基于组件的软件系统可靠性评估的不确定性研究

宋绍云
（玉溪师范学院 信息技术工程学院，云南 玉溪 653100）

摘 要：当前一些基于体系结构的软件可靠性模型，在操作剖面和组件可靠性中，不管这些模型是否准确，只要有相当多的不确定性存在，在计算软件可靠性时，就会存在较多的不确定性。若采用传统方法估算软件的可靠性，忽略了属于参数不确定性存在的差异，那么结果可能不准确。提出了一种新的基于体系结构的不确定性的分析方法，该方法适合大型复杂的基于组件的应用程序及整个软件生命周期。

关键词：可靠性；软件体系结构；分层软件系统；构件
中图分类号：TP301 **文献标识码**：A

Assessing uncertainty in reliability of component based software systems

SONG Shao Yun
(Modern Information Technology Center, Yuxi Normal University, Yuxi 653100, China)

Abstract: Many architecture based software reliability models were proposed in the past. Regardless of the accuracy of these models, if a considerable uncertainty exists in the estimates of the operational profile and components reliabilities then a significant uncertainty exists in calculated software reliability. In this paper we propose a methodology for uncertainty analysis of architecture based software reliability models suitable for large complex component based applications and applicable throughout the software life cycle.

Key words: reliability; software architecture; hierarchical software system; component

一些分析模型已提出解决软件可靠性量化的问题，但这些模型是把组件作为一个黑箱进行处理，主要集中于测试阶段的可靠性增长，只考虑组件与外部环境的相互作用，而没有考虑组件内部的结构[1]。因此，黑箱操作模式并不适合以组件为基础的大型系统。需要用一种白箱的方式，这种方式要求考虑软件体系结构中的组件相关输出信息。已有的基于模型的软件可靠性预测存在两个重要问题，第一个问题涉及模型是否恰当，如果模型的假设在实践中未必成立，则该模型有可能不恰当。第二个问题涉及参数值的准确性[2]。

传统上，最常见的软件可靠性不确定性分析方法是敏感性研究。因此，黑箱中上下文的操作剖面可靠性的敏感估计属于可靠性增长模型。通过软件体系结构模型获得的可靠性估计的敏感性研究已有了一些文献，如文献[3]。

以前用熵分析不确定性，根据信息论可知，熵能够从操作剖面量化整个系统的可靠性不确定性和组件的不确定性。虽然取得的成效对基于组件的系统核查和验证是有用的，但这种方法并不提供软件可靠性估计。

1 不确定性分析方法

为了估计基于结构的系统模型可靠性，需要知道软件体系结构（组件交互的结构）、软件操作剖面的使用（组件交互的相对频率）以及软件故障行为（组件可信度或失败概率）。这种基于体系结构的软件可靠性模型所提供的可靠性估计接近实际测量的可靠性。本文提出了一种不确定性分析方法，并着重于软件可靠性中不确定性的评估，属于模型参数的不确定性。

1.1 软件体系结构

软件行为涉及不同组件交互的方式，将其定义为

软件体系结构。使用基于状态的办法来建立基于体系结构的软件可靠性模型。该方法使用了控制流图，用于表示软件的体系结构。组件表示活动状态，弧线表示组件的状态转移。基于这样的假设：即组件间的控制转移具有马尔可夫特性， 模型化的结构具有离散时间马尔可夫链（DTMC）性质，转移概率矩阵为：$P=[P_{ij}]$，P_{ij}表示从组件 i 到组件 j 的状态转移概率。

(1)预测方法：用于软件开发的早期阶段。基于对历史数据的估计，从有关软件体系结构的同类产品或对高层次信息中得到规范的设计文件。

(2)通知方法：在软件开发的后期阶段，如果测试数据或现场数据可用，使用覆盖测试工具可获得组件操作剖面及软件执行的一套路径，用频率数值作为其转移概率。

软件体系结构中的动态信息，依赖于软件的使用情况，那就是操作剖面。一般来说，操作剖面的可靠性估计比较困难，因为它需要预先知道软件的使用域、应用程序的先验知识和系统的运行环境。将过程控制应用于软件各组件，并由一序列复杂事件激活，其频率很难预先估计[4]。

