基于模糊层次分析法的测试效果量化预测评估方法*)

陆瀛海　杨宗源

(华东师范大学计算机科学技术系　上海 200062)

摘　要　本文提出了一种基于模糊层次分析法的测试效果定量预测评估方法。将测试用例使用测试用例优先级技术进行优先级标定，对测试用例进行层次化管理，在此基础上对不同测试用例赋予不同的预期测试时间，从而计算出预期的测试效果并预测不同测试时间赋予方案所能发现软件缺陷的相对能力。基于此设想实现了的工具 TPMT，在实际的软件测试中的初步试验表明，这种方法得到的测试效果预测值和实际发现软件缺陷的数量比例是一致的，所得的测试效果预测值具有较高可信度。

关键词　层次分析法，测试用例优先级，定量分析

A Fuzzy Analytical Hierarchy Process Based Quantitative Test Effect Evaluation Methodology

LU Ying-Hai　YANG Zong-Yuan

(Department of Computer Science and Technology, East China Normal University, Shanghai 200062)

Abstract　This paper proposes an analytical hierarchy process based numerical test effect evaluation methodology. It can evaluate the test effect of different test time allocation strategies and predict the relative ability of defect finding by using test case priority technique, arranging the test cases hierarchically and assign different test time to these test cases. Based on this methodology, a tool named TPMT is implemented, the application to the real software testing process reveals that the evaluation value produced by this methodology is consistent with the ratio of defects found, thus its evaluation value is reliable.

Keywords　Analytical hierarchy process, Test case priority, Quantitative method

在软件测试场景中，测试人员往往将记录测试用例，以备以后的测试中使用。但是，在回归测试中无侧重地运行所有的这些测试需要耗费大量的资源，特别是时间。例如在一个包含 20,000 行代码的软件中，运行全部的测试用例集需要 7 个星期[1]，所以，研究人员提出了多种不同的技术[4]来减少回归测试的花费，这些技术包括了测试用例的优先级技术[1,4~10]。它允许测试人员按照某种标准对测试用例的优先级排序，这样，在回归测试中，优先级高的测试用例就能先于优先级低的测试用例执行或者要求获得更多的测试资源。从而降低发现软件缺陷所需要的时间，提高测试的效率。

但是各种不同的测试技术中同时存在的一个问题是，对于一种优先级标定方案和测试时间分配方案只能模糊地估计所能获得的测试效果，没有一种定量的方法计算测试效果值。而在回归测试阶段，对不同的测试时间分配方案所能得到的测试效果的定量评估对于制定正确的测试方案，合理分配测试资源，避免盲目的，对所有测试用例评价力度的运行，从而达到更好的测试效果有着十分重要的作用。本文中提出了一种对回归测试中量化预测评估效果的方法解决了这个问题。

1　量化预测评估测试效果

测试用例优先级问题可以形式化地定义为：

定义 1(测试用例优先级问题[1])　给定：测试集 T，T 的一个重新排列 PT，一个从 PT 到实数的一个映射

问题：寻找一个 $T'\in PT$ 使得 $(\forall T'')(T''\in PT)(T''\neq T')[f(T')\geqslant f(T'')]$ 在这个定义中，PT 代表所有 T 的可能优先级排序，f 是可以应用于该排序上的一个函数，通过该函数可以求得该排序的测试效果值(不失一般性，我们假定，高的效果值代表了高的测试有效性)[1]。

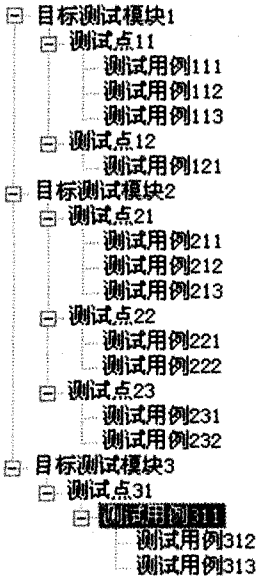

图 1　层次化的测试用例

*)本文获得博士学科点专项基金项目资助，基金项目编号：20060269002。**陆瀛海**　硕士研究生，主要研究领域为 CBSD，软件测试；**杨宗源**　教授，主要研究领域为软件工程、工具及环境、形式化、面向对象、组件、中间件、分布计算、过程管理、测试与度量、语言处理。

