软件可靠性评估的控制论观点

蔡开元^{1*}, 孙昌爱², 聂长海³

1. 北京航空航天大学自动化科学与电气工程学院, 北京 100191

2. 北京科技大学计算机与通信工程学院, 北京 100083

3. 南京大学计算机科学与技术系, 南京 210023

* 通信作者. E-mail: kycai@buaa.edu.cn

收稿日期: 2019-06-11; 接受日期: 2019-09-02; 网络出版日期: 2019-11-13

国家自然科学基金 (批准号: 61872039) 资助项目

摘要 软件可靠性评估是确定软件能否交付使用的关键步骤. 针对软件可靠性评估中测试资源受限的问题, 本文阐述一种软件可靠性评估的控制论观点, 采用反馈与自适应控制策略指导软件测试过程, 介绍一种基于梯度下降方法的软件可靠性评估方法. 理论与实验结果表明, 基于软件控制论的可靠性评估方法能够在有限的测试资源下获得精度高、稳定性好的可靠性评估结果.

关键词 软件可靠性, 可靠性评估, 软件控制论, 软件测试, 适应性测试策略

1 软件可靠性评估

近年来, 信息技术日新月异, 从早期的科学计算已经发展为无处不在的计算. 与此同时, 软件规模急剧上升, 千万行级别的程序已经司空见惯 (如 Windows XP 系统、神舟飞船监控系统). 随着“软件定义世界”的兴起, 可以预见软件在计算机系统中地位与占比进一步提高, 软件规模还会进一步增加, 软件将成为人类社会的一种基础设施. 另一方面, 一些安全攸关的计算机系统的失效往往直接导致生命财产的重大损失. 最近的波音 737 MAX 客机由于飞行控制系统的升级软件存在缺陷导致了机毁人亡. 当软件渗透到人类生活的方方面面时, 如何评估软件能否按照期望的方式正确地运行 (从软件可靠性角度) 则显得尤为重要.

软件可靠性与软件中潜藏的缺陷数、软件规模、类型、开发环境、采用的软件开发方法学、开发人员的技术水平和管理水平等密切相关. 其中, 缺陷数直接影响被测软件的可靠性程度. 在软件交付之前, 通常要进行充分的测试. 软件测试的主要目的有两类. (1) 可靠性增长: 通过软件测试发现缺陷并剔除缺陷; (2) 可靠性评估: 通过测试确定其可靠性是否达到要求. 需要指出的是, 可靠性评估过程中被测软件代码被冻结, 不剔除发现的缺陷, 软件可靠性呈稳定状态.

引用格式: 蔡开元, 孙昌爱, 聂长海. 软件可靠性评估的控制论观点. 中国科学: 信息科学, 2019, 49: 1528–1531, doi: 10.1360/SSI-2019-0123
Cai K-Y, Sun C-A, Nie C-H. Software reliability assessment: a software cybernetics perspective (in Chinese). Sci Sin Inform, 2019, 49: 1528–1531, doi: 10.1360/SSI-2019-0123

图 1 (网络版彩图) 软件测试的控制论原理图

Figure 1 (Color online) Cybernetic principle of software testing

目前, 软件可靠性评估手段主要包括基于软件体系结构的方法、基于软件测试的方法、基于相似产品信息的方法 (例如神经网络、遗传算法、模糊系统等软计算方法). 这些评估方法存在如下不足之处: (i) 依赖于不同类型的信息 (软件需求、软件体系结构、软件测试数据、形式化验证信息、相似软件系统信息、开发人员经验等), 尚不存在能够统一量化这些信息的评估框架; (ii) 难于验证超高软件可靠性指标, 例如软件失效率是否低于 10^{-9} 次/小时; (iii) 缺乏严格的理论基础.

由于软件测试通常存在时间与资源方面的约束, 如何在有限的测试时间 (较少的测试) 内实现高效的 (精度高、稳定性好) 软件可靠性评估是一个挑战. 本文从控制论角度探讨软件可靠性评估问题, 将反馈控制和自适应控制等引入测试过程, 用于指导选择与测试用例, 提高软件可靠性评估效果.