1.2 组件失效行为

首先考虑组件失效行为，即每个组件的可靠性估计。组件 i 的可靠性是其完成自身功能正确性的概率 Ri。软件中组件可靠性估计依赖于各种因素。

软件可靠性增长模型可以应用到每一个软件开发中探测组件的失败数据。不过，由于缺乏失效数据，所以不能够利用软件可靠性增长模型。另一种可能性是从非失败执行来考虑估计组件的可靠性。可靠性估计要建立一个合理的统计，可以使用故障注入技术来估计组件的可靠性。

1.3 软件体系结构与失败行为的结合

所提出的不确定性分析方法，可应用于任何基于体系结构的软件可靠性模型中。结合软件体系结构与失败行为，采用复合方法进行估计，把分别代表正确输出和失败的状态 C 和 F 增加到 DTMC 中。转移概率矩阵 $\boldsymbol{P}$ 修改 $\bar{\boldsymbol{P}}$ 为：原来两个组件 i 和 j 之间的转移概率 p_{ij} 修改为 $R_i p_{ij}$，代表组件 i 产生的正确结果以及控制权转移到组件 j 的概率。组件 i 的失败概率 $(1-R_i)$ 是状态 F 到状态 C 建立的一条有向边。$\boldsymbol{Q}$ 是矩阵 $\bar{\boldsymbol{P}}$ 删去行与列相对应的吸收状态 C 和 F。$(1,n)$ 阶矩阵 $\boldsymbol{Q}_k$ 表示状态从 1 经过 k 转移到 n 的概率，从初始状态 1 到终止状态 n,大量转移时 k 可以从 0 到无穷大。可以证明 $\boldsymbol{S}=\sum_{k=0}^{\infty}\boldsymbol{Q}^k=(1-\boldsymbol{Q})^{-1}$ 即 $(1, n)$ 个元素的矩阵 $\boldsymbol{S}$ 表示从状态 1 到达状态 n 的概率。这意味着整个系统的可靠性，由 $R=S_1nRn$ 给出。

1.4 不确定性分析

在基于体系结构的软件可靠性中，有两个不确定性的因素：软件使用方法（即运行剖面）和组件失效行为（即组件的可靠性）。根据 1.3 节中的模型得到关于系统的转移概率 P_{ij} 和组件可靠性 R_i 的可靠性 R 函数的表达式：$R=S_{1n}R_n$。不论该模型的可靠性精确与否，如果在操作剖面和组件可靠性中存在不确定性，那么在计算系统可靠性时也存在不确定性。因此，计算软件可靠性的传统方法是不恰当的，因为该方法是通过插入参数到模型中进行估计的。值得强调的是矩方法和蒙特卡罗模拟法可用于评估在软件测试中没有发现任何故障情况下的软件可靠性的不确定性[5]。

2 矩方法

矩方法是一种近似的方法，允许从组件矩的可信度产生系统瞬间的可靠性矩，即在软件可靠性中不确定性的量化，属于组件可靠性的不确定性。本文导出的表达式对独立随机变量和操作剖面的不确定性是有效的。对矩方法来说，首先通过函数 $R=f(R_1,R_2,\cdots,R_n)$ 获得系统可靠性 R 和组件的可靠性 R_1，R_2，$\cdots R_n$ 之间的关系。如果把每个组件可靠性作为一个随机变量，那么系统的可靠性也是一个随机变量。$E[Ri]$ 是第 i 个组件可靠性的均值，$\mu k[R_i]=e[(R_i-E[R_i])k$ 表示它的中心矩。矩方法使基于下列基础上的期望值 $E[R]$ 的估计和系统可靠性的第 k 个中心矩 $\mu k[R]$：（1）系统结构知识和操作剖面 $R=f(R_1,R_2,\cdots,R_n)$；（2）来自组件失败数据 $E[Ri]$ 和 $\mu k[Ri]$ 的估计。