要想对测试活动的效果能有一个量化的评价方法，就需要对测试中使用的单个测试用例能够有一个量化的评价。在研究过程中，作者发现测试用例的优先级在很大程度上反映了测试用例的重要程度。如果按照对软件缺陷的检测能力大小对测试用例进行优先级排序，这样的排序就反映了测试用例的重要程度。

同时还注意到，目前的测试用例优先级技术中，测试用例可以通过层次化的方法进行组织。一个大规模的软件，往往包含了多个测试模块，每个测试模块又包含了多个测试点，这样测试用例呈献一种层次化的关系，如图 1 所示。

在测试回归测试开始之前，如何根据现有的测试用例的优先级来评估该总优先级排序方法所能达到的效果值，即对应于某个 T'，如何得到定量的 $f(T')$。这对于包含大量目标测试模块和测试点的软件的回归测试具有十分重大的意义。

2 确定测试用例权重的方法原理以及确定步骤

假定对于每个测试用例，已经对它们运用测试用例优先级技术确定了一个优先级值(1：不重要～9：重要)[10~12,14]。探讨测试用例关系及其程度就是为了确定测试用例与其影响因素间的关系矩阵，即建立“本构关系”，从理论上讲，本构关系矩阵的确定有许多方法，如层次分析法、运筹法、计量方法、多元回归法、模糊评价法和灰色关联法等均可采用。本文所采用的是美国著名运筹学家皮兹堡大学教授 T．L．Saaty 于 20 世纪中期提出的层次分析法(Analytic Hierarchy Process，简称 AHP)，它对各个平均因素相互间的重要度，通过两两比较后，再进行特征根计算，在允许的相容性范围内，按照中和重要度排出其评价顺序，两两比较是对若干因素的大小或强弱关系进行比较，比较值一般取 1～9 的整数或倒数[2]。

该方法被普遍认为是操作简便而有效的多目标决策方法，以能将定性与定量因素同时处理而得到广泛应用。

AHP 确定指标权重的方法是从两两比较矩阵求解权重，当然也可以使用其他方法如对数最小二乘法[17]。应用 1～9 标度方法建立因素间的两两判别矩阵 $P=(p_{ij})_{n\times n}$，其中，用例 p_{ij} 具有：(1) $p_{ij}>0$；(2) $p_{ij}=1/p_{ji}$；(3) $p_{ii}=1$。通过判别矩阵计算某级指标对相应上级指标的影响程度(权重)，从而确定各指标对应评价目标的影响权重。1～9 标度的含义见表 1。

表 1　1～9 标度的含义

标值	意义
1	表示两个用例相比，具有同样重要性
3	表示两个用例相比，一个用例比另一个用例稍微重要
5	表示两个用例相比，一个用例比另一个用例明显重要
7	表示两个用例相比，一个用例比另一个用例强烈重要
9	表示两个用例相比，一个用例比另一个用例极端重要
2、4、6、8	上述相邻判断的中值

求解判断矩阵 P 的特征根，找出最大特征根 λ_{max} 及其对应的特征向量 W，即得到同一层各指标相对于上一层某指标的相对重要性的权重排序。

利用 T．L．Saaty 的平均随机一致性指标对于判断矩阵 P 进行一致性检验(各平均一致性随机指标 RI 见表 2)；由一致性指数 $CI=(\lambda_{max}-n)/(n-1)$ 和 RI 可以通过计算得到随机一致性比率 $CR=CI/RI$，若 $CR<0.1$，则认为 P 具有满意的一致性，否则必须重新调整 P，直到具有满意的一致性。

表 2　判断矩阵的随机一致性指标

n	1	2	3	4	5	6
RI	0	0	.58	.96	1.12	1.24
n		7	8	9	10	11
RI		1.32	1.41	1.45	1.49	1.51

