文章编号:1001 - 2486(2007)02 - 0081 - 04

软件构件可靠性与费用分配最优模型*

沈雪石,陈英武

(国防科技大学 信息系统与管理学院,湖南 长沙 410073)

摘 要:针对软件构件可靠性和费用分配问题,给出一种可靠性和费用分配最优模型。文中将软件系统可靠性定义为软件构件失效密度、操作剖面、构件使用矩阵以及软件无失效运行时间的函数,描述了费用最优模型的建立和利用非线性规划理论求解模型的步骤,有效地处理了带有复杂计算的目标函数和约束条件的可靠性和费用最优分配问题。计算实例表明,利用该模型进行可靠性和费用分配是可行的。

关键词:软件可靠性;可靠性费用;最优模型

中图分类号:TP302.7 文献标识码:A

An Optimal Model for the Allocation of Software Reliability and Cost

SHEN Xue-shi, CHEN Ying-wu

(College of Information System and Management, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: An optimal model is presented in this paper, aiming at facilitating cost effective effort allocation among software components so that a quantified system reliability goal is attained. The software reliability is defined as a function of component failure intensities, the operational profile, expected component utilizations and the stated mission time for failure-free execution. The details of building the optimal model and the steps of the solution are presented. In the process of solution, nonlinear programming theory can be used to solve the optimal model. Finally, a simple example is provided to show the feasibility of the optimal model for the allocation of software reliability and cost.

Key words: software reliability; reliability cost; optimal model

随着计算机软件广泛应用,软件可靠性和费用已成为软件系统的重要质量特征[1],软件构件可靠性与费用的分配和权衡问题已成为软件工程领域的重要研究内容,受到越来越多的研究者的关注。如,Poore,Mills 和 Mutchler[2] 使用电子表格的方法,实现了软件构件可靠性分配的多种策略。Zahedi 和 Ashrafi[3] 采用 AHP 方法,以费用为约束,对软件系统结构进行建模,给出了实现系统可靠性最大化模型。Boehm 等人提出用 COCOMO 模型对软件项目开发费用进行估计[10]。Huang 和 Lyu[4-5] 在软件可靠性增长测试中,运用 LTE(logistic testing-effort)函数,实现可靠性测试费用的最优化分配。并且,还有一些文献[6-7] 对有关最佳软件发布时间问题中的费用和可靠性间的权衡进行了讨论。

本文根据软件可靠性的特点,即软件可靠性与软件的操作剖面、构件的期望运行时间等软件运行状态密切相关[1],将软件系统可靠性定义为软件构件失效密度、操作剖面、构件使用矩阵以及软件无失效运行时间的函数,这一定义不同于上述相关文献的研究。本文采用类似 Zahedi 和 Ashrafi 的 AHP 方法,给出一种适用于软件系统开发早期可靠性和费用分配的最优模型。模型以定量的方式对软件构件可靠性(失效密度)进行分配,目标是实现软件系统可靠性所需费用最小。最后,通过算例对所给模型进行验证和分析,表明利用该模型进行软件构件的可靠性和费用的最优分配是可行的。

* 收稿日期:2006 - 12 - 28
基金项目:国家自然科学基金资助项目(70272002)
作者简介:沈雪石(1965—),男,博士生。

1 最优化模型及求解

1.1 最优化模型

设 n 表示组成软件的构件数，$\rho(0<\rho<1)$ 表示软件系统可靠性目标，$\tau(\tau>0)$ 是软件运行的任务时间。即，软件系统在 $[0,\tau]$ 的时间间隔内无失效运行的概率至少为 ρ。

设 $TC(\lambda_1,\lambda_2,\cdots,\lambda_n)$ 为由 n 个构件组成的软件系统所需要的总费用，$\lambda_1,\lambda_2,\cdots,\lambda_n$ 是对应构件失效密度的决策变量，$R(\lambda_1,\lambda_2,\cdots,\lambda_n;\tau)$ 为软件系统可靠性。则，最优化模型为：

$$\text{Min } TC(\lambda_1,\lambda_2,\cdots,\lambda_n) \tag{1}$$

$$\text{s.t. } R(\lambda_1,\lambda_2,\cdots,\lambda_n;\tau)\geqslant\rho \tag{2}$$

$$\lambda_j\geqslant 0,\quad j=1,2,\cdots,n \tag{3}$$

式(1)、(2)和(3)是一个多变量、非线性的约束费用最优模型。如果管理者的目标是在给定费用的条件下，实现软件系统可靠性最大，那么对本模型进行重构，则可将其转化为系统可靠性最优模型。

