为制约模型性能发挥的瓶颈。由于软件可靠性的特征参数较多,使用传统 PSO-SVM 预测模型必然使计算量很大。对 SVM 进行最小二乘支持向量机(Least Squares Support Vector Machine,LSSVM)变形,从 SVM 的内部结构上进行优化,降低 SVM 的复杂程度,从而减小预测模型的计算量,提高预测效率。

2.1　分块种群初始化机制

粒子群算法(PSO)是一种全局最优搜索算法,全局搜索是其区别于传统寻优算法的最大特点,因此在运用该算法时要充分发挥其全局搜索的特点,以期更快地获得全局最优值。然而在算法初始化时,传统的粒子群种群是在整个搜索区域内随机产生的,粒子初始位置的随机性使得种群在开始时不能保证充分地分散分布于整个搜索空间,如果能将搜索空间分块,能在一定程度上改善初始化不均匀的状况。采用分块种群初始化其主要思想是使每个粒子几乎均匀分布,假设种群粒子数为 n,则将整个搜索空间分为 n 个小区域,每个小区域中随机产生一个粒子[5]。

$$X_{i,k} \in [a_k + f_k(b_k - a_k), a_k + f_{k+1}(b_k - a_k)];\ k = 1,2,\cdots,D$$

$$\begin{cases} f_k = (i-1) \bmod C^{D-k} \\ f_D = (i-1) - \sum_{j=1}^{D-1} f_j C^{D-j} \\ C = \sqrt[D]{n} \end{cases} \qquad (2)$$

其中:a_k 和 b_k 表示粒子位置向量上第 k 维上值的取值范围,mod 表示取模,然后可以按照式(3)的方法产生第 i 个粒子的初始位置:

$$X_{i,k} = a_k + f_k(b_k - a_k) + \frac{1}{\sqrt[D]{n}}(b_k - a_k)\cdot r \qquad (3)$$

其中 r 为[0,1]内取值的随机数。

2.2　自适应惯性因子策略

粒子群算法的惯性因子 w 使得算法在粒子速度更新时具有历史记忆性,协调历史速度与全局最优与局部最优速度的关系,影响粒子的全局搜索能力与局部搜索能力之间的平衡。当迭代开始时,较大的惯性权重可以加强 PSO 算法的全局搜索能力,即搜索较大的区域,较快地定位最优解的大致位置;迭代的后期,较小的惯性权重可以加强 PSO 算法的局部搜索能力,即粒子速度减慢,有利于精细的局部搜索。由此可以看出,迭代的开始阶段决定了算法的全局搜索能力,最后阶段决定了局部搜索能力,如果能延长前后两端的搜索时间,将对算法整体性能的改善大有好处,因此采用如下的权重自适应更新方法:

$$w = w_{\min} + (w_{\max} - w_{\min}) \times \exp(-20 \times (t/t_{\max})^n) \qquad (4)$$

其中:$w_{\max}$、$w_{\min}$ 分别取 0.9 和 0.1,图 1 中由左至右为式(4)中 $(t/t_{\max})$ 的指数依次取为 2 ~ 10 时的变化曲线。对应的惯性权重变化曲线如图 1。

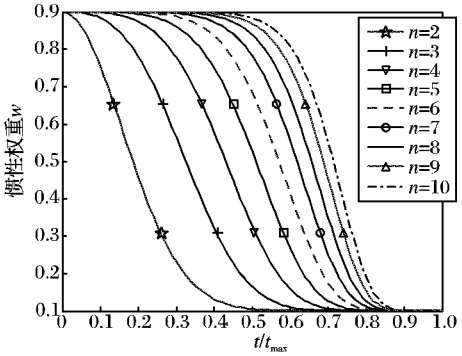

图 1　自适应惯性权重变化曲线

从图 1 中可知,与取其他值相比,当指数取 6 时,粒子的全局搜索时间和局部搜索时间都比较长。该方法使惯性权重

在迭代初期长时间保持了较大的值,在迭代后期长时间保持较小的值,从而延长了迭代初期的全局搜索时间和迭代后期的局部搜索时间,加强了在迭代初期的全局搜索强度以及迭代后期的局部搜索强度,使全局搜索能力与局部搜索能力获得良好的平衡。

