试验设计在软件可靠性测试中的应用

付剑平 陆民燕 阮 镰 黄百乔

(北京航空航天大学 工程系统工程系, 北京 100191)

摘 要: 为了有效利用资源、优化测试数据,在软件可靠性测试中引入试验设计:在操作剖面开发过程中应用正交试验设计,选择典型操作;在测试数据生成过程中使用组合设计,结合发生概率得到优化测试数据集.完整的软件可靠性测试过程包括确定系统模式剖面,然后根据正交试验设计确定操作剖面,根据组合设计确定运行分类,最后生成可靠性测试数据执行测试.在此基础上开发了测试数据自动生成工具,进行了实例应用.结果表明该方法不仅能为软件生成可靠性测试数据,还能发现更多的软件缺陷.

关 键 词: 可靠性; 测试; 概率; 缺陷

中图分类号: TP 311.5

文献标识码: A **文 章 编 号**: 1001-5965(2008)12-1379-05

Application of experimental design in software reliability test

Fu Jianping Lu Minyan Ruan Lian Huang Baiqiao

(Dept. of System Engineering of Engineering Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: Experimental design was introduced to software reliability test to take full advantage of resources and optimize test data. Typical operations were selected by orthogonal experiment design in the development of operational profiles and optimized results were produced by combinatorial design in the generation of test data considering of occurrence probabilities. A full software reliability test consists of determining the system modes profile, developing operational profiles by orthogonal experiment design, choosing run categories by combinatorial design and generating reliability test data to execute test finally. An automation test data generation tool was developed according to the method and an example was applied with it. Results show that it can not only generate software reliability test data but also finding more software defects.

Key words: reliability; testing; probability; defects

软件可靠性测试是基于操作剖面的随机测试,根据实际情况建立操作剖面和根据操作剖面准备测试数据是其中两项关键内容.软件测试非常昂贵,一般的软件测试大概需要花费软件开发费用的33%~40%,还将极大地延长软件开发周期[1].与一般软件测试相比,软件可靠性测试需要构造操作剖面、准备更多的测试,因此花费更多.操作内部的分布可能并不均匀,不同的运行可能具有不同的发生概率,在操作内部随机选取测试数据的方式并不符合软件的实际使用情况.为了有效利用资源、更好地发现软件缺陷,需要选择合适的操作和测试数据进行可靠性测试.

由于试验设计能优化试验数据,在一定的覆盖强度下减少试验次数,国内外一些研究者开始在软件可靠性测试中使用试验设计.文献[2]在评估一些具有高关键度低概率事件的软件可靠性时就使用了试验设计.当软件在各种紧急条件或例外事件下执行测试时,试验设计能用于确定不同模式的执行顺序.文献[3]提出了基于正交试验设计的软件可靠性测试方法,该方法根据正交表的特性对输入用例进行正交处理,从而简化测试用例集,进而简化操作剖面.文献[4]根据软件可靠

收稿日期: 2007-12-13

作者简介: 付剑平(1979—),男,湖南衡阳人,博士生,fu_jianping@dse.buaa.edu.cn.

性测试使用随机方式,未对测试数据进行优化,导致用例质量不高的特点,采用组合设计思想优化测试数据提高用例质量.

上述方法虽然解决了可靠性测试中出现的一些问题,但也存在不足.文献[2]使用试验设计估计不同模式间延续效应的影响,但对于节约测试资源和优化测试数据的帮助不大.文献[3]虽然提出可以简化操作剖面,但实际还是解决测试用例的选取问题,未对操作剖面的构造提出更有建设性的方法.文献[4]使用组合设计优化测试数据,由于未考虑发生概率,得到的测试数据不能反映软件的实际使用情况.

针对这些情况,本文提出在操作剖面构造和测试数据选取过程中使用不同的试验设计方法,从而实现在现有的测试资源下尽可能的优化测试数据、提高软件可靠性测试的故障发现概率.