1.2 模型求解

将式(1)、(2)和(3)表示的最优化模型转化为最小化约束的标准形式，如式(4)、(5)和(6)所示：

$$\text{Min } TC(\lambda_1,\lambda_2,\cdots,\lambda_n) \tag{4}$$

$$\text{s.t. } g_0(\lambda_1,\lambda_2,\cdots,\lambda_n)=\rho\ln(\rho)-\rho\ln[R(\lambda_1,\lambda_2,\cdots,\lambda_n;\tau)]\leqslant 0 \tag{5}$$

$$g_j(\lambda_1,\lambda_2,\cdots,\lambda_n)=-\lambda_j\leqslant 0,\quad j=1,2,\cdots,n \tag{6}$$

模型求解的第一步是将式(4)、(5)和(6)转变为一组等式，如下所示：

$$\left[\frac{\partial TC}{\partial\lambda_1},\frac{\partial TC}{\partial\lambda_2},\cdots,\frac{\partial TC}{\partial\lambda_n}\right]^{\mathrm T}+u\left[\frac{\partial g_0}{\partial\lambda_1},\frac{\partial g_0}{\partial\lambda_2},\cdots,\frac{\partial g_0}{\partial\lambda_n}\right]^{\mathrm T}-[v_1,v_2,\cdots,v_n]^{\mathrm T}=0 \tag{7}$$

$$ug_0(\lambda_1,\lambda_2,\cdots,\lambda_n)=u\rho\ln(\rho)-u\rho\ln[R(\lambda_1,\lambda_2,\cdots,\lambda_n;\tau)]=0 \tag{8}$$

$$v_jg_j(\lambda_1,\lambda_2,\cdots,\lambda_n)=-v_j\lambda_j\leqslant 0,\quad j=1,2,\cdots,n \tag{9}$$

其中，u 和 $v_1,v_2,\cdots,v_n$ 是分别与不等式(5)和(6)相关的 Lagrange 乘子，变换细节可参见文献[8]。

式(7)、(8)和(9)由 $2n+1$ 个等式组成，式(7)和式(9)各为 n 个，式(8)为 1 个，并且式(7)、(8)和(9)中有 $2n+1$ 个未知变量，分别是构件失效密度 $\lambda_1,\lambda_2,\cdots,\lambda_n$，和 Lagrange 乘子 u 和 $v_1,v_2,\cdots,v_n$。决策变量 $\lambda_1,\lambda_2,\cdots,\lambda_n$，$u$ 和 $v_1,v_2,\cdots,v_n$ 在式(7)、(8)和(9)中的可行解，也是式(1)、(2 和(3)或式(4)、(5)和(6)的可行解，u 和 $v_1,v_2,\cdots,v_n$ 是非负数。式(7)、(8)和(9)为 Kuhn-Tucker 方程，其解为 λ_1^*，$\lambda_2^*,\cdots,\lambda_n^*$，$u^*$ 和 $v_1^*,v_2^*,\cdots,v_n^*$。当模型满足 Kuhn-Tucker 充分条件时，如果 $u^*\geqslant 0$，$v_j^*\geqslant 0(j=1,2,\cdots,n)$，则 $\lambda_1^*,\lambda_2^*,\cdots,\lambda_n^*$ 也是式(1)、(2)和(3)或式(4)、(5)和(6)的最优解。

2 可靠性函数和可靠性费用函数

2.1 可靠性函数

本文将软件系统可靠性定义为软件构件失效密度、操作剖面、构件使用矩阵以及软件无失效运行时间的函数。假设构件失效事件是统计独立的，软件构件结构具有与(AND)类型。基于这一假设的系统可靠性是一种保守估计值[9]，Musa 等人对相关假设作了进一步的讨论[1]。

如上所述，则软件系统可靠性表示为：

$$R(\lambda_1,\lambda_2,\cdots,\lambda_n;\tau)=\exp\left\{-\sum_{j=1}^{n}\sum_{i=1}^{m}p_i\mu_{ij}\lambda_j\tau\right\} \tag{10}$$

其中，n 为软件构件数，m 为操作数，τ 为规定运行时间，λ_j 为构件 j 的失效率。$p_i(0<p_i<1)$ 表示执行操作 $i(i=1,2,\cdots,m)$ 的概率，$\mu_{ij}(0\leqslant\mu_{ij}\leqslant 1,i=1,2,\cdots,m,j=1,2,\cdots,n)$ 表示构件 j 在操作 i 中所占的规定运行时间比。操作剖面的 m 维向量 p_i 表示为 $\boldsymbol{p}$，$m\times n$ 维的构件使用矩阵 μ_{ij} 表示成 $\boldsymbol{\mu}$。式(10)中的参数是预先确定的参数，对于每一行 $i(i=1,\cdots,m)$，有 $\sum_{i=1}^{m}p_i=1$ 和 $\sum_{j=1}^{n}\mu_{ij}=1$。