2.3 LSSVM 变形策略

最小二乘支持向量机(LSSVM)的主要变形包括两点:一是采用最小二乘线性系统作为损失函数,将经验风险由偏差的一次方改为二次方,用等式约束替换标准支持向量机的不等式约束,简化约束条件;二是将求解二次规划问题替换为求解线性方程组,避免了不敏感损失函数,大大降低了学习难度,极大地提高了学习效率和训练精度。标准 SVM 的原问题可简化为:

$$\min J(\boldsymbol{\omega},\xi) = \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega} + C\sum_{i=1}^{n}\xi_i^2 \tag{5}$$

$$\text{s.t. } y_i = \boldsymbol{\varphi}(x_i)\cdot\boldsymbol{\omega} + b + \xi_i + \varepsilon$$

3 对比仿真实验

为了评价新模型的性能,以及同传统模型相比的优势和劣势,安排对比仿真实验。这里,以一个军用软件系统为例说明,表 1 列举了该军用软件系统的 13 个模块的若干度量元以及模块缺陷数。其中 SN 表示模块编号;LOC 表示模块规模(单位是行代码数);FO 表示模块扇出;FI 表示模块扇入;PATH 表示模块控制流路径;FAULTS 表示模块缺陷数。

表 1 实验数据

SN	LOC	FO	FI	PATH	FAULTS
1	29	4	1	4	0
2	29	4	1	4	2
3	32	2	2	2	1
4	33	3	27	4	1
5	37	7	18	16	1
6	41	7	1	14	4
7	55	1	1	12	2
8	64	6	1	14	0
9	69	3	1	8	1
10	101	4	4	12	5
11	120	3	10	22	6
12	164	14	10	221	11
13	270	9	1	80	17

为了评价优化模型预测软件模块缺陷数的精度。进行两次仿真实验。

实验 1 将全部 13 个数据样本分成 2 个部分,其中前 10 个作为训练集来建立模型,后 3 个作为测试集来评价模型的效果。

实验 2 将前 6 个作为训练集来建立模型,仍将后 3 个作为测试集来评价模型的效果。

将训练样本和预测样本一起进行规范化处理后[4],分别输入 BP 网络预测模型、传统 PSO-SVM 预测模型和优化的 PSO-LSSVM 预测模型中进行训练。其中,BP 预测模型采用带动量因子的 BP 预测模型,选用模型的隐层节点数为 18,训练目标为 0.000 01;按照交叉验证和深度优先搜索的算法,经过 2 轮的筛选,传统 PSO-SVM 预测模型参数选用 $C = 499$,核参数为 $\sigma^2 = 5$,传统 PSO-SVM 预测模型和改进 PSO-LSSVM 预测模型的核函数都采用径向基函数(Radial Basis Function, RBF)。模型训练过程中,对 BP 预测模型的误差曲线和改进 PSO-LSSVM 的误差曲线进行模拟,结果如图 2 和图 3 所示。

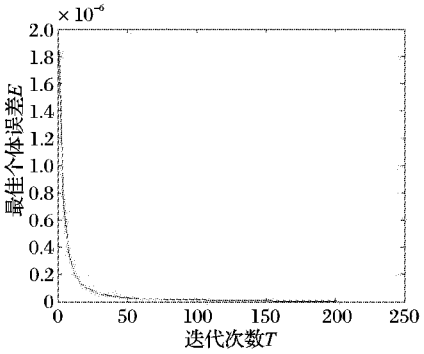

图 2 改进 PSO 训练 LSSVM 误差曲线

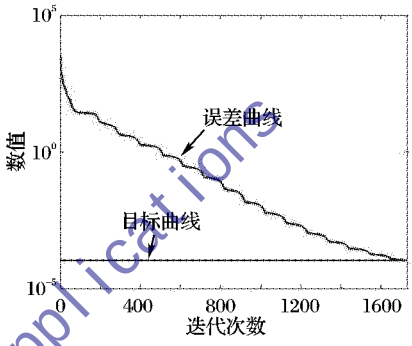

图 3 BP 网络误差曲线

从图 2、3 中可知:改进的 PSO-LSSVM 预测模型训练误差下降很快,经过约 200 代训练趋于停止;而 BP 预测模型经过 1 733 代才满足训练要求,改进 PSO-LSSVM 预测模型的训练效率明显高于 BP 预测模型。