假设一个评价指标体系结构如图 2，则该评价指标体系权重确定步骤如表 3～表 6。

表 3　目标层判断矩阵表

A	B	C	D	ω_A	一致性检验
B	1	3	5	0.634	$\lambda_{max}=3.037$
C	1/3	1	3	0.260	$CI=0.019$
D	1/5	1/3	1	0.106	$CR=0.037<0.1$ 一致性满足

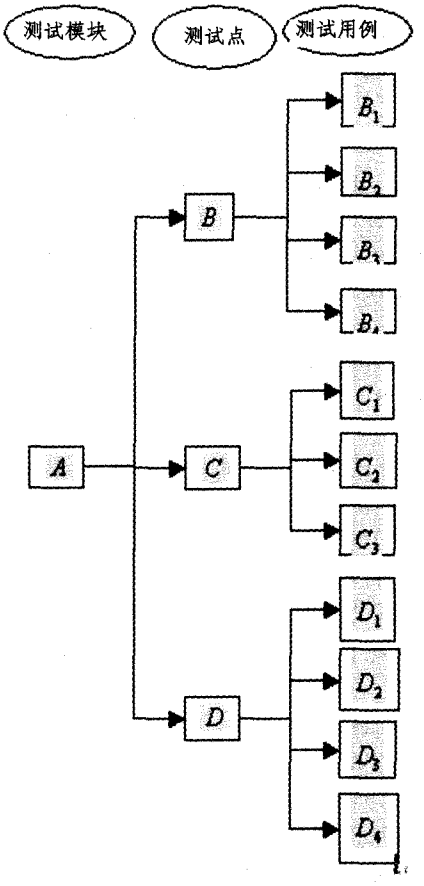

图 2　测试用例优先级体系 Example_I 示意

表 4　测试点指标层 B 判断矩阵表

B	B_1	B_2	B_3	B_4	ω_B	一致性检验
B_1	1	3	5	3	0.489	$\lambda_{max}=4.059$
B_2	1/3	1	3	1	0.217	$CI=0.019$
B_3	1/5	1/3	1	1/3	0.076	$CR=0.022<0.1$
B_4	1/3	1	3	1	0.217	一致性满足

表 5　测试点指标层 C 判断矩阵表

C	C_1	C_2	C_3	ω_C	一致性检验
C_1	1	2	2	0.500	$\lambda_{max}=3$　$CI=0$
C_2	1/2	1	1	0.250	$CR=0$　完全一致
C_3	1/2	1	1	0.250	

表 6　测试点指标层 D 判断矩阵表

D	D_1	D_2	D_3	D_4	ω_D	一致性检验
D_1	1	4	1	3	.403	$\lambda_{max}=4.117$
D_2	1/4	1	1/2	1/2	.109	$CI=0.039$
D_3	1	2	1	3	.339	$CR=0.043<0.1$
D_4	1/3	2	1/3	1	.149	一致性满足

3　测试用例评估体系的综合评价过程

采用二级综合评价方法对评价指标体系 Example_I 进行评价，相应的数学描述如下：

一层指标分别记作 $V_1=\{V_{11},V_{12},V_{13}\}$，$V_2=\{V_{21},V_{22},V_{23},V_{24}\}$，$V_3=\{V_{31},V_{32},V_{33},V_{34}\}$；其中 V_{ij} 表示一层第 i 个指标对应的二层中的第 j 个指标。

为操作简单，测试所用时间等级记作 $T=\{T_1,T_2,T_3,T_4,T_5\}$，各等级分别为：$T_1$——多；$T_2$——较多；$T_3$——一般；$T_4$——较少；$T_5$——少。

根据不同回归测试的具体情况，可以划分更多的等级达到更高的精确度。同时将测试所用的时间段划分到这些等级中。

对于任意指标的评价记作 U_i。评价模型为(V,T,U)。

针对一层不同指标建立相应评价矩阵 U_i，根据已建立的评价等级，通过结合预期测试花费时间确定测试用例的花费时间等级见表 7。

表 7　测试点指标层 D 的单因素评分

花费时间等级	D_1	D_2	D_3	D_4
权重	0.403	0.109	0.339	0.149
多	6	4	8	4
较多	4	6	6	6
一般	6	6	4	4
较少	2	4	2	4
少	2	0	0	2