2 基于控制论的软件可靠性评估

软件可靠性评估的控制论观点从一个新的角度看待软件测试问题, 认为软件测试问题是一个反馈控制问题. 如图 1 所示, 被测软件和测试策略构成一个闭环反馈控制系统, 其中被测软件称为被控对象, 测试策略称为控制器. 测试策略根据测试历史数据、被测软件的性质、当前状态、测试目标, 从测试用例库中选取后续测试用例, 优化软件测试过程. 简言之, 将软件测试问题转化为反馈控制问题, 软件可靠性评估的关键是寻找最优的测试策略.

具体地, 需要在无偏估计 (参数估计量的条件期望等于参数真实值) 的前提下, 获得最准确的软件可靠性评估结果 (即方差最小化). 首先假设: (1) 被测软件的代码被冻结; (2) 软件输入域根据分类准则划分为 m 个子域; (3) 被测软件运行剖面为 $\{(D_i, p_i), i = 1, 2, \dots, m\}$, 其中 p_i 表示从子域 D_i 中选择测试用例的概率; (4) 测试输出与测试历史无关; (5) 测试用例的执行要么成功要么失败, $P_r\{D_i \text{ 中的测试用例检测到缺陷}\} = \theta_i$. 根据这些假设, 软件可靠性为 $R = \sum_{i=1}^m p_i(1 - \theta_i)$. 测试完成后, 通过得到的测试数据进行可靠性的评估. 缺陷检测率 θ 的无偏估计为 $\hat{\theta}_i = \sum_{j=1}^{n_i} \frac{z_{ij}}{n_i}$, 其中 n_i 表示从子域 D_i 选择执行的测试用例数量, $z_{ij} = 1$ 表示子域 D_i 中第 j 个测试用例检测出缺陷, $z_{ij} = 0$ 表示子域 D_i 中第 j 个测试用例未检测出缺陷. 相应地, 软件可靠性的无偏估计为 $R_E = 1 - \sum_{i=1}^m p_i \hat{\theta}_i = 1 - \sum_{i=1}^m p_i \sum_{j=1}^{n_i} \frac{z_{ij}}{n_i}$. 由于 z_{ij} 实际上服从参数为 θ_i 的二项分布, 可靠性估计的均值为 $E(R_E) = 1 - \sum_{i=1}^m p_i \theta_i$, 满足无偏性; 可靠性估计的方差为 $\text{Var}(R_E) = \sum_{i=1}^m p_i^2 \frac{\theta_i(1-\theta_i)}{n_i}$, 即满足方差最小化的可靠性评估的优化目标函数.

上述优化目标函数, 为测试资源受限情形下如何有效分配测试用例提供了理论指导. 利用 Lagrange 乘子法求解, 不同子域所选取的测试用例数目 n_i 与 n_j 应满足如下关系: $\frac{n_i}{n_j} = \frac{\sqrt{p_i^2 \theta_i (1-\theta_i)}}{\sqrt{p_j^2 \theta_j (1-\theta_j)}}$. 测试过程中, θ_i 的真实值无法获知, 可以采用如下参数估计方法来估算. (i) 最小二乘法方法或遗传算法: 由于这两类方法都属于局部最优方法, 很难达到最优分配资源的效果; (ii) 梯度下降方法: 一种一阶求导的优化算法, 该方法也属于局部最优方法, 但该方法可以在保证局部最优的同时, 渐近收敛于全局最优. 为此, 我们提出了一种利用梯度下降方法来进行可靠性评估的适应性测试策略 (AT-GD)^[1], 从理论与实验两个方面, 验证基于 AT-GD 测试策略得到的软件可靠性估计结果满足无偏估计的要求, 并且渐进收敛于软件可靠性的真实值.

上述可靠性评估的适应性测试策略能够在有限的测试资源下获得精度高、稳定性好的可靠性评估结果. 该研究成果不仅充实了软件可靠性评估的理论基础, 还为软件控制论研究提供一个理论与实验相统一的案例.

3 软件控制论展望

软件控制论旨在探讨计算机软件与控制领域的跨学科问题. 其核心科学问题包括如何建立软件行为控制模型、设计方法和控制理论, 以及有效、量化的控制策略. 具体说来, 关注 4 方面的问题: (i) 形式化和量化刻画软件行为的反馈和自适应控制机制; (ii) 将控制原理和方法用于改进软件开发过程和控制软件系统运行行为; (iii) 将软件工程原理和理论应用于控制系统设计与综合; (iv) 为软件工程和工程建立统一的理论框架.