使用矩方法产生的系统可靠性矩是基于扩展函数 $R=f(R_1,R_2,\cdots,R_n)$ 的一个多变量泰勒级数，其中每一个组件可靠性的期望值为 $E[R_i]$。手工推导出系统可靠性的表达式和相应的泰勒系数是较复杂，而且只适应于小系统。因此，以矩方法为基础，用 Mathematica 软件推导出表达式 $R=f(R_1,R_2,\cdots,R_n)$，并用泰勒级数对其进行展开来获得系统可靠性。

由于中高阶泰勒级数展开的遗漏，所以矩方法是一种近似的，而不是一种精确的方法。第一阶泰勒级数展开由下面公式给出：

$$R\sim a_0+\sum_{i=1}^{n}a_i(R_i-E[R_i]) \tag{1}$$

其中：

$$a_0=f(E[R_1],E[R_2],\cdots,E[R_n]) \tag{2}$$

$$a_i=\frac{\partial R}{\partial R}\Big|_{R_i=E[R_i]}\quad i=1,2,\cdots,n \tag{3}$$

系统可靠性的均值和方差由下式给出：

$$E[R]\sim a_0 \tag{4}$$

$$Var[R]=\mu_2[R]\sim\sum_{i=1}^{n}a_i^2 Var[R_i] \tag{5}$$

明显从第一阶泰勒级数展开产生系统可靠性的均值（4）和方差（5），需要知道前两个组件可靠性（$E[Ri]$ 和 $Var[Ri]$）的中心矩。方差对软件可靠性的评估很重

要，因为它是可靠性估计信任度的衡量。因此，方差的值越小对应的信任度越高。

$E[R]$和 Var $[R]$ 的精度可以由高阶泰勒级数展开进行改善。下面推导出了二阶泰勒级数的展开式。

$$R \sim a_0 + \sum_{i=1}^{n} a_i (R_i - E[R_i]) + \frac{1}{2}\sum_{i=1}^{n} a_{ii}(R_i - E[R_i])^2 + \sum_{i=1}^{n}\sum_{j=1}^{i-1} a_{ij}(R_i - E[R_i])(R_j - E[R_j]) \quad (6)$$

其中 a_0 和 a_i 由方程(2)和(3)给出，并且：

$$a_{ii} = \frac{\partial^2 R}{\partial R_i^2}\bigg|_{R_i = E[R_i]} \quad i=1,2,\ldots,n \quad (7)$$

$$b_{ij} = \frac{\partial^2 R}{\partial R_i \partial R_j}\bigg|_{R_k = E[R_k]} \quad k=1,2,\ldots,n \quad (8)$$

系统可靠性的均值和方差，由第二阶泰勒近似给出：

$$E[R] \sim a_0 + \frac{1}{2}\sum_{i=1}^{n} a_{ii} Var[R_i] \quad (9)$$

$$Var[R] \sim \sum_{i=1}^{n} a_i^2 Var[R_i] + \sum_{i=1}^{n}\sum_{j=1}^{i-1} a_{ij}^2 Var[R_i]Var[R_j] + \frac{1}{4}\sum_{i=1}^{n} a_{ii}^2 E[(R_i - E[R_i])^4] + \sum_{i=1}^{n} a_i a_{ii} E[(R_i - E[R_i])^3] - \frac{1}{4}\sum_{i=1}^{n} a_{ii}^2 (Var[R_i])^2 \quad (10)$$

3 蒙特卡罗模拟

蒙特卡罗模拟要考虑双方不确定资源所产生的不确定性（即操作剖面和组件可靠性）。当模型参数能很好地表示概率分布时，则是功能强大的系统可靠性估算方法。直接采样的蒙特卡罗方法构成了随机变量的重复产生，它结合了所得系统方程参数可靠性的分布。从本质上讲，这等于对相同系统进行了许多的实验或运行了很多次的测试。