这样测试目标模块层 D 的评价矩阵为：

$$U_D=\begin{bmatrix}0.6&0.4&0.6&0.2&0.2\\0.4&0.6&0.6&0.4&0.0\\0.8&0.6&0.4&0.2&0.0\\0.4&0.6&0.4&0.4&0.2\end{bmatrix}$$

同理可以建立

$$U_B=\begin{bmatrix}0.6&0.8&0.4&0.2&0.0\\0.4&0.6&0.4&0.4&0.2\\0.4&0.6&0.6&0.4&0.0\\0.6&0.4&0.4&0.4&0.2\end{bmatrix}$$

$$U_C=\begin{bmatrix}0.6&0.6&0.4&0.4&0.0\\0.4&0.8&0.4&0.2&0.2\\0.8&0.4&0.4&0.4&0.0\end{bmatrix}$$

第一级综合评价：对测试目标模块每一个指标 V_j 分别进行按一级模型进行评价，获得该指标的一级评价结果矩阵 I^*，算法为：

$I^*=\omega_i U_i$；接上述算例，则

$D^*=\omega_D U_D=(0.403,0.109,0.339,0.149)$

$$\begin{bmatrix}0.6&0.4&0.6&0.2&0.2\\0.4&0.6&0.6&0.4&0.0\\0.8&0.6&0.4&0.2&0.0\\0.4&0.6&0.4&0.4&0.2\end{bmatrix}$$

$=(0.616,0.519,0.502,0.213,0.110)$

同理

$B^*=\omega_B U_B=(0.489,0.217,0.076,0.217)$

$$\begin{bmatrix}0.6&0.8&0.4&0.2&0.0\\0.4&0.6&0.4&0.4&0.2\\0.4&0.6&0.6&0.4&0.0\\0.6&0.4&0.4&0.4&0.2\end{bmatrix}$$

$=(0.541,0.654,0.415,0.302,0.087)$

$C^*=\omega_C U_C=(0.500,0.250,0.250)$

$$\begin{bmatrix}0.6&0.6&0.4&0.4&0.0\\0.4&0.8&0.4&0.2&0.2\\0.8&0.4&0.4&0.4&0.0\end{bmatrix}$$

$=(0.600,0.600,0.400,0.315,0.125)$

以一级指标层评价获得的各指标结果矩阵 I^* 构造目标层综合评价的单因素评价矩阵 U_A，获得最终综合评价结果 A^*。

$$U_A=\begin{bmatrix}B^*\\C^*\\D^*\end{bmatrix};$$

$$=\begin{bmatrix}0.541&0.654&0.415&0.302&0.087\\0.600&0.600&0.400&0.315&0.125\\0.616&0.519&0.502&0.213&0.110\end{bmatrix}$$

权重 $\omega_A=(0.634,0.260,0.106)$；

最终综合评价结果为：

$A^*=\omega_A U_A=(0.634,0.260,0.106)$

$$\begin{bmatrix}0.541&0.654&0.415&0.302&0.087\\0.600&0.600&0.400&0.315&0.125\\0.616&0.519&0.502&0.213&0.110\end{bmatrix}$$

$=(0.564,0.626,0.420,0.296,0.099)$

最终得到的是一个评价等级向量。根据最大隶属原则可以判断测试方案所能达到的测试效果。

和传统的测试方案评估方法相比，基于 FAHP 的量化评估方法有以下的主要特点：(1)传统的评估方法没有使用测试用例优先级方法，故而不能量化衡量测试用例的重要程度，也就没有量化评价的基础；(2)在使用测试用例重要性权值的评估方法以绝对值判断重要度。在测试点和测试用例多时评判标准很难把握。FAHP 仅取两项进行两两比较，操作简单，同时评估精度可以通过计算相容度进行修正；(3)和传统的评估方法使用固定的资源配置权值相比，FAHP 允许对用于测试用例的时间进行不同精确度的划分，通过对于给定的花费时间区间划分成不同的等级，可以方便地调整计算精度；(4)使用改进的 FAHP 方法具有单调的模糊测度，并且以最大值为 1 规范话，能够缓和权重计算机的加法性；(5)当重要度相同或接近时，传统的方法无法确切地把握其相互关系，而 FAHP 可以使用模糊隶属函数进行更精细的分析评价[2]。