1 正交设计在剖面开发中的应用

操作剖面在构造过程中难免产生大量操作,限于资源,不能对所有操作进行分析,需要选择具有代表性的操作.正交试验设计具有均匀分散的特点,能在输入空间内选择分布均匀的操作,而测试的操作在输入空间内分布越均匀,就越有可能发现软件缺陷,因此可以在软件操作剖面开发过程中使用正交设计.如果仅仅使用正交设计,生成的操作数目可能仍然无法控制,超出实际分析的需要;同时如果在输入空间内选择一部分操作而抛弃其余的操作,就不能满足输入空间的覆盖要求.另外由于软件可靠性测试是基于使用的测试,在应用正交设计的过程中还应考虑发生概率.综合这些情况,本文提出基于正交设计的软件操作剖面构造方法.

1) 确定额定值.额定值是允许的操作剖面的最大操作数目,根据分析费用计算,用于控制操作剖面的最终操作数目.

2) 建立初始操作剖面.为软件的每个显著不同的处理分配一个操作或属性值后,确定它们的发生概率.对于能够细分的操作暂时不作细分,只确定其中的属性值和发生概率.

3) 判断当前操作数目是否超出额定值.由于分析费用的限制,可能一开始的操作数目就大于分析需要,此时将不再进行操作细分工作.

4) 判断是否存在需要细分的操作.如果不存在这种操作,也没有必要进行下一步.

5) 细分操作.此时首先需要选择发生概率最大的操作,应用正交试验设计进行细分,细分过程中需要注意操作数量不能超出额定值;细分完成后根据当前的操作数量决定是否继续细分.

6) 整理剖面.上述过程完成后必然形成一个剖面,可能是初始操作剖面,也可能是操作细分后的剖面.此时可根据各种不同的需要对剖面进行处理,如使用其它方法减少操作或根据测试人员经验调整操作,无论怎样,都要求最后的操作数目与分析费用一致、所有操作的发生概率之和为1.

2 组合设计在数据选取中的应用

文献[4]提出的基于组合设计的软件可靠性测试用例生成方法是将操作中的每个参数作为一个因素,参数的不同取值作为水平,然后采用组合设计算法实现基于 n 维覆盖的测试用例集.本文方法与之类似,也是首先选取因素和水平,然后使用组合设计.但在具体实现上,本文使用了运行分类(run category)概念,同时考虑了操作内部的发生概率,方法的具体过程如下:

1) 确定操作的输入变量.操作是一个主要的系统逻辑任务,持续时间短,结束时将控制权还给系统[5].操作可看成与操作相关的输入变量集合,而输入变量又可分为直接输入变量和间接输入变量.由于一些隐蔽但重要的间接输入变量容易被忽略,在分析过程中需要对软件的使用环境仔细考虑,尽量避免遗漏一些间接输入变量.

2) 划分输入变量取值水平.输入变量的取值不同,软件的执行可能相同,也可能不同.将对软件影响相同的取值分为一组,就成为变量的一个取值水平.由于不同取值的发生概率可能不同,方法为每个取值水平确定发生概率,并假设取值水平内部为均匀分布.

3) 选择运行分类.运行分类是一组同质度接近1的运行集合[5].在组成操作的每个输入变量的取值水平中任取一个水平,组成一个运行集合,集合内部的所有元素很可能具有相同的失效行为,这样的集合就可以看作操作的一个运行分类.由于这样组成的运行分类数目很大,可能是所有变量取值水平数目的乘积,因此还需进行优化.将输入变量作为因素、变量的取值水平作为水平进行组合设计,在组合设计过程中首先找出那些组合发生概率(各变量取值水平概率之积)较大的运行分类,就能得到优化的运行分类集合.

4) 确定测试数据.在数据准备或测试执行过程中,选定操作的运行分类后,根据均匀分布假设,其中的每个运行都可作为测试数据.

3　基于试验设计的软件可靠性测试

传统的软件可靠性测试方法[6]包括构造系统模式剖面、构造功能剖面、构造操作剖面和生成可靠性测试数据 4 步.其中构造系统模式剖面是指通过分析软件使用方式将其划分成一组系统模式并确定各模式的发生概率;构造功能剖面是指对某一特定的系统模式分析其所有功能和功能的发生概率;构造操作剖面首先需要确定各功能的操作和发生概率,然后通过合并相同操作的方式得到操作剖面;可靠性测试数据生成分为随机选择操作和随机抽取输入变量取值两步.

由于功能剖面可以看作是系统模式剖面的进一步细化,本文未再单独提出构造功能剖面.通过在构造操作剖面和生成测试数据过程中应用前面提到的方法,本文提出新的基于试验设计的软件可靠性测试方法如下.