本文使用基于体系结构的软件可靠性模型，得出了可靠性表达式 $R=f(p_{ij},R_i)$。然后，对组件可靠性 R_i 和转移概率 p_{ij} 给出了其概率分布函数，这些分布函数是基于一定的理论假设或观测到的数据。假设组件的可靠性是随机变量，且服从 Beta 分布，分布函数由下式给出。

$$f(R_i) = \frac{\Gamma(a_i + b_i)}{\Gamma(a_i)\Gamma(b_i)} R_i^{a_i - 1}(1 - R_i)^{b_i - 1} \quad (11)$$

其中，$0 \leqslant R_i \leqslant 1$。假设在转移概率矩阵中的行是独立的，并且服从 Dirichlet 分布。因此，转移概率矩阵中第 i 行的密度函数 PDF 具有下列形式：

$$f(p_{i1}, p_{i2}, \ldots, p_{in}) = \frac{\Gamma(a_{i1} + a_{i2} + \ldots + a_{in})}{\Gamma(a_{i1})\Gamma(a_{i2})\ldots\Gamma(a_{in})} \prod_{j=1}^{n} p_{ij}^{a_{ij} - 1} \quad (12)$$

其中，$P_{ij} \geqslant 0$ 且 $\sum_{j=1}^{n} p_{ij} = 1$

标准 Dirichlet 分布定义为 $(Y_1, Y_2, \cdots, Y_n)$ 的分布，其中 $Y_k = Z_k / \sum_{j=1}^{n} Z_j$，且 $Z_j, j=1,2,\cdots,n$ 相互独立，形状参数随机变量 a_j 服从标准 Gamma 分布。服从 Dirichlet 分布的不同参数的选择，可以采取各种各样的形状。

因此，即使在使用 Dirichlet 分布的情况下，也没有隐含的理论，由于它的形状不同，所以只能作为一种近似。

4 不确定性分析案例

假设某软件系统使用了 10 个组件，且符合 DTMC，图 1 给出了其体系结构。转移概率 P_{ij} 和 R_i 由表 1 给出。

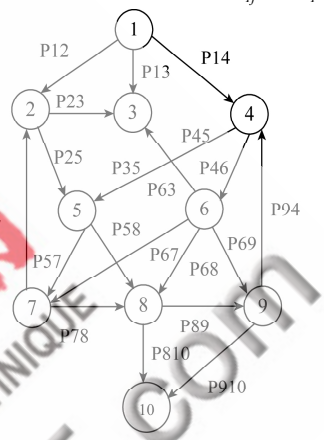

图 1 软件体系结构

表1 转移概率和组件可靠性的均值

p_{12}=0.60	p_{13}=0.20	p_{14}=0.20		R_1=0.999
p_{23}=0.70	p_{25}=0.30			R_2=0.980
p_{35}=1.00				R_3=0.990
p_{45}=0.40	p_{46}=0.60			R_4=0.970
p_{57}=0.40	p_{58}=0.60			R_5=0.950
p_{63}=0.30	p_{67}=0.30	p_{68}=0.10	p_{69}=0.30	R_6=0.995
p_{72}=0.50	p_{78}=0.50			R_7=0.985
p_{89}=0.25	$p_{8,10}$=0.75			R_8=0.950
p_{94}=0.10	$p_{9,10}$=0.90			R_9=0.975
				R_{10}=0.985

