

# 基于 AHP 的回归测试用例优先级排序方法

冯沈峰 高建华

(上海师范大学信息与机电工程学院 上海 200234)

**摘要** 测试用例优先级排序方法是基于特定准则对测试用例进行排序以提高回归测试效率。现有方法局限于单目标或少数影响因素,影响了其对测试用例的综合分析和评价,对此,文中提出了一种基于层次分析法的回归测试用例优先级排序方法。该方法以优化测试用例序列为目标,以影响因素为准则,以测试用例为方案,建立了层次结构模型,构造了对比矩阵,进行了排序与一致性检验,优化了影响因素比例。实验采用 Matlab 软件环境,以平均故障检测率为评估指标。结果表明,与已有几种优先级方法相比,该方法能达到更高的平均故障检测率 85%,提高了测试效率,此外可根据实际需求扩充影响因素数量,具有灵活性。

**关键词** 测试用例优先级,影响因素,层次分析法,平均故障检测率

中图分类号 TP311 文献标识码 A DOI 10.11896/j.issn.1002-137X.2019.08.038

## Test Case Prioritization Method Based on AHP for Regression Testing

FENG Shen-feng GAO Jian-hua

(The College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China)

**Abstract** Test case prioritization methods are based on specific criteria to sort test cases to improve the test efficiency. Considering that the existing techniques are limited to single objective or a few influencing factors, which affect the comprehensive analysis and evaluation of test cases, this paper proposed a test case prioritization method based on analytic hierarchy process. This method aims at optimizing test case sequence, takes the influencing factors as the criterion, and takes the test cases as schemes. It constructs hierarchical structure model and judgment matrices. Lastly, it sorts the test cases, carries out the consistency check, and optimizes the ratio of influencing factors. The experiment uses Matlab software and the APFD as the metric to evaluate. Experimental results show that compared with other existing prioritization methods, this method achieves higher APFD value of 85% and improves the test efficiency. In addition, according to actual requirements, it increases the number of influencing factors, so that it can be flexible.

**Keywords** Test case prioritization, Influencing factors, Analytic hierarchy process, APFD

## 1 引言

回归测试是软件生命周期中一项频繁进行且开销巨大的过程。统计数据表明,回归测试一般占软件测试预算的 80% 以上,占软件维护预算的 50% 以上<sup>[1]</sup>。为了降低这部分开销,提高回归测试中测试用例的使用效率,常对测试用例进行优化,常用方法包括:失效测试用例的识别和修复、测试用例选择、测试用例优先级排序、测试用例集缩减和测试用例集扩充等<sup>[2-3]</sup>。

测试用例优先级排序方法通过设定特定的排序准则,对测试用例进行排序以优化其执行次序,旨在最优化排序目标。回归测试中存在多种会影响测试用例优先级排序效果的因素,这些影响因素可以作为测试优先级排序的准则,如语句、分支覆盖率、测试费用、需求、缺陷严重性等。

针对回归测试用例优先级排序的研究工作最早可以追溯到

到 Wong 等在 1997 年发表的论文<sup>[4]</sup>,目前已有较多回归测试用例优先级排序方法被提出。Rothermel 等<sup>[5]</sup>和 Elbaum 等<sup>[6]</sup>提出了 APFD(平均故障检测率)度量,并将其作为测试用例优化评价标准,采用优先级技术(如语句覆盖、功能覆盖、分支覆盖、缺陷覆盖等)来提高缺陷的早期检测速率。Maheswari 等<sup>[7]</sup>提出了一种基于汉明距离的方法来优化测试用例序列,即用二进制的形式来表示每个测试用例中缺陷分布的情况,通过优化尽可能在测试早期测试用例达到最大的缺陷覆盖。上述两种方法在测试用例排序时仅考虑了各种类型的覆盖,缺乏对测试用例的执行开销和缺陷的危害程度的考虑。当测试用例具有相同的最大覆盖率时,随机选择会影响排序效果。为了解决此问题,Wang 等<sup>[8]</sup>结合贪婪算法提出了一种基于缺陷严重性的测试用例优化方法。Kavitha 等<sup>[9]</sup>提出了一种考虑缺陷检测率和缺陷严重性两个因素的测试用例优化方法,以尽可能在测试早期检测出严重性较高的缺陷。

到稿日期:2018-07-31 返修日期:2018-10-13 本文受国家自然科学基金项目(61672355)资助。

冯沈峰(1994-),男,硕士生,主要研究方向为软件可靠性设计;高建华(1963-),男,博士,教授,CCF 会员,主要研究方向为软件可靠性理论与设计、软件开发环境与开发技术、数据安全与计算机安全、网络测试、LSI/VLSI 测试等,E-mail:jhgao@shnu.edu.cn(通信作者)。

Tyagi等<sup>[10]</sup>在Kavitha的方法上更改了严重性检测能力的计算方法,并将缺陷检测比纳入考虑范围。上述3种方法考虑了缺陷严重性、执行开销,但在多影响因素情况下排序的效果并非最优,且在一定程度上缺乏灵活性。Mahmood等<sup>[11]</sup>通过考虑缺陷、需求、软件复杂性、测试人员、历史信息等16个影响因素,以相同权值计算排序值得到优化序列,虽然此方法考虑了软件生命周期中对回归测试的众多影响因素,但分配权值相等并不一定满足实际回归测试需求。