1) 确定系统模式剖面

2) 为每个系统模式确定操作剖面

需要为已确定好的每个系统模式都建立一个操作剖面.不同的操作剖面中可能出现相同的操作,但其发生概率未必相同.在为每个系统模式建立操作剖面时,对于那些包含多属性的操作暂时不作细分.

3) 建立系统操作剖面

系统操作剖面综合了所有需要测试的操作剖面,用于宏观控制测试资源.

4) 调整系统操作剖面

调整系统操作剖面就是应用第 1 节中提到的方法,对系统操作剖面中包含多属性的操作进行细分,同时整理剖面.该步骤应用正交试验设计,既考虑资源限制又顾及发生概率,能够选出典型操作.

5) 调整其它操作剖面

对系统操作剖面进行调整后,操作情况可能发生变化:某些操作被删除,某些操作被细分.为了保持操作的一致性,还需要调整各系统模式下的操作剖面.

6) 确定运行分类

该步骤对系统操作剖面中的每个操作确定输入变量、划分取值水平并选择运行分类,具体内容如第 2 节所述.

7) 执行可靠性测试

综合考虑可靠性评估的需要和实际条件的限制,执行一定数量的测试,测试过程中注意失效信息的收集和记录.每次测试的具体过程如下:

① 选择系统模式,根据系统模式设置测试环境;

② 根据此系统模式的操作剖面选取一个操作;

③ 选择操作的一个运行分类;

④ 从选择的运行分类中随机抽取一个运行执行测试.

4　应　用

本文根据上述方法开发了一个软件可靠性测试数据自动生成工具 OT(Orthogonal Tool).OT 能根据用户建立的系统模式剖面和不同系统模式下的操作剖面,自动调整各操作剖面,将操作数目控制在要求范围内;同时在给定测试数目的前提下,自动生成软件可靠性测试数据.

4.1　应用过程

本文使用 OT 对一个下载的某超市集团的信息管理系统进行了软件可靠性测试,生成的系统模式剖面如表 1.

表 1　系统模式剖面

系统模式	发生概率
经理模式	0.3
会计模式	0.3
售货员模式	0.2
仓库管理员模式	0.2

为每个系统模式确定操作剖面后,OT 根据输入的额定值生成了一个含有 40 个操作的系统操作剖面.对系统操作剖面中的每个操作确定输入变量、划分输入变量取值水平并确定发生概率后,点击 OT 的“测试数据生成”,输入需要测试的数目 850,软件就自动生成 850 个测试数据.

4.2　软件缺陷设置

被测软件是基于数据库的管理软件,其主要操作就是添加、删除和查询记录.包括这些公共模块在内,软件共有模块 31 个.在缺陷设置过程中,共设置了软件缺陷 62 个,每个模块 2 个.为了能连续测试,缺陷设置为消息形式,即当软件执行到某缺陷时将自动弹出包含该缺陷编号的消息框,确定后软件可以继续运行.设置的缺陷包括单因素、双因素、三因素和四因素缺陷 4 类,其中单因素缺陷设置在模块的主流程上,双因素缺陷设置在某一条件分支内,三因素和四因素缺陷分别处于嵌套的条件分支内.

4.3　可靠性测试结果

在软件可靠性测试过程中,执行了 OT 生成

的850个测试数据,共发现软件缺陷47个.除此之外,还根据另一种基于使用剖面的软件可靠性测试数据生成方法[7]生成了相同数目的测试数据,该方法在剖面构造和测试数据生成过程中都没有考虑试验设计,共发现41个软件缺陷.在可靠性测试过程中,对于缺陷编号相同的消息无论弹出多少次都只记做一个软件缺陷.两种方法发现的软件缺陷情况如表2、表3所示.