首先，研究软件可靠性的不确定性属于组件可靠性的不确定性。为此，根据表 1 给出的组件参数，用矩方法和蒙特卡罗模拟方法可得到组件可靠性均值 $E[R_i]$为：

$Var[R_1] = Var[R_3] = Var[R_6] = 0.0001$，

方差为：

$Var[R_2] = Var[R_4] = Var[R_5] = Var[R_7]$

$= Var[R_8] = Var[R_9] = Var[R_{10}] = 0.001$

从表 2 可以看出，用矩方法估计的系统可靠性均值 E[R]和方差 Var[R]接近蒙特卡罗模拟[6]。

表2 系统可靠性的均值和方差

	第一阶	第二阶	蒙特卡罗模拟
均值	0.8299	0.8319	0.8304
方差	0.0036	0.0035	0.0035

5 不确定性分析方法的比较

在软件可靠性中不确定性分析方法的选择依赖于某一特定应用标准，如数据需求、可靠性度量推导和解决问题的准确性。在表3中总结了不同方法的基本特征，基于熵理论和马尔可夫链的不确定性分析方法使人们在操作剖面和单值可靠性的不确定性方面具有一定的特点，熵作为一种研究不确定性的方法，与矩方法和蒙特卡罗模拟是相辅相成的。

表3 不确定性分析方法的比较

方法	需要的数据	可靠性度量	解决方法的精度
熵	点估计	N/A	精确的分析方法
矩方法	组件可靠性的矩	矩	近似方法：分析的问题由高阶高阶泰勒级数增加准确性
蒙特卡罗模拟	(1)移概率和组件可靠性的分布函数。 (2)随即数的产生	(1)分布 (2)矩 (3)百分比 (4)参数对方差的影响	近似方法：模拟 (1)增加样本大小来增加精确度。 (2)样本错误包括长尾分布情况

矩方法相比蒙特卡罗模拟其优点是较少数据的要求。这种方法只需要组件可靠性的矩，可以很容易地直接从试验数据进行计算。此外，矩方法是一种分析方法，因此不需要随机数值的产生，只要进行高阶泰勒级数的展开，矩方法的精确度就可以增加。蒙特卡罗模拟原则上可以简单地增加模拟值来增加准确性。此外，矩方法的准确性不容易量化，如果需要精确的不确定性分析，应使用蒙特卡罗方法。同时，蒙特卡罗模拟提供了一套可靠性度量措施，如矩、分布函数和系统可靠性的百分比，以及灵敏度参数的差异范围。另一个蒙特卡罗模拟的好处在于能够量化操作剖面和组件可靠性的不确定性。注意矩方法可应用于操作剖面中的不确定性，不过，由于很复杂的原因，将很难导出这些表达式。

本文提出一种新的软件可靠性不确定性分析的方法，可应用于整个软件生命周期。按照这一方法，用矩方法和蒙特卡罗模拟分析如何将参数的不确定性纳入软件可靠性的估计计算。由于基于结构的方法能洞察到软件的执行动态行为，研究了不同参数对软件可靠性不确定性的影响。显然，不确定性分析提供了比传统点估计的软件可靠性度量的更丰富措施。这些措施可用于指导测试工作的分配，使软件质量满足不同操作使用的量化要求，并为基于构件软件系统的可靠性提供验证。因此，软件可靠性的不确定性分析方法不仅是重要的而且是必要的，尤其适用于对早期生命周期以及跟踪软件进化的预测。

参考文献

[1] ADAMS T. Total variance approach to software reliability estimation [J]. Software Engineering, 1996,22(9):687-688.

[2] CHEN M, MATHUR A, REGO V.A case study to investigate sensitivity of reliability estimates to errors in operational profile [J]. Software Reliability Engineering. 1994, (5): 276-281.

[3] CHEUNG R. A user-oriented software reliability model[J]. Software Engineering, 1980,6(2):118-125.

[4] FARR W. Software reliability modeling survey. Handbook of Software Reliability Engineering, M. R. Lyu (Ed.), mcGraw-Hill, 1996: 71-117.

[5] GOKHALE S, TRIVEDI K. Reliability prediction and sensitivity analysis based on software architecture, 13th Int'l Symp. Software Reliability Engineering, 2002:64-75.

[6] GOSEVA K, TRIVEDI K.Architecture based approach to reliability assessment of software system[J]. Performance Evaluation, 2001, 45(2-3): 179-204.

(收稿日期：2009-03-19)