$\sum_{i=1}^{m} p_i\mu_{ij}$ 为构件 j 所占系统操作中总任务时间的规定比率,构件 j 的规定运行时间是 $\sum_{j=1}^{n} p_i\mu_{ij}\tau$。设 $\phi_j = \sum_{i=1}^{m} p_i\mu_{ij}, 0 < \phi_j < 1 (j = 1,2,\cdots,n)$,且 $\sum_{j=1}^{n}\phi_j = 1$,则称 $\phi_j\tau$ 为构件 j 的运行时间。假设构件 j 的失效密度为 λ_j,则在其运行时间内的可靠性为:

$$R(\lambda_j;\phi_j\tau) = \exp\{-\sum_{i=1}^{m} p_i\mu_{ij}\lambda_j\tau\} = \exp(-\phi_j\lambda_j\tau) \tag{11}$$

则由 ϕ_j 的定义可知,式(10) 与式(12) 等价,即:

$$R(\lambda_1,\lambda_2,\cdots,\lambda_n;\tau) = \prod_{j=1}^{n}\exp(-\phi_j\lambda_j\tau) \tag{12}$$

2.2 可靠性费用函数

费用最优模型要求软件构件失效密度与达到规定失效密度所需费用之间存在一种定量关系。因此,本文给出了表示三种定量关系的费用函数:线性函数、对数指数函数和反幂函数,三种函数的表示式分别为式(13)、(14)和(15),图1(a)、(b)和(c)分别是它们的示例图。三种函数都源于费用和可靠性之间的真实假设。即,λ_j 中的费用函数应是严格递减的,则 $\partial C/\partial\lambda_j < 0$,并且呈现凸性。当 C 是严格递减时,则 $\partial^2 C/\partial\lambda_j^2 \geqslant 0$。式(13)、(14)和(15)的三种函数都满足假设所需的性质。

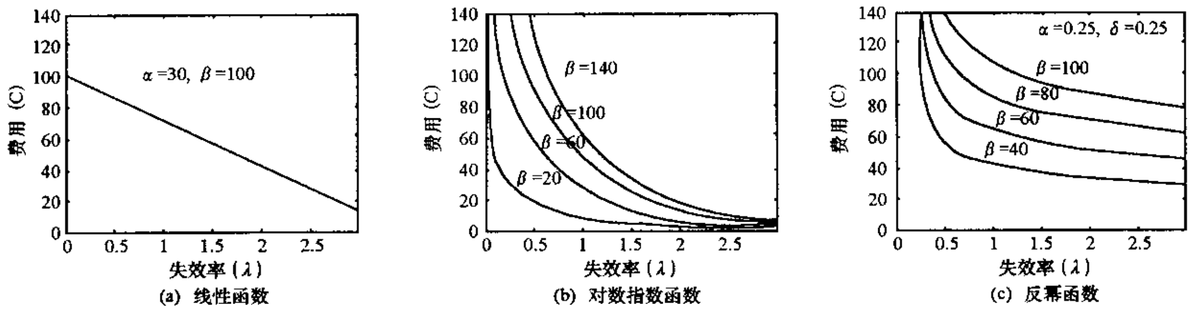

图1 三种费用函数示例

Fig. 1 Examples of the three cost functions

$$C(\lambda) = \begin{cases} -\alpha\lambda + \beta, & \lambda \leqslant \beta/\alpha \\ 0, & \text{其它} \end{cases} \tag{13}$$

$$C(\lambda) = \begin{cases} -\beta\ln(1-\exp(-\lambda)), & \lambda > 0 \\ +\infty, & \text{其它} \end{cases} \tag{14}$$

$$C(\lambda) = \begin{cases} \dfrac{\beta}{(\lambda-\delta)^{\alpha}}, & \lambda > \delta \\ +\infty, & \text{其它} \end{cases} \tag{15}$$

假设构件 j 的费用是可以从总费用 TC 中分离,则:

$$TC(\lambda_1,\lambda_2,\cdots,\lambda_n) = \sum_{j=1}^{n} C(\lambda_j) \tag{16}$$

式(16)中的 $C(\lambda_j)$,可选择使用式(13)、(14)和(15)三种费用函数之一进行计算。

3 算 例

由于篇幅所限,本文仅给出由两个构件组成的软件系统算例,其使用环境由两个操作构成,分别占系统运行时间的 70% 和 30%。操作 1 执行时,调用构件 1 和构件 2,构件 1 所需的平均处理时间为 60%,构件 2 为 40%。操作 2 执行时只调用构件 2。相关参数如下:

$$n = 2, \rho = 0.99, \boldsymbol{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}, m = 3, \tau = 1, \boldsymbol{\mu} = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} = \begin{bmatrix} 0.6 & 0.4 \\ 0.0 & 1.0 \end{bmatrix}$$