表2 基于试验设计的方法测试结果

缺陷类型	设置缺陷/个	发现缺陷/个	发现比例/%
单因素缺陷	30	28	93.3
双因素缺陷	15	10	66.7
三因素缺陷	15	8	53.3
四因素缺陷	2	1	50
合计	62	47	75.8

表3 未考虑试验设计的方法测试结果

缺陷类型	设置缺陷/个	发现缺陷/个	发现比例/%
单因素缺陷	30	25	83.3
双因素缺陷	15	8	53.3
三因素缺陷	15	7	46.7
四因素缺陷	2	1	50
合计	62	41	66.1

4.4 分 析

对上述结果进行分析发现基于试验设计的方法比未考虑试验设计的方法发现的缺陷数量多.在单因素和双因素缺陷的发现比例上,考虑了试验设计的方法明显高于未考虑试验设计的方法;三因素和四因素缺陷的发现数量基本相同,这是因为考虑试验设计的方法在应用过程中设置的测试数据覆盖强度仅为2,没有考虑更高的覆盖,两种方法对这些缺陷的发现概率都与它们所在操作的发生概率有关.下面是对基于试验设计的软件可靠性测试方法测试结果的详细分析.

1) 方法发现了大多数的单因素和双因素缺陷.设置的单因素与双因素缺陷共45个,可靠性测试发现了其中的38个,占84.4%,大于缺陷的平均发现比例.这说明应用组合设计生成的测试数据具有较强的针对性,能够发现大多数覆盖强度内的软件缺陷.由于当前OT生成测试数据的覆盖强度是2,因此对大于覆盖强度的软件缺陷如三因素和四因素缺陷的发现比例较低.

2) 发现的软件缺陷大都集中在发生概率较高的操作内.发现的47个软件缺陷中有37个集中在发生概率靠前的20个软件操作内,这表明使用越频繁的操作越容易暴露软件缺陷,而方法将发生概率作为操作划分和运行分类的一条选择标准是比较合理的.剩余10个缺陷分布在余下的软件操作中.

3) 未发现的软件缺陷中有9个由于所处操作的发生概率太低,没有生成相关的测试数据,如果增加测试次数也许能为这些操作生成测试数据;有2个虽然分布在发生概率较大的操作内,但由于所处的运行分类的发生概率太低,没有被组合设计选中,方法也不会为这些运行分类准备测试数据;还有4个是三因素或四因素缺陷,前面已指出由于OT的原因这类缺陷不可能全部发现.

5 方法总结

传统方法是在选择操作后直接在操作内部随机选取一个测试数据,本文方法则还需选择操作的运行分类,再在运行分类内随机选取一个测试数据,这样得到的测试数据更接近软件的实际使用情况.

操作的运行分类集合在优化过程中排除了部分发生概率较低且对覆盖没有贡献的运行分类,这一方面使得相同次数下集合内部的运行分类测试的更充分,更有可能发现软件缺陷;另一方面又导致测试偏离了软件的实际使用情况,从而影响了最终的软件可靠性评估.一种比较简单的处理方式是假设排除的运行分类也参与了选择和测试数据生成,且生成的测试数据均匀地分布在实际生成的测试数据中.由于软件在可靠性测试前已经过充分的功能测试,由小概率的运行分类得到的测试数据即使执行也很难发现新的缺陷,因此假设没有新的失效.假设每次测试的执行时间都大体相同,则失效时间修正系数 $F=$总测试时间/实际测试时间$=$总测试数/实际测试数$\approx 1/P_{\text{OptimalSet}}$,$P_{\text{OptimalSet}}$是个常数,表示优化集合内各运行分类的发生概率之和.修正方法如下:

$$t'_i = Ft_i \approx \frac{t_i}{P_{\text{OptimalSet}}} \qquad i=1,2,\cdots,N \tag{1}$$

其中,t'_i 表示修正后的第 i 个失效时间;t_i 表示第 i 个失效时间;N 是失效总数.根据修正后的数据可以得到相应的软件可靠性评估值,评估结果的准确性还有待进一步研究验证.

6 结束语

本文提出了一种基于试验设计的软件可靠性测试方法.方法在操作剖面的构造过程中应用正交试验设计控制操作数量,避免产生的操作数目过多需要再加以控制的现象,减少操作剖面的开发费用;在数据选取过程中应用组合设计优化操

作的运行分类集合,应用表明这样生成的可靠性测试数据更具针对性,能发现更多的软件缺陷.