对此,本文提出了一种基于AHP的回归测试用例优先级排序方法,其以优化测试用例序列为目标,根据多个影响因素、多个测试用例建立层次结构模型,可扩充影响因素的数量并为各影响因素分配不同的权值,从而满足回归测试需求,优化影响因素比例。本文采用Matlab求解模型,得到最优测试用例序列。

本文第2节介绍优先级排序及层次分析法的相关术语;第3节将层次分析法引入测试用例优化中,建立完整的层次结构模型,还设计了完整的基于层次分析法的测试用例优化方法;第4节通过两组实验进行对比分析;最后,总结全文并介绍下一步工作。

## 2 基本概念和术语

### 2.1 测试用例优先级排序

测试用例优先级是指根据一定的排序准则将测试用例集进行排序,使满足测试要求的测试用例较先被执行<sup>[12]</sup>。不同的测试用例具有不同的特性,如代码覆盖率、缺陷检测率、需求等,若按照一定的准则对测试用例排序,提高覆盖率、检测率较高的测试用例的优先级,则有利于提高回归测试的效率并取得较好的测试效果。

Rothermel等<sup>[13]</sup>于2001年将测试用例优先级排序问题形式化描述为:给定测试用例集 $T$ , $T$ 的全排列集为 $PT$ ,排序目标函数为 $f$ 。 $f$ 的定义域为 $PT$ ,值域为实数。测试用例优先级排序的目的是找出 $T' \in PT$ ,使得:

$$(\forall T'')(T'' \in PT)(T'' \neq T')[f(T') \geq f(T'')]$$

由上述描述可知,集合 $PT$ 包含了集合 $T$ 中所有可能的测试用例排序;函数 $f$ 用来定量描述评价排序的效果, $f$ 值越大,测试用例的排序效果越好。

### 2.2 APFD

软件测试的目的是设计出高质量的测试用例来尽可能多地检测出软件中存在的缺陷。为了度量测试用例排序结果检测缺陷的能力,Rothermel等<sup>[5]</sup>首先提出了平均故障检测率(Average Percentage of Fault Detection,APFD)评测指标,其计算公式如下:

$$APFD = 1 - \frac{TF_1 + TF_2 + TF_3 + \dots + TF_m}{mn} + \frac{1}{2n} \quad (1)$$

其中, $n$ 是测试用例集 $T$ 中测试用例的数量, $m$ 是被测软件的缺陷总数量, $TF_i$ 表示排序后测试用例集 $T'$ 中首次检测到缺陷 $i$ 的测试用例在该执行次序中的次序。APFD的值域为 $[0,1]$ ,排序在前的测试用例检测出的缺陷越多,则APFD越高,缺陷检测速度越快,测试集的排序效果越好。

### 2.3 层次分析法

层次分析法的基本思想<sup>[14]</sup>是将待分析问题层次化,根据

问题的性质和要达成的目标,将问题分解为不同的组成因素,并按照这些因素间的关联影响及其隶属关系,将因素按不同的层次进行聚集组合,形成一个多层次的分析结构模型。最后对问题进行优劣比较并排序。层次分析法有4个基本步骤<sup>[15]</sup>:1)建立层次结构模型;2)构造对比矩阵;3)层次单排序及其一致性检验;4)层次总排序及其一致性检验。

#### 2.3.1 层次结构模型

应用层次分析法分析决策问题时,首先要把问题条理化、层次化,以构造出一个有层次的结构模型。如图1所示,该模型一般分为3层:1)目标层,一般为分析问题的预定目标或理想结果;2)准则层,包括所考虑问题的准则、子准则;3)方案层,包括为实现目标可供选择的各种措施、决策方案等。

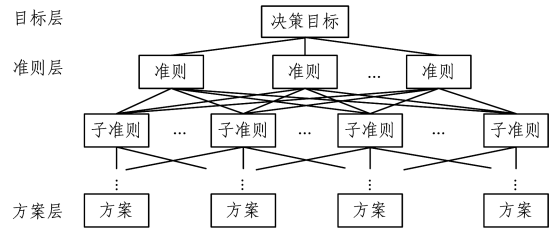


图1 层次结构模型

Fig.1 Hierarchical structure model

#### 2.3.2 对比矩阵

对比矩阵,又称判断矩阵,表示所有元素针对上一层某一个元素的相对重要性的比较。假定要比较 $n$ 个因子 $W = \{\omega_1, \omega_2, \dots, \omega_n\}$ 对某准则 $C$ 的影响程度,Saaty等<sup>[15]</sup>建议可以采取1-9比例标度对因子进行两两比较,运用比值建立对比矩阵,即每次取两个因子 $\omega_i$ 和 $\omega_j$ , $a_{ij}$ 表示 $\omega_i$ 和 $\omega_j$ 对准则 $C$ 的影响程度之比,全部比较结果用矩阵 $A = (a_{ij})_{n \times n} = (\omega_i / \omega_j)_{n \times n}$ 表示。若对比矩阵 $A$ 的元素满足等式: $a_{ij} a_{jk} = a_{ik}$ , $\forall i, j, k = 1, 2, 3, \dