

文章编号:1001-9081(2008)06-1630-02

构件软件的可靠性估算模型

周娜琴¹,张友生^{1,2}

(1. 湖南师范大学 计算机应用技术研究中心, 长沙 410081; 2. 湖南大学 工商管理学院, 长沙 410082)

(doudou187230@163.com)

摘要:把基于构件的软件看作是一个 Markov 过程,为弥补以往忽视连接件作用的情况,针对过程中不同状态类型,构建出基于不同状态的构件和连接件使用频率计算模型。在此基础上,提出了整个基于构件的软件可靠性计算方法,并将其实例化。与传统的方法相比,该方法不仅提供了一种更精确分析软件可靠性的方法,而且拓宽了模型的应用范围。

关键词:基于构件的软件;可靠性;软件可靠性

中图分类号: TP311 **文献标志码:** A

Reliability model of component-based software

ZHOU Na-qin¹, ZHANG You-sheng^{1,2}

(1. Research Center of Computer Application Technologies, Hunan Normal University, Changsha Hunan 410081, China;

2. College of Business Administration, Hunan University, Changsha Hunan 410082, China)

Abstract: Component-based software was considered as Markov process. To attach more attention to the reliability of links and become fit to different Markov process states, different state components and link use frequency computing models were proposed. Then, the calculation method for reliability of component-based software was presented, and an example was also given. Compared with traditional methods, the proposed method not only provides a more accurate analysis method for the reliability of software, but also widened the application of the model.

Key words: component-based software; reliability; software reliability

0 引言

目前,已经提出了许多估算软件可靠性的模型,大多数模型是围绕软件测试阶段的可靠性增长模型(黑箱模型)。黑箱模型把软件作为一个整体,而不考虑模型的内部结构^[1]。然而,随着面向对象的系统设计和基于 Web 的软件开发的广泛应用,基于构件的软件开发正在不断增长。软构件可以是第三方构件、自主研发的构件或合约研发的构件,整个应用开发是在一个异构环境下。因此,应用现有的软件可靠性增长模型估算基于构件的软件可靠性不再适应^[2]。

本文提出了一种新的基于构件的可靠性评估方法。该方法根据转移概率及 Markov 过程不同状态类型分别推导出构件和连接件的使用频率。在此基础上,描述了系统可靠性计算方法,并给出了相应的实例化研究。

1 Markov 过程

Markov 过程是一类重要的随机过程,它的原始模型是马尔可夫链。粗略地说,马尔可夫性可以用下述直观语言来刻画:在已知系统目前的状态(现在)的条件下,它未来的演变(将来)不依赖于它以往的演变(过去)。换言之,当一个系统处于 K 个有限的状态 $X_n = 1, 2, \dots, K, X_n = i$ 的概率为 $p(X_n = i)$,称状态概率;从 $X_n = i$ 到状态 $X_n = j$ 的概率为 $p_{i,j}$,称为转移概率;若 X_{n+1} 的取值只取决于 X_n 的值和 $p_{i,j}$,而与 $X_{n-1}, X_{n-2}, \dots, X_1$ 的取值无关,具有这种特征的随机过程称为马尔可夫过程。

马尔可夫过程可有二种不同类型的状态构件,分别为非

吸收态和吸收态。如果从状态 i 到达状态 j 后,永久停留在 j ,则称状态 j 为吸收态,否则为非吸收态。

由转移概率 $p_{i,j}$ 构成的矩阵 P ,称为转移概率矩阵。一个由有 r 个吸收态和 $n-r$ 个非吸收态的 Markov 链,经过适当排列,其转移概率矩阵 P 总可以表示为如下形式:

$$P = \begin{matrix} r \text{ 个吸收态} & n-r \text{ 个非吸收态} \\ r \text{ 个吸收态} & \begin{bmatrix} 1 & 0 \\ D & Q \end{bmatrix} \\ n-r \text{ 个非吸收态} & \end{matrix} \quad (1)$$

2 基于构件的软件可靠性模型

在基于构件的软件开发中,软件系统由构件和连接件组成。整个基于构件的软件可以看成是一个 Markov 过程。本文通过估算构件和连接件在整个系统中的使用次数,估算整个系统的可靠性。

例如,当某个系统由 N 个构件组成,构件 C_i 的可靠性为 R_{C_i} ,使用次数为 θ_i ,连接件 $L_{i,j}$ 的可靠性为 $R_{L_{i,j}}$,使用次数为 $\sigma_{i,j}$,系统的可靠性为 R_s :

$$R_s = \prod_{i=1}^N pow(R_{C_i}, \theta_i) \times \prod_{j=1}^N \prod_{k=1}^N pow(R_{L_{j,k}}, \sigma_{j,k}) \quad (2)$$

定义 1 设 C_1, C_2, \dots, C_N 是软件系统 S 中 N 个不同构件,其中 $L_{i,j}$ 表示第 C_i 个构件与第 C_j 个构件之间的连接件,把这 N 个构件看作由 N 个不同状态(用 X_i 表示, i 从 1 到 N) 构成的 Markov 过程。假设整个系统有 s 个初态 $X_i (1 \leq i \leq s)$ 和 r 个吸收状态 $X_j (N-r+1 \leq j \leq N)$,令 P_i 表示访问初态的概率。 $N_{i,j}$ 表示到达吸收态之前,从状态 X_i 出发访问 X_j 的次数。 $N_{i,j}$ 的期望 $E(N_{i,j})$ (也可用 $u_{i,j}$ 表示)。 $N_{i,k,j}$ 表示到达吸收态之前,

收稿日期:2007-12-13;修回日期:2008-02-01。 基金项目:国家自然科学基金资助项目(70572058)。

作者简介:周娜琴(1982-),女,湖南邵阳人,硕士研究生,主要研究方向:软件体系结构、软件可靠性、网络软件;张友生(1969-),男,湖南邵阳人,副教授,博士,主要研究方向:软件工程、软件体系结构。

从状态 X_i 出发到达 X_j 的过程中访问连接件 $L_{k,j}$ 的次数 $\circ N_{i,k,j}$ 的期望 $E(N_{i,k,j})$ 。 $\eta_{i,j}$ 表示从状态 X_i 出发访问吸收态 X_j 的次数。 $\eta_{i,k,j}$ 表示从状态 X_i 出发访问吸收态 X_j 的过程中访问连接件 $L_{k,j}$ 的次数。 α_j 表示访问状态 $X_j (s+1 \leq j \leq N-r)$ 的次数, β_j 表示访问状态 $X_j (N-r+1 \leq j \leq N)$ 的次数, $\alpha_{k,j}$ 表示访问连接件 $L_{k,j}$ 的次数 $(1 \leq k \leq N-r, N-r+1 \leq j \leq N)$ 的次数。

$$u_{i,j} = E(N_{i,j}) = E(N_{i,j}/X_1) \times P(X_1/X_i) + \dots + E(N_{i,j}/X_{N-r}) \times P(X_{N-r}/X_i) + E(N_{i,j}/X_{N-r+1}) \times P(X_{N-r+1}/X_i) + \dots + E(N_{i,j}/X_N) \times P(X_N/X_i) + \sum_{k=1}^{N-r} E(N_{i,j}/X_k) \times P(X_k/X_i) + \sum_{k=N-r+1}^N E(N_{i,j}/X_k) \times P(X_k/X_i) \quad (3)$$

当 $N-r+1 \leq k \leq N, X_k$ 表示吸收态, 此时 $\sum_{k=N-r+1}^N E(N_{i,j}/X_k) \times P(X_k/X_i) = 0$,

$$u_{i,j} = \delta_{ij} + \sum_{k=1}^{N-r} u_{i,k} \times P_{i,k}; \text{ 当 } i=j \text{ 时 } \delta_{ij} = 1, \text{ 否则 } \delta_{ij} = 0 \quad (4)$$

令由 $u_{i,j}$ 构成的矩阵为 $M, M = I + Q \times M$, 那么 $M = (I - Q)^{-1}$ 。

$$\alpha_j = \sum_{i=1}^s P_i \times u_{i,j} \quad (5)$$

$$E(N_{i,k,j}) = E(N_{i,k}) \times P(X_j/X_k) = u_{i,k} \times P_{k,j} \quad (6)$$

$$\alpha_{k,j} = \sum_{i=1}^s P_i \times E(N_{i,k,j}) = \sum_{i=1}^s P_i \times u_{i,k} \times P_{k,j} \quad (7)$$

$$\eta_{i,j} = (u_{i,1} \times P_{1,j} + u_{i,2} \times P_{2,j} + \dots + u_{i,N-r} \times P_{(N-r),j}) = \sum_{k=1}^{N-r} u_{i,k} \times P_{k,j} = ((I - Q)_{(N-r) \times (N-r)}^{-1} \times D_{(N-r) \times r})_{ij} \quad (8)$$

$$\beta_j = \sum_{i=1}^s P_i \times \eta_{i,j} = \sum_{i=1}^s P_i \times ((I - Q)_{(N-r) \times (N-r)}^{-1} \times D_{(N-r) \times r})_{ij} \quad (9)$$

$$\eta_{i,k,j} = u_{i,k} \times P_{k,j} \quad (10)$$

$$\beta_{k,j} = \sum_{i=1}^s P_i \times \eta_{i,k,j} = \sum_{i=1}^s P_i \times u_{i,k} \times P_{k,j} \quad (11)$$

整个软件的可靠性 R_s 为:

$$R_s = \prod_{i=1}^s \text{pow}(R_{C_i}, P_i) \times \prod_{j=s+1}^{N-r} \text{pow}(R_{C_j}, \alpha_j) \times \prod_{k=1}^{N-r} \prod_{l=1}^{N-r} \text{pow}(R_{L_{k,l}}, \alpha_{k,l}) \times \prod_{t=N-r+1}^N \text{pow}(R_{C_t}, \beta_t) \times \prod_{g=1}^{N-r} \prod_{h=N-r+1}^N \text{pow}(R_{L_{g,h}}, \beta_{g,h}) \quad (12)$$

当 $s=1, r=1$ 时, 系统为单输入/单输出系统, 此时整个系统的可靠性 R_s 可以进一步简化。

$$R_s = R_1 \times R_N \times \prod_{i=2}^{N-1} \text{pow}(R_{C_i}, u_{1,i}) \times \prod_{j=1}^{N-1} \prod_{k=1}^{N-1} \text{pow}(R_{L_{j,k}}, u_{1,j} \times P_{j,k}) \times \prod_{l=1}^{N-1} \text{pow}(R_{L_{l,N}}, u_{1,l} \times P_{l,N}) \quad (13)$$

3 实例研究

不失一般性, 本节应用文献[3]的实例作为研究实例, 该实例由 10 个构件构成, 构件与构件之间的关系如图 1 所示, 并在此基础上增加连接件和给出相应的可靠性。

其中系统中各转移概率分别为: $P_{1,2} = 0.6, P_{1,3} = 0.2,$

$P_{1,4} = 0.2, P_{2,3} = 0.7, P_{2,5} = 0.3, P_{3,5} = 1.0, P_{4,5} = 0.4, P_{4,6} = 0.6, P_{5,7} = 0.4, P_{5,8} = 0.6, P_{6,3} = 0.3, P_{6,7} = 0.3, P_{6,8} = 0.1, P_{6,9} = 0.3, P_{7,2} = 0.5, P_{7,9} = 0.5, P_{8,4} = 0.25, P_{8,10} = 0.75, P_{9,8} = 0.1, P_{9,10} = 0.9$ 。 构件的可靠性分别为: $R_{c1} = 0.999, R_{c2} = 0.980, R_{c3} = 0.990, R_{c4} = 0.970, R_{c5} = 0.950, R_{c6} = 0.995, R_{c7} = 0.985, R_{c8} = 0.950, R_{c9} = 0.975, R_{c10} = 0.985$ 。 连接件的可靠性分别为: $R_{L1,2} = 0.99, R_{L1,3} = 0.98, R_{L1,4} = 0.99, R_{L2,3} = 0.97, R_{L2,5} = 0.98, R_{L3,5} = 0.96, R_{L4,5} = 0.99, R_{L4,6} = 0.98, R_{L5,7} = 0.98, R_{L5,8} = 0.95, R_{L6,3} = 0.99, R_{L6,7} = 0.97, R_{L6,8} = 0.98, R_{L6,9} = 0.98, R_{L7,2} = 0.97, R_{L7,9} = 0.97, R_{L8,4} = 0.99, R_{L8,10} = 0.96, R_{L9,8} = 0.98, R_{L9,10} = 0.99$ 。

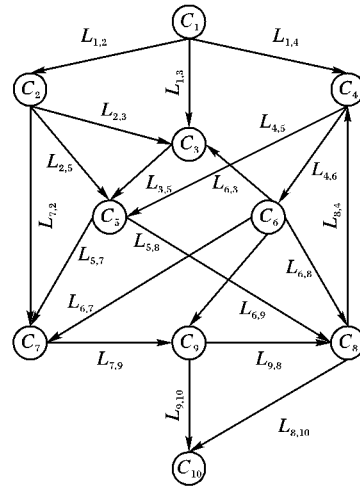


图 1 一个基于构件的软件系统

通过第 2 章的计算公式, 可以计算出:

$$u_{1,1} = 1.0000, u_{1,2} = 0.9077, u_{1,3} = 0.9107, u_{1,4} = 0.4184, u_{1,5} = 1.3504, u_{1,6} = 0.2510, u_{1,7} = 0.6155, u_{1,8} = 0.8737, u_{1,9} = 0.3831$$

$$\alpha_{1,2} = 0.6, \alpha_{1,3} = 0.2, \alpha_{1,4} = 0.2, \alpha_{2,3} = 0.6354, \alpha_{2,5} = 0.2723, \alpha_{3,5} = 0.9107, \alpha_{4,5} = 0.1674, \alpha_{4,6} = 0.2510, \alpha_{5,7} = 0.5402, \alpha_{5,8} = 0.8102, \alpha_{6,3} = 0.0753, \alpha_{6,7} = 0.0753, \alpha_{6,8} = 0.0251, \alpha_{6,9} = 0.0753, \alpha_{7,2} = 0.3078, \alpha_{7,9} = 0.3078, \alpha_{8,4} = 0.2184, \alpha_{9,8} = 0.0383, \alpha_{8,10} = 0.6553, \alpha_{9,10} = 0.3448$$

系统可靠性 R_s 为: $R_s = 0.6890$ 。

把连接件当作构件用文献[3]提出的模型估算, 所估算的可靠性与本文的方法估算结果相同, 但是用该模型估算的过程中, 所用到的状态矩阵相比本文成数量级增加。因此本文提出的方法更优于基于构件的软件可靠性估算。

4 结语

本文提出的可靠性计算模型可以在软件开发的各个阶段对软件进行可靠性分析。与其他软件可靠性分析和评估方法相比, 本文的方法弥补了其他方法对连接件考虑的不足, 更精确地估算基于构件的软件可靠性, 拓宽了模型应用范围。

参考文献:

[1] GOSEVA-POPSTOJANOVA K, MATHUR A P, TRIVEDI K S. Comparison of architecture-based software reliability models [C]// 12th International Symposium on Software Reliability Engineering (ISSRE'01). Washington: IEEE Computer Society, 2001: 22-31.

图 4 是使用 RBB 算法对第二组基准例题 C2(C21 - C23) 得到的输出排样图,材料利用率都是 100%。这里卷板是水平放置的,其宽度为 40,占用长度为 15。输出排样图中,由于有些毛坯太小,其编号没能写在毛坯方框中。

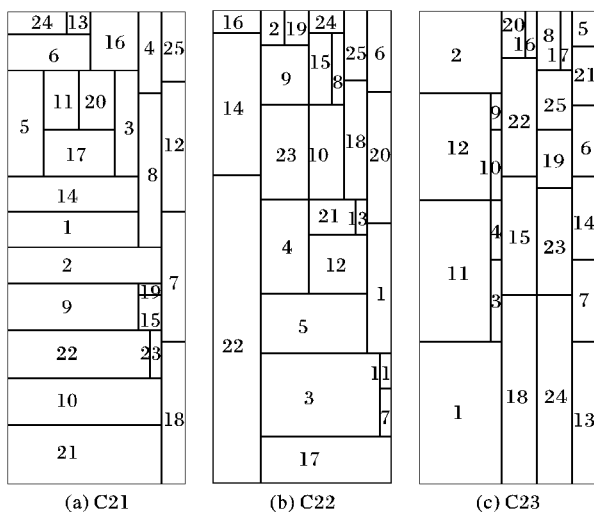


图 4 RBB 算法对基准例题 C2(C21 - C23) 的排样图

表 2 所示为计算结果,同时列出文献中报道的其他算法的计算结果。和最优解的差距按下式计算: $(H - H_0)/H_0$, H 为由某种算法所得卷板高度。表中最后一列为各种算法七组例题的总计算时间(没有分组记录)。HR、HRBB、SPGAL 和 SAHR 分别为文献[5 - 8]中的算法;GABLF 和 SABLFF 为文献[9]中报道的两种算法;RBB 代表本文所述算法。

表 1 测试题目的特性组别

C1 (C11 - C13)	16 或 17	20	20
C2 (C21 - C23)	25	40	15
C3 (C31 - C33)	28 或 29	60	30
C4 (C41 - C43)	49	60	60
C5 (C51 - C53)	73	60	90
C6 (C61 - C63)	97	80	120
C7 (C71 - C73)	196 或 197	160	240

表 2 和最优解的差距

算法	C1/%	C2/%	C3/%	C4/%	C5/%	C6/%	C7/%	平均/%	时间/s
HR	8.3	4.5	6.7	2.2	1.9	2.5	1.8	4.0	5.6
HRBB	1.7	0.0	1.1	1.1	1.1	1.1	1.1	1.0	49.4
SPGAL	1.7	0.9	2.2	1.4	0.0	0.7	0.5	1.0	139.0
SAHR	5.0	4.5	2.2	2.2	1.9	2.5	3.2	3.1	344.6
GABLF	4.0	7.0	5.0	3.0	4.0	4.0	5.0	4.6	7869.6
SABLFF	4.0	6.0	5.0	3.0	3.0	3.0	4.0	4.0	40441.9
RBB	1.7	0.0	0.0	1.1	0.4	0.8	1.8	0.8	12.4