则,由上述参数可计算下列参数:

$$\lambda_s = -\ln(\rho)/\tau = 0.0101 \quad \begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix} = \begin{bmatrix}p_1\mu_{11} + p_2\mu_{21}\\ p_1\mu_{12} + p_2\mu_{22}\end{bmatrix} = \begin{bmatrix}0.42\\ 0.58\end{bmatrix}$$

3.1 线性费用函数

如果算例采用线性费用函数,则构件 1 和构件 2 所对应的参数如下:

$$\alpha_1 = 95, \beta_1 = 1350; \alpha_2 = 55, \beta_2 = 1250$$

费用最优模型的解为 $\lambda_1^* \approx 0.0239$ 和 $\lambda_2^* \approx 0$,可靠性所需总费用为 2597.73,对应的 Lagrange 乘子 $u^* \approx 228.4752$, $v_1^* = 0$ 和 $v_2^* \approx 76.1905$。

线性费用函数的优点是简单。对于简单项目,或复杂性较小的构件,可靠性与其所需费用之间采用线性关系是适当的。

3.2 对数指数费用函数

如果算例采用对数指数费用函数,则构件 1 和构件 2 所对应的参数如下:

$$\beta_1 = 238, \beta_2 = 305$$

费用最优模型的解为 $\lambda_1^* \approx 0.0105$ 和 $\lambda_2^* \approx 0.0097$,可靠性所需总费用为 2500.24,相应的 Lagrange 乘子 $u^* \approx 54299.6265$, $v_1^* = 0$ 和 $v_2^* = 0$。

对数指数费用函数具有非线性特征,与线性函数相比,更能反映可靠性与费用间的真实关系,并且每个构件只需确定一个参数(β_j)。其不足是缺乏灵活性,不能改变曲线的形状。

3.3 反幂费用函数

如果算例选择反幂费用函数(式(15)),则构件 1 和构件 2 所对应的参数如下:

$$\alpha_1 = 0.30, \beta_1 = 320, \delta_1 = 0; \alpha_2 = 0.25, \beta_2 = 410, \delta_2 = 0$$

费用最优模型的解为 $\lambda_1^* \approx 0.0123$ 和 $\lambda_2^* \approx 0.0084$,可靠性所需总费用为 2550.67,相应的 Lagrange 乘子 $u^* \approx 70102.1942$, $v_1^* = 0$ 和 $v_2^* = 0$。反幂费用函数具有基本 COCOMO 估计模型的一般形式[10],其不足是,必须为每个构件设定两个或三个参数(取决参数 δ 是否为 0)。

4 结 论

本文给出了一种软件构件可靠性和费用分配最优模型,描述了最优化模型建立和求解的步骤。所给模型不依赖于软件测试阶段的测试数据,适用于软件构件可靠性和费用的分配,而不是可靠性估计,这有利于管理者在软件开发早期,采用这种分配方法计划软件构件的可靠性和费用。所给出的三种可靠性费用函数,为回归分析提供了合理的起点。最后,通过算例对模型进行验证和分析,表明利用该模型对软件构件可靠性和费用优分配是可行的。

参考文献:

[1] Musa J D. Software Reliability Engineering[M]. US:McGraw Hill, 1999.
[2] Poore J H, Mills H D, Mutchler D. Planning and Certifying Software System Reliability[J]. IEEE Software, 1993, 10(1):88–99.
[3] Zahedi F,Ashrafi N. Software Reliability Allocation Based on Structue, Utility, Price and Cost[J]. IEEE Trans. Software Eng., 1991, 17(4):345–356.
[4] Huang C Y, Lo J H, Kuo S Y, et al. Optimal Allocation of Testing-resource Considering Cost, Reliability, and Testing-Effort[C]// Proceedings 2004 Pacific-Rim Dependable Computing, French Polynesia, 2004:103–112.
[5] Kuo S Y, Huang C Y, Lyu M R. A Framework for Modeling Software Reliability, Using Various Testing-Efforts and Fault-Detection Rates[J]. IEEE Transactions on Reliability, 2001, 50(3):310–320.
[6] Kitchenham B A. Empirical Studies of Assumptions that Underlie Software Cost-estimation Models[J]. Information and Software Technology, 1992, 34(4):211–218.
[7] Leung Y W. Optimal Reliability Allocation for Modular Software Systems Designed for Multiple Customers[J]. IEICE Trans. Information and Systems, 1996, E79–D(12):1655–1662.
[8] 黄红选,韩继业. 数学规划[M]. 北京 清华大学出版社,2006
[9] Bergman B. On Reliability Theory and Its Applications[J]. Scandinavian J. Statistics, 1985, 12:1–41.
[10] Boehm B, Valerdi R, Lane J A. et al. COCOMO Suite Methodology and Evolution[J]. CrossTalk,2005, 4:20–25.