软件系统的可靠性分配模型及优化研究

张天丹
(南通师范高等专科学校,江苏 南通 226000)

【摘 要】本文着重研究基于架构的软件系统的可靠性分配模型以及相应的优化算法,以最少的成本创建具有最优可靠性的系统,不仅考虑了开发成本的最小化,而且优化了软件系统的可靠性。本文所提出的模型和算法对于降低软件系统开发成本、提高软件系统可靠性具有一定的理论参考价值,并且能够在软件开发之前对已经设计出的软件架构进行预评估,从而预测软件系统的开发成本并对软件可靠性进行优化分配,具有较高的实际应用价值。

【关键词】软件可靠性;可靠性分配;成本函数;软件架构
【中图分类号】TP311.52 【文献标识码】A 【文章编号】1673-1891(2014)03-0060-04

1 可靠性研究现状

在许多项目开发过程中,对可靠性没有提出明确的要求,开发商(部门)也不在可靠性方面花很多的精力,往往只注重速度、结果的正确性和用户界面的友好性等,而忽略了可靠性的研究。于是项目在投入使用后遇到大量可靠性问题,增加了维护困难和工作量,还有就是维护费用的增加,严重时只有束之高阁,无法投入实际使用。在硬件元器件价格急剧下降的今天,软件可靠性问题造成费用迅速飙升。

2 软件系统的可靠性分配

2.1 软件可靠性

1983年美国IEEE计算机学会对“软件可靠性”作出了明确定义,此后该定义被美国标准化研究所接受为国家标准,1989年我国也接受该定义为国家标准。该定义包括两方面的含义:

(1)在规定的条件下,在规定的时间内,软件不引起系统失效的概率;

(2)在规定的时间周期内,在所述条件下程序执行所要求的功能的能力[1];

其中的概率是系统输入和系统使用的函数,也是软件中存在的故障的函数,系统输入将确定是否会遇到已存在的故障(如果故障存在的话)。

2.2 软件系统可靠性分配定义

软件可靠性分配是软件开发过程中的一个重要的可靠性工作项目,其目的是将软件系统的可靠性分配给软件的各个子系统。

进行软件可靠性分配,必须具备下列四个条件[2]:

(1)必须制定出软件需求说明书和软件的可靠性开发大纲;

(2)必须要有定量的可靠性指标和定义了软件的运行剖面;

(3)必须对系统失效作出了明确的定义;

(4)程序必须能够进行结构分解;

软件系统可靠性分配必须满足的基本要求如下:

$$\lambda(\overline{\lambda_1},\overline{\lambda_2},\cdots\overline{\lambda_n}) \le \lambda_s \qquad (1)$$

其中λ_s— 系统的故障率指标;$\overline{\lambda_i}$(i=1,2,……,n) —第i各子系统分配的故障率指标;$\lambda(\overline{\lambda_1},\overline{\lambda_2},\cdots\overline{\lambda_n})$—系统的故障函数;

从上面可以看出有n个待求解的参数,这种情况表明满足上式的解有无限多组。如何从各种可能的解中选出一组合理的可行解呢?这正是软件系统的可靠性分配所要解决的问题。

3 基于架构的软件系统的可靠性分配模型

3.1 基于架构的软件可靠性概述

3.1.1背景介绍

软件架构是在一个抽象的层次上对软件整体结果的抽象,虽然好的架构并不一定能保证可以获得高可靠性的软件产品,但可以肯定建立在不良架构上基础上的软件是不可能获得较高的可靠性的。

由于架构的不合理使得软件达不到预期可靠性目标而进行结构修改的代价是相当高的。因此,在软件开发初期,对基于架构的软件系统进行可靠性分配,是保证软件产品质量、减少开发费用的一个很重要的方法。

基于架构的软件系统的可靠性分配DRS(Architecture-based software system reliability allocation)是在保证系统可靠性的前提下,对组成软件架构的成分做出分析,分离为一个个的基本软件元素,并由此对基于架构的各个软件元素进行可靠性目标的分配,以达到在可靠性预估成本一定的情

收稿日期:2014-05-22
作者简介:张天丹(1978-),女,江苏南通人,硕士,讲师,研究方向:数据库、计算机网络及其应用技术。

况下,开发得到的软件可靠性最大,或在系统可靠性目标一定的情况下,可靠性预估成本最小。软件元素被定义为诸如操作、子架构、组件、对象或者是其它能够用于可靠性分配的实体。

软件系统的架构模型如图1:

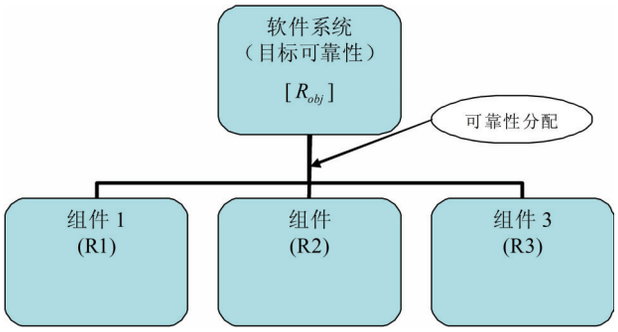

图1　基于架构的软件系统的可靠性分配模型

3.1.2 软件可靠性成本

基于架构的软件可靠性分配模型,克服了在系统设计完成后,要提高软件系统的可靠性需要花费巨大成本的困难。并且随着软件开发的进行,越往后期发展,所需花费的成本就会越多,所以研究可靠性成本预估函数,并且以可靠性预估成本最小化为目标函数建立非线性规划模型,从而得到组建可靠性与成本之间的优化结果,并且通过模型进行软件可靠性分配是可行的[3]。

软件可靠性成本的发生是为了保证软件系统的质量。因此,在软件开发过程中要努力达到其两者之间的最优化结合。

3.2 基于架构的软件系统的可靠性分配DRS

3.2.1 可靠性预估成本

可靠性预估成本是在软件架构设计完成之后,对基于此架构的软件系统进行开发,预计满足可靠性要求的软件所需可能的成本总和。具体可以根据基于架构的软件可靠性相关理论,分析系统架构,从中抽取出我们需要的简单结果,进行成本估计,然后再进行系统整合从而得到整个系统的成本预估。

费用函数是指可靠性与开发成本之间的关系,即达到软件可靠性要求所需花费的各种人力、物力、财力、时间等的综合,最初称其为努力函数(effort function)。由于很难获得各软件元素费用与可靠度之间的统计数据,无法建立经验关系式,1986年Dale和Winterbottom提出了费用函数必须满足的一些性质,克服了这一问题:

假设对于任意组件i的可靠性R_i,存在$0\leqslant R_i(1)\leqslant R_i(2)\leqslant 1$;

记$c(R_i(1),R_i(2))$表示将可靠性R_i从$R_i(1)$提高到$R_i(2)$所花费的努力,并且其有六个性质,也可以说是满足的条件,具体为以下六个:

1.$c(R_i(1),R_i(2))\geqslant 0, 0\leqslant R_i(1)\leqslant R_i(2)\leqslant 1$;

2.$c(R_i(1),R_i(3))=c(R_i(1),R_i(2))+c(R_i(2),R_i(3)), 0\leqslant R_i(1)\leqslant R_i(2)\leqslant R_i(3)\leqslant 1$;

3. $c(R_i)$可微;

4. $\frac{\partial^2}{\partial_{R^2}} c(R_i)\geq 0, 0\leq R_i\leq 1$

5.任意固定的R_i,$0\leqslant R_i(1)$,若$R_i(2)$趋于1,则$c(R_i(1),R_i(2))$趋于∞;

6.$c(R_i)$是单调增函数;

假设基于架构的软件系统的失效率完全来自于组成其本身的组件;而组件的失效率与其本身的规模成正比,也就是说规模越大失效率就越高,然后根据失效率与可靠性之间的指数关系建立软件可靠性与开发成本之间的费用函数——可靠性预估成本函数,用于基于架构的软件可靠性分配[4]。函数关系为:

$$C(r_i)=-f_i/\ln r_i+B_i \qquad (2)$$

其中,i表示组成软件架构的任意一个组件;$C(r_i)$表示组件i的可靠性预估成本;f_i表示组件i的可靠性提高的费用系数(简称系数)与组件的规模、性质等参数有关,$f_i>0$;r_i表示组件i的初始可靠性,B_i表示组件的可靠性预估成本的基值,即组件i的最低可靠性预估成本。所以系统的所有组件的总成本C:

$$C=\min\left(\sum_{i=1}^{n} c(r_i)\right) \qquad (3)$$

于是,利用matlab编制出一个关于绘制可靠性预估成本函数的程序,可以通过输入费用系数f和可靠性预估成本B的基值,看到成本与可靠性之间的曲线变化关系。

组件可靠性r与成本C之间费用函数,如图2所示:(递增函数)