4　应用实例

根据本文上述的思想，我们实现了用于回归测试中测试效果预测评估的工具(Test cases Priority Management Tool in regression test, TPMT)[14~16]。并且成功地应用于 Honeywell 亚太研发中心 Da-Vinci 门禁系统配置软件的测试中。

Honeywell Da-Vinci 门禁系统配置软件是面向银行，机场等安全敏感部门开发的就有高安全性，高可扩展性，高可配

(下转第 297 页)

间为3；而 t_4^* 和 t_6^* 的引发周期为7，所以 t_4 每隔7个时间单位引发1次(引发速率为1)且持续的时间为3，因此库所 p_3 的标识在任意时刻都不会是负增长(事实上 $\forall t>0$，$\lfloor t/5 \rfloor \times 3 \times 3 \geqslant \lfloor t/7 \rfloor \times 3 \times 1$，其中 t 表示时间)，即系统是不安全的。

结论 本文给出了流体随机Petri网(FSPN)转换成一阶混杂Petri网(FOHPN)的形式化描述方法从而能够借用FOHPN的建模原语和分析方法来分析FSPN。实例表明，通过把FSPN转换成FOHPN后，可利用演变图来分析FSPN从而克服了FSPN数值分析方法不能实现系统有无死锁、系统是否安全等等的局限性。进一步把FOHPN转换成混杂自动机可借用混杂自动机的验证技术和工具来实现FSPN的验证，目前作者正在对这一工作进行探讨。

参考文献

1 Becker M, Bessey T. Comparison of the Modeling Power of Fluid Stochastic Petri nets and Hybrid Petri Nets [C]. IEEE International Conference on Systems , Man and Cybernetics (SMC), 2002

2 Horton G, Kulkarni V G. Fluid Stochastic Petri nets: Theory, Applications and Solution [J]. European Journal of Operation Research, 1998,105 (1):184～201

3 Gribaudo M, Sereno M, Bobbio A. Fluid Stochastic Petri nets: An Extended Formalism to Include Non-Markovian Models [C]. In: Proceedings of PNPM'99, Zaragoza, Spain, 1999. 74～82

4 Wolter K. Second Order Fluid Stochastic Petri nets: An Extension of GSPNs for Approximate and Continuous Modeling [C]. In: Proceedings of WCSS'97, Singapore, Sept. 1997. 328～332

5 David R. Modeling of hybrid systems using continuous and hybrid Petri nets [C].. In: Proceedings of PNPM '97, St-Malo, France, June 1997. 47～57

6 Balduzzi F, Giua A, Menga G. First-Order Hybrid Petri Nets: a Model for Optimization and Control [J]. IEEE Transactions On Robotics And Automation, 2000, 16(14):382～399

7 卢光松，葛运建. 流体随机Petri网与混合Petri网的比较分析[J]. 小型微型计算机系统，2005，26 (12)：2144～2146

(上接第287页)

置性的门禁管理系统。

在对规则配置软件的回归测试中，对5个模块，20个测试点，407个测试用例进行优先级标定，在经过第一轮的测试用例运行过程后对这些优先级进行进一步修正。所得的优先级进入TPMT工具中，TPMT工具中动态生成对测试用例的优先级重要性的矩阵。在测试计划的评估阶段，对每个测试用例分配预计使用的测试时间，预测测试方案所能获得的不同测试效果。同时TPMT工具对于所有测试模块，测试点，测试用例的任意子集同样可是使用本文提到的方法对这一子集进行测试效果的预测。我们对于所有5个模块的测试用例通过选取其中所有测试模块，测试点和测试用例(方案1)，选取其中两个测试模块，5个测试点，34个测试用例(方案2)，其中3个测试模块，13个测试点，286个测试用例(方案3)进行分别使用不同测试时间的对照试验，试验表明，所获得的测试效果和预测的测试效果是一致的。如表8所示。