训练结束后,用三种预测方法分别得到适用于样本数据的预测模型,将预测样本输入预测模型中进行预测。由于 BP 预测模型受初始参数影响较大,为了降低预测的随机性,对 BP 网络连续进行 10 次预测求平均值,并计算预测结果的平均百分比误差,预测结果及误差对比如表 2、3 所示(平均绝对百分比误差(Mean Absolute Percentage Error, MAPE))。

表 2 三种预测模型预测结果对比

实验序号	测试集序号	真实值	BP 预测模型	传统 PSO-SVM 预测模型	改进 PSO-LSSVM 预测模型
1	1	6	5.720 4	6.266 4	6.100 9
	2	11	10.166 6	11.474 8	11.192 4
	3	17	16.043 2	17.628 3	17.203 7
2	1	6	5.223 4	6.632 6	6.134 5
	3	11	9.199 8	12.389 5	11.207 8
	4	17	14.586 0	19.006 8	17.325 7

表 3 三种预测模型预测误差对比 %

实验序号	测试集序号	BP 预测模型	传统 PSO-SVM 预测模型	改进 PSO-LSSVM 预测模型
1	1	4.66	4.44	1.68
	2	7.57	5.22	1.75
	3	5.62	3.69	1.21
MAPE		5.95	4.45	1.54
2	1	12.94	10.54	2.24
	2	16.37	12.62	1.89
	3	14.20	11.76	1.93
MAPE		14.51	11.64	2.02

(下转第 1772 页)

况,据表中数据分析,CPU 平均使用率为 69.841%,小于测试目标 75%,表示 CPU 的在 50 个 Vuser 压力下运行稳定。同时内存三大指标每秒失效页数、每秒读磁盘页数、每秒读写磁盘页数整体居高,说明在系统读取某页时,页面缺页率高,并且磁盘读写时间比率指标为 31.723%,说明磁盘驱动器消耗 31.723% 的时间在缺页读取与写入上,且磁盘平均队列时长指标值偏低说明磁盘运行正常,由此可见内存可能为系统瓶颈。

表 3 事务成功率

事务名称	通过	停止
Action_Transaction	1719	8
cxx	1719	0
cxx2	1719	0
dl	50	0
vuser_end_Transaction	50	0
vuser_init_Transaction	50	0

表 4 系统资源利用率

项目	处理器总体消耗时间比率/%(处理器)	处理器队列长度(系统)	每秒失效页数(内存)	磁盘读写时间比率/%(磁盘)	每秒读写磁盘页数(内存)	处理器非内核消耗时间比率/%(处理器)	处理器网络消耗时间比率/%(处理器)	磁盘平均队列时长(磁盘)	可用物理内存(内存)	每秒读磁盘页数(内存)
最大值	99.869	47.000	46876.714	564.843	1936.831	94.110	3.385	5.648	789.000	369.855
最小值	25.263	0.000	30.674	3.643	0.000	25.132	0.000	0.036	709.000	0.000
平均值	69.841	5.403	9750.674	31.723	317.229	62.560	0.556	0.317	779.179	62.257
标准值	24.195	7.802	9605.018	38.219	279.618	21.786	0.556	0.382	6.352	51.923

在本测试中,系统在大量用户使用和长时间反复运行中,未出现大的不良反应,除内存缺页错误率偏高,其余包括 CPU、磁盘 I/O、网络传输等相对稳定,系统反应良好,在大吞吐量情况下系统响应时间较好,系统稳定性比较可靠。建议适当扩充内存优化系统响应时间。

4 结语

Web 应用系统性能决定着整个系统的质量,对 Web 应用系统性能的测试已成为系统上线前的必经环节,本文通过对 Web 应用系统体系结构及性能指标的研究,提出了通用的性能测试过程模型并应用该模型对本市数字城管系统综合评价子系统实施测试,找出了系统性能瓶颈。下一步工作主要研究 Web 应用系统性能测试框架的搭建以及在 Linux 系统下如何高效地完成 Web 应用系统性能测试。

参考文献:

[1] MYERS G J, BADGETT T, THOMAS T M, *et al*. 软件测试的艺术[M]. 王峰, 陈杰, 译. 2 版. 北京: 机械工业出版社, 2006.