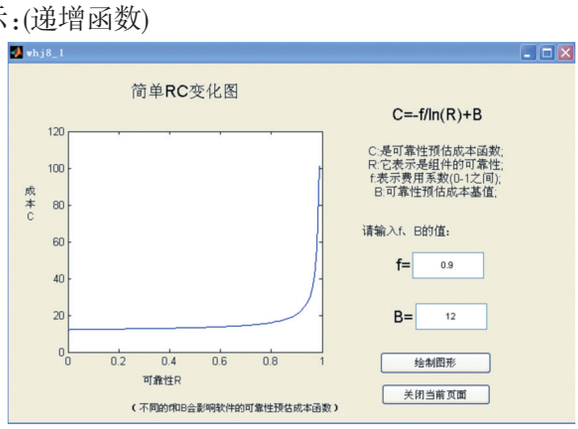

图2　RC图(各个曲线的f,b各不相同)

从图2看出:提高软件可靠性与降低软件的开发成本是一对矛盾;并且不同的费用系数f或是不

同的成本基值B会导致不同的结果；而已有的多组实验数据与本模型曲线能达到很好的拟合。根据此模型，可得出结论：一个软件要达到接近100%的可靠性需要花费相当大的开发成本，或者说，具有100%的可靠性的软件是不可能的。

3.2.2 影响基于架构的软件可靠性分配DRS的因素

影响因素有：组件的可靠性，组件间的关系，架构结果，架构风格等。因此，要通过改变组件的可靠性、组件间的关系结果、架构风格等方法来提高系统的可靠性，首先要给出它与组件、组件分布、架构风格等因素间的关系，然后根据系统可靠性目标、软件架构和组件关系，结合可靠性成本费用函数，为组件分配满足要求的，并且使基于架构的系统预估可靠性成本相对较小的可靠性值。若每个组件均达到了可靠性要求，则系统肯定能够达到其可靠性目标。

3.3 优化的基于架构的可靠性分配算法1—QDRS

3.3.1 QDRS算法

在软件设计阶段，对于由n个组件组成的软件架构，根据基于架构的软件可靠性评估理论，得出系统的可靠性。此时，要提高软件系统的可靠性指标，系统的可靠性预估成本显然要提高。并且，可靠性提高的幅度越大，所需要的成本越高[5]。

该优化算法QDRS（基于架构的软件系统的可靠性分配之比值）是根据成本C与可靠性R之间的比值Q（C/R），要使可靠性高而又保证成本尽可能的低，则Q值必须尽可能的小；而C和R之间是呈非线性关系，这个非线性规划如下。

假设，如果要使得系统的架构的软件可靠性由R提高到R_{obj}的，其总成本为$\sum_{i=1}C(r_i)(i=1,2,3\cdots)$（公式2），则要想在其为最小值的情况下将可靠性分配给各个构件，则有：

目标函数：$\min(\sum_{i=1}^{n}c(r_i))$

约束函数为：$R\geq R_i(i=1,2\cdots n)$

其中：n—组件的个数

R—基于架构的软件的可靠性

R_{obj}—系统的可靠性指标

$C(r_i)$—软件元素i的可靠性预估成本

注意：$C(r_i)=-f_i/\ln r_i+B_i$（可靠性预估成本函数），$0<R_s<1$；

求解这个问题，首先要获得基于架构的软件可靠性评估函数，建立可靠性分配模型，然后根据可靠性预估成本函数，在软件系统可靠性目标给定的约束条件下，通过动态规划进行分配，达到最小开发预估成本与最高可靠性目标之间的优化组合。

QDRS算法：

```
1.Input n个模块的R值
2.Input n个模块的F值
3.Input n个模快的B值
4.for i=1:n
1） for j=1:n
2） for z=1:n
3） for k=1:n
4） R=r1(i)*r2(j)*r3(z)*r4(k)……; %系统可靠性R及成本C
5） C=c1(i)+c2(j)+c3(z)*c4(k)……;
6） Q=C./R;          %Q值
7） if R>Robj转向步骤(4)   %保证约束可靠性
```

可以利用matlab实现QDRS算法，矩形实验室Matlab（matrix laboratory），它是一种用于数值计算、符号运算、图形计算以及程序设计的软件；matlab作为一种高级计算语言，可以像c等其他高级语言一样进行程序设计，即编制以m为扩展名的文本文件（简称m文件）；而QDRS（基于架构的可靠性分配之比值）借助matlab的功能，对C、R、Q进行一定的约束，并且求出较优解。

3.3.2 QDRS的数值仿真与分析

设一个系统由4个相互独立的组件R_1，R_2，R_3，R_4串联组成，并且任意组件的失效都会导致系统失效。设各组件的费用系数f为：0.40,、0.72、0.5，0.62；并且各个组件的成本基值分别为：10、18、25、20；为了使系统开发成本总量最小，且系统整体可靠性不低于0.94，则各个组件应该分配多大的可靠性。计算精度为0.01。