表8 在Da-Vinci门禁配置软件中测试效果预测值和实际发现的缺陷效果

方案	测试效果预测值	实际发现缺陷数/测试用例集合最终发现的缺陷总数
1	6.5	90/145
2	8.1	33/40
3	7.2	50/91

可以看到，测试效果预测值和实际发现的缺陷数/测试用例集合最终发现的缺陷总数基本上成线性关系。

结束语 本文基于测试用例优先级技术，提出了一种在回归测试中对测试用例使用不同测试强度所能达到的测试效果进行度量和预测的方法。它解决了在测试用例优先级技术中量化计算测试效果值f的方法。通过在Da-Vinci门禁系统配置软件中的应用表明，该方法为回归测试中，对测试努力所能达到的测试效果的评估提供了有效的途径。我们实现的TPMT工具为测试用例优先级跟踪管理和评估提供了有力的支持。

在Da-Vinci门禁配置软件测试过程中对本方法的实际使用中，初步的试验表明，通过本方法达到的预测效果值和最终的实际发现软件缺陷数是基本一致的。当然本文提到的方法不仅可以使用于软件中，对于测试用例可以标定优先级，同时测试用例可以划分为模块，测试点，测试用例这样的层次结构的软，硬件测试，均可以使用本方法。

参考文献

1 Rothermel G, Untch R H, et al. Prioritizing Test Cases For Regression Testing [J]. IEEE Transactions on Software Engineering

2 熊伟，新藤久和，渡边喜道. 软件需求定量分析及其映射的模糊层次分析法[J]. 软件学报，2005,16(3)：427～433

3 黄显勇，毛明海. 运用层次分析法对水利旅游资源进行定量评价[J]. 浙江大学学报(自然科学版)，2001,28(3):327～332

4 Harrold M J, Gupta R, Soffa M. A Methodology for Controlling the Size of a Test Suite. ACM Transactions on Software Engineering and Methodology [J], 1993,2(3):270～285

5 Gupta R, Harrold M J, Soffa M L. An approach to regression testing using slicing [J]. In: Proc. of the IEEE Conf. on Software Maintenance, 1992. 299～308

6 Harman M, Danicic S. Using program slicing to simplify testing. Software Testing [J]. Verification and Reliability,1995,5:143～162

7 Kamkar M, Fritzson P, Shahmehri N. Interprocedural dynamic slicing applied to interprocedural data flow testing [J]. In : Proc. of the Conf. on Software Maintenance -93,1993. 386～395

8 Bates S, Horwitz S. Incremental program testing using program dependence graphs [J]. In : Conf. Record of the Twentieth ACM *Symposium on Principles of Programming Languages*, ACM, 1993

9 Offutt A J, et al. VOAS. Procedures for Reducing the Size of Coverage-based Test Sets [J]. In: Twelfth International Conference on Testing Computer Software, June 1995. 111～123

10 Pyhala T, Hanko K. Specification Coverage Aided Test Selection [J]

11 Wong W, Horgan J, London S, et al. A study of effective regression in practice [J]. In: Proceedings of the Eighth International Symposium on Software Reliability Engineering, November 1997. 230～238

12 Wong W, Horgan J, London S, et al. Effect of test set size and block coverage on the fault detection effectiveness [J]. In: Proceedings of the Fifth International Symposium on Software Reliability Engineering, Nov. 1994. 230～238

13 Wong W, Horgan J, London S, et al. Effect of test set minimization on fault detection effectiveness [J]. In: Proceedings of the 17th Internationl Conference on Software Engineering, Apr. 1995. 41～50

14 Watkins D. Fundamentals of Matrix Computations [M]. Wiley, 2002

15 孙兰芬，陈一巾 . 线性代数[M]. 浙江大学出版社，1994

16 徐士良 . 计算机常用算法[M](第二版). 北京：清华大学出版社，1995