[2] 兰景英, 王永恒. Web 系统性能测试研究[J]. 计算机技术与发展, 2008, 18(11): 90 - 93.

[3] 崔冬华, 王立群, 丁周芳. 一种改建的 Web 性能测试模型[J]. 微电子与计算机, 2009, 26(6): 179 - 182.

[4] 魏晓玲. Web 应用程序性能测试的研究与应用[J]. 信息技术, 2010, 34(8): 178 - 180.

[5] 段念. 软件性能测试过程详解与案例剖析[M]. 北京: 清华大学出版社, 2006: 13.

[6] 李健, 王亚民. 一种基于 Web 信息系统的性能测试模型[J]. 知识组织与知识管理, 2009, 184(10): 45 - 49.

[7] 浦云明, 王宝玉. 基于负载性能指标的 Web 测试[J]. 计算机系统应用, 2010, 19(5): 220 - 223.

[8] 韩庆良. 软件性能测试过程研究与应用[D]. 济南: 山东大学, 2007.

[9] 柳胜. 性能测试从零开始[M]. 北京: 电子工业出版社, 2008.

[10] 李东昱, 苗放. LoadRunner 在 Web 应用程序性能测试中的应用[J]. 产品开发与应用, 2007(10): 49 - 51.

(上接第 1764 页)

4 结语

由于改进 PSO-LSSVM 预测模型使用优化后的模型参数和核参数,预测精度明显优于未经过优化的传统 PSO-SVM 预测模型和 BP 预测模型;而随着训练集中样本数量的减少,优化 PSO-LSSVM 预测模型凭借出色的泛化性能,在训练样本较少的情况下,预测精度明显高于传统 PSO-SVM 预测模型和 BP 预测模型。由此可见,优化 PSO-LSSVM 预测模型无论在训练效率还是预测精度都比传统的 PSO-SVM 和 BP 预测模型高,在目前国内软件行业规模小、历史样本不够充足的情况下,该模型对软件可靠性的预测具有实用性;而对于某些类似历史样本较少的其他项目,也可能会成为首选的预测方法。

参考文献:

[1] BRIAND L C, MELO W L, WUST J. Assessing the applicability of fault-proneness models across object-oriented software project[J]. Software Engineering, 2002, 28(7): 706 - 720.

[2] KHOSHGOFTAAR T M, SZABOR M. Using neural networks to predict software faults during testing[J]. IEEE Transactions on Reliability, 1996, 45(3): 456 - 462.

[3] CRISTIANINI N, TAYLOR J S. 支持向量机导论[M]. 李正国, 译. 北京: 电子工业出版社, 2004.

[4] KENNEDY J, EBERHART R C. Particle swarm optimization[C]// Proceedings of IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Service Center, 1995: 1942 - 1948.

[5] PAQUET U, ENGELBRECHT A P. A new particle swarm optimizer for linearly constrained optimization[C]// Proceedings of IEEE Congress on Evolutionary Computation. New York: IEEE, 2003: 227 - 233.

[6] PAQUET U, ENGELBRECHT A P. Training support vector machines with particle swarms[C]// Proceedings of International Joint Conference on Neural Networks. New York: IEEE, 2003: 1593 - 1598.

[7] 段晓东, 刘向东, 王存睿. 粒子群算法及其应用[M]. 沈阳: 辽宁大学出版社, 2007.

[8] 朱经纷. 软件可靠性综合模型的分析和研究[J]. 计算机科学, 2009, 36(4): 181 - 184.

[9] 郑艳艳, 郭伟. 软件可靠性工程学综述[J]. 计算机科学, 2009, 36(2): 20 - 25.