计算的本质是一种控制过程^[2], 无论是串行计算或并发计算都存在控制机理. 实现计算的计算机软件通常表现为服务, 其本质也是一种控制系统. 软件开发是一种基于构造的工程化活动, 通常存在多个重复迭代的步骤, 产生不同类型的软件制品 (包括需求、设计、代码等). 软件控制论将上述不同类型的开发活动与软件制品纳入研究范畴, 探讨与实施有效的控制策略, 为有效实现软件工程追求的三大目标 (保证软件质量、确保开发进度、控制开发费用) 提供一种新思路. 近年来, 人们开始尝试将软件控制论应用于需求分析、软件测试等软件开发活动^[3~5], 以及 Web 服务质量保证、云计算资源分配、智能手机软件重生等问题, 并取得了良好的应用效果^[6]. 综上, 我们认为软件控制论提供了一种新的软件开发范型, 将为软件工程的发展做出实质性贡献.

参考文献

- 1 Lv J P, Yin B B, Cai K-Y. On the asymptotic behavior of adaptive testing strategy for software reliability assessment. *IEEE Trans Softw Eng*, 2014, 40: 396-412
- 2 Cai K-Y. Software cybernetics. In: *Control Sciences*. Beijing: Science Press, 2015. 567-575 [蔡开元. 软件控制论. 见: 控制科学. 北京: 科学出版社, 2015. 567-575]
- 3 Sun C-A, Dai H P, Liu H, et al. Adaptive partition testing. *IEEE Trans Comput*, 2019, 68: 157-169
- 4 Bertolino A, Miranda B, Pietrantuono R, et al. Adaptive test case allocation, selection and generation using coverage spectrum and operational profile. *IEEE Trans Softw Eng*, 2019. doi: 10.1109/TSE.2019.2906187
- 5 Nie C H, Leung H, Cai K-Y. Adaptive combinatorial testing. In: *Proceedings of the 13th International Conference on Quality Software (QSIC 2013)*, 2013. 284-287
- 6 Yang H J, Chen F, Aliyu S. Modern software cybernetics: new trends. *J Syst Softw*, 2017, 124: 169-186

Software reliability assessment: a software cybernetics perspective

Kai-Yuan CAI^{1*}, Chang-Ai SUN² & Changhai NIE³

1. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China;

2. School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China;

3. Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China

* Corresponding author. E-mail: kycai@buaa.edu.cn

Abstract Software reliability assessment is a key step that determines whether the software can be delivered for the use or not. Because only limited testing resources are allowed for software reliability assessment, this paper proposes a software cybernetics perspective that introduces feedback and self-adaptive control to guide the software testing process in the software reliability assessment. Also, an adaptive testing strategy based on gradient descent is proposed for software reliability assessment. The theoretical and experimental results show that reliability assessment methods based on software cybernetics can achieve highly precise and stable software-reliability-assessment results using restricted testing resources.

Keywords software reliability, reliability assessment, software cybernetics, software testing, adaptive testing strategy

Kai-Yuan CAI received his B.S., M.S., and Ph.D. degrees from Beihang University (Beijing University of Aeronautics and Astronautics), Beijing, China, in 1984, 1987, and 1991, respectively. He has been a full professor at Beihang University since 1995. He is a Cheung Kong Scholar (chair professor), jointly appointed by the Ministry of Education of China and the Li Ka Shing Foundation of Hong Kong in 1999. His main research interests include software testing, software reliability, reliable flight control, and software cybernetics.

ing, software reliability, reliable flight control, and software cybernetics.

Chang-Ai SUN received his B.S. degree in computer science from the University of Science and Technology Beijing, China, and the Ph.D. degree in computer science from Beihang University, China. He is a professor at the School of Computer and Communication Engineering, University of Science and Technology Beijing. His research interests include software testing, program analysis, and service-

oriented computing.

Changhai NIE received his B.S. and M.S. degrees in mathematics from Harbin Institute of Technology, Harbin, China, and his Ph.D. degree in computer science from Southeast University, Nanjing, China. He is a professor of software engineering at the State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing. His research interests include software analysis, testing, and debugging.

bugging.