参考文献(References)

[1] Cohen D M, Dalal S R, Kajla A, et al. The automatic efficient test generator (AETG) system[C]//Proceedings of the 5th ISSRE. Los Alamitos: IEEE Computer Society Press, 1994: 303－309

[2] Alam M S, Chen W H, Ehrlich W K, et al. Assessing software reliability performance under highly critical but infrequent event Occurrences[C]//Proceedings of the 8th ISSRE. Los Alamitos: IEEE Computer Society Press, 1997: 294－307

[3] 严进宝,徐锡山,钟读杭. 基于正交试验设计的软件可靠性测试[J]. 计算机工程, 2006, 32(4): 52－54
Yan Jinbao, Xu Xishan, Zhong Duhang. Software reliability test based on orthogonal experiment design[J]. Computer Engineering, 2006, 32(4): 52－54(in Chinese)

[4] 马长捷,朱小冬,袁野. 基于组合设计的软件可靠性测试用例设计方法[J]. 微计算机信息, 2006, 22(7－3): 263－265
Ma Changjie, Zhu Xiaodong, Yuan Ye. An approach to designing software reliability test case based on combinatorial design[J]. Microcomputer Information, 2006, 22(7－3): 263－265(in Chinese)

[5] Musa J D. Software reliability engineering[M]. Beijing: China Machine Press, 2003: 97－157

[6] Musa J D. Operational profile in software reliability engineering[J]. IEEE Software, 1993, 10(2): 14－32

[7] 艾骏,陆民燕,阮镰. 面向软件可靠性测试数据生成的剖面构造技术[J]. 计算机工程, 2006, 32(22): 7－9, 45
Ai Jun, Lu Minyan, Ruan Lian. Usage profile construction technique for generation of software reliability test data[J]. Computer Engineering, 2006, 32(22): 7－9, 45(in Chinese)

(责任编辑:娄　嘉)

(上接第 1374 页)

[14] Wittlin G, Rapaport M B. Naval rotorcraft water impact crash simulation using program KRASH[C]// American Helicopter Society 49th Annual Forum. MO, USA:[s. n.],1993:737－747

[15] Wittlin G, Smith M, Richards M. Airframe water impact analysis using a combined MSC/DYTRAN-DRI/KRASH approach [C]// American Helicopter Society 53rd Annual Forum. VA, USA:[s. n.], 1997:1138－1150

[16] Pentecote N, Vigliotti A. Crashworthiness of helicopters on water: test and simulation of a full-scale WG30 impacting on water[J]. International Journal of Crashworthiness, 2003,8(6):559－572

[17] Randhawa H S, Lankarani H M. Finite element analysis of impacts on water and its application to helicopter water landing and occupant safety [J]. International Journal of Crashworthiness, 2003,8(2):189－200

[18] Clarke C W, Shen Y. The development of a rotorcraft water impact methodology[C]// AHS/SAFE Crashworthiness Specialists Meeting on Crash Safety. Arizona, USA: [s. n.], 1998:23－27

[19] Sareen A K, Smith M R, Hashish E. Crash analysis of an energy absorbing subfloor during ground and water impacts [C]//American Helicopter Society 55th Annual Forum. Quebec, Canada:[s. n.],1999:1603－1612

[20] Wittlin G, Schultz M, Smith M. Rotary wing aircraft water impact test and analyses correlation[C]//American Helicopter Society 56th Annual Forum. VA, USA:[s. n.], 2000:1472－1485

[21] Ortiz R, Portemont G, Charles J L, et al. Assessment of explicit F E capabilities for full scale coupled fluid/structure aircraft ditching simulations[C]// 23rd International Congress of Aeronautical Sciences. Canada:[s. n.],2002:711.1－711.10

[22] Pentecote N, Kohlgruber D. Crash on water: a highly multi-physics problem[C]//14th European Conference and Exhibition on Digital Dimulation for Virtual Engineering. Paris, France:[s. n.], 2004:11－13

[23] Seddon C M, Moatamedi M. Review of water entry with applications to aerospace structures [J]. International of Impact Engineering, 2006, 31(7):1045－1067

[24] Sottorf W. Landing procedure in model ditching tests of Bf109[R]. NACA-Tm-1247,1945

[25] 叶天麒,周天孝. 航空结构有限元分析指南[M]. 北京:航空工业出版社,1996
Ye Tianlin, Zhou Tianxiao. FEM analysis guide for aeronautical structure[M]. Beijing: Press of Aviation Industry, 1996 (in Chinese)

[26] Edwin L F, Karen E J. Best practices for crash modeling and simulation[R]. NASA-TM-2002-211944, ARL-TR-2849, 2002

(责任编辑:刘登敏)