从表 2 的数据可以看出,本文提出的 RBB 算法给出的解的平均质量最高,即最节省板料,和最优解的平均差距只有

0.8%,它与机器性能无直接关系。GABLF 和 SABLFF 使用奔腾 Pro 200 MHz、65 MB 内存的计算机;SPGAL 使用 2 GHz 奔腾 CPU 的计算机;HR 使用 2.4 GHz 的 DELL GX260 计算机;SAHR 使用 3.0 GHz 的 DELL GX270 计算机;HRBB 使用主频 2.8 GHz,内存 512 MB 的计算机。RBB 的计算时间明显短于 SPGAL、SAHR 和 HRBB,因为它们使用计算能力相同或接近的计算机;RBB 的计算时间比 HR 长,但解的质量比 HR 高得多;由于所用计算机的性能差异较大,难以将 RBB 在计算时间上和 GABLF 及 SABLFF 两种算法相比,但 RBB 解的质量远比它们的高。

4 结语

本文算法设计简单,对于所采用的一组 21 道实验例题,解的质量比其他算法都高;计算时间比大多数算法要短。将算法加以扩展,考虑圆形等其他形状毛坯的特点,实现圆形等其他形状毛坯的分层递归优化排样;考虑模块之间的互连线长度,解决 VLSI 模块布图规划问题,可以作为本文的后续研究内容。

参考文献:

- [1] LODI A, MARTELLO S, MONACI M. Two-dimensional packing problems: A survey [J]. European Journal of Operational Research, 2002, 141: 241 - 252.
- [2] HOPPER E, TURTON B C H. A review of the application of meta-heuristic algorithms to 2D strip packing problems [J]. Artificial Intelligence Review, 2001, 16: 257 - 300.
- [3] HIFI M. Exact algorithms for the guillotine strip cutting/packing problem[J]. Computers & Operations Research, 1998, 25: 925 - 940.
- [4] WU Y, HUANG W, LAU S, *et al.* An effective quasi-human based heuristic for solving the rectangle packing problem [J]. European Journal of Operational Research, 2002, 141: 341 - 358.
- [5] ZHANG D, KANG Y, DENG A. A new heuristic recursive algorithm for the strip rectangular packing problem [J]. Computers & Operations Research, 2006, 33: 2209 - 2217.
- [6] CUI Y, YANG Y, CHENG X, *et al.* A recursive branch-and-bound algorithm for the rectangular guillotine strip packing problem [J]. Computers & Operations Research, 2006. Doi: 10.1016/j.cor.2006.08.011.
- [7] BORTFELDT A. A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces [J]. European Journal of Operational Research, 2006, 172: 814 - 837.
- [8] ZHANG D, LIU Y, CHEN S, XIE X. A meta-heuristic algorithm for the strip rectangular packing problem[J]. Lecture Notes in Computer Science, 2005, 3612: 1235 - 1241.
- [9] HOPPER E, TURTON B C H. An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem [J]. European Journal of Operational Research, 2001, 128: 34 - 57.

(上接第 1631 页)

- [2] GOKHALE S S, TRIVEDI K S. Reliability prediction and sensitivity analysis based on software architecture [C]// 13th International Symposium on Software Reliability Engineering (ISSRE'02). Washington: IEEE Computer Society, 2002: 64 - 75.
- [3] LO J-H, KUO S-Y, LYU M R, *et al.* Optimal resource allocation and reliability analysis for component-based software application [C]// Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment. Washington: IEEE Computer Society, 2002: 7 - 12.
- [4] GOKHALE S S. Software reliability analysis with component-level

- fault tolerance [C]// Proceedings of IEEE Reliability and Maintainability Symposium. New York: IEEE Press, 2005: 610 - 614.
- [5] WANG WEN-LI, WU YE, CHEN MEI-HWA. Architecture-based software reliability analysis [C]// Proceedings of Pacific Rim International Symposium on Dependable Computing. Hong Kong: [s. n.], 1999, 12: 143 - 150.
- [6] CHEUNG R C. A user - oriented software reliability model [J]. IEEE Transactions on Software Engineering, 1980, 6(2): 118 - 125.
- [7] 毛晓光, 邓勇进. 基于构件软件的可靠性通用模型[J]. 软件学报, 2004, 15(1): 27 - 32.
- [8] 张友生. 软件体系结构[M]. 北京: 清华大学出版社, 2004.