●分析：

首先，本例由四个元素构件1，2，3，4组成，则$R=r_1*r_2*r_3*r_4$（串联系统的可靠率，根据快速分配法）

然后，将$C(r_i)=-f_i/\ln r_i+B_i$和$R=r_1*r_2*r_3*r_4$代入公式2，建立模型，根据动态规划方法，进行组合，得到满足软件可靠性要求的构建可靠性组合：

目标函数：$\min C=\int_{i=1}^{4}(\frac{-f_i}{\ln r_i})+B_i$ ；

约束条件：$R=r_1*r_2*r_3*r_4$ ；

$R\geq R_{obj}$ ；

即 $r_1\times r_2\times r_3\times r_4\geq R_{obj}$ ；

基于架构的软件系统的可靠性分配算法实质上是要求在相对较小的成本C下，得到有较优可靠性R的算法；则，可以看出，该算法是要求C/R尽

可能的小,从而达到优化的可能;利用matlab软件,运行程序后,最后进行比较,得到最后的较优解。

把最后的结果进行比较,即得到了可靠性满足要求,即

R(5) (4) (5) (4) =9.412880e-001,C(5) (4) (5) (4) = 2.338770e+002,Q(5)(4)(5)(4)=2.484648e+002

所以:结果为:r_1=0.99, r_2=0.98, r_3=0.99; r_4=0.98;

R=$r_1 * r_2 * r_3 * r_4$=0.94;

总成本C=233.88;

3.3.3 QDRS小结

本优化算法QDRS(基于架构的软件系统的可靠性分配模型之比值),不依赖于测试数据,通过软件可靠性与开发成本之间的可靠性预估成本函数,通过动态规划进行可靠性分配,并且利用Q值来约束成本与可靠性之间的优化搭配;此方法利于开发人员在软件开发初期阶段,采用这种分配方法有计划的分配构件的可靠性和成本[6]。根据观察图形变化,可以知道优化之后的R-C跟R-Q分布优化了(图3)。

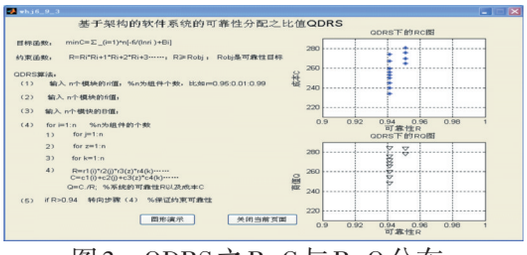

图3　QDRS之R-C与R-Q分布

与传统的软件系统的可靠性分配模型相比较,现如今的分配算法要远比传统的优化;无论是快速分配算法还是等分配算法,都没有能够很好的实现对于可靠性以及成本的优化分配,这是本文研究的内容,在以传统为借鉴的基础上,更好的实现可靠性分配。

传统的可靠性分配算法简单易懂,却没有很好实现可靠性和成本之间的双向优化,但是其对以后的可靠性分配研究有借鉴意义;QDRS在以前分配的基础上,进一步的提高了优化程度。

其实不同的算法都有自己不同的优缺点,我们需要从每一种算法中取长补短,这样才可以得到更加优化的软件系统的可靠性分配算法。

注释及参考文献:

[1]Mary E.H,Ming zhao,Niclas Ohlsson.Planning Models for Software Reliability and Cost[J].IEEE Trans.on Software Engineering,1998,24(6):420-433.

[2]陆文,徐峰,吕建.一种开放环境下的软件可靠性评估方法[J].Chinese Journal of computers,2010,33(3)452-462.

[3]徐高潮,刘新忠,胡亮,等.引入关联缺陷的软件可靠性评估模型[J].Journal of Software,2011,22(3):439-450.

[4]Gokhale SS,Trivedi KS.Analytical models for architecture-based software reliability prediction:A unification framework.IEEE Trans.on Reliability,2006,55(4):578-590.

[5]Trung PT,Thang HQ.Building the reliability prediction model of component-based software architectures. Int'l Journal of Information Technology,2009,5(1):18-25.

[6]沈雪石,陈英武.软件构件可靠性与费用分配最优模型[J].国防科技大学学报,2007(2):81-84

Research on Reliability Allocation Model and Optimization of Software System

ZHANG Tian-dan

(Nantong Normal College, Nantong, Jiangsu 226000)

Abstract: This paper focuses on the study of the reliability distribution model of software system architecture and corresponding optimization algorithm based on,creating system of optimum reliability with minimum cost, which not only considers minimizing development costs, but also optimizes the reliability of the software system. In this paper, the proposed model and algorithm for the theory has a certain reference value to reduce software development costs, improve the reliability of software systems, and can evaluate designed software architecture before software development so as to predict the development of software system cost and make an optimal allocation of software reliability, possessing higher practical application value.

Key words: software reliability; reliability allocation; cost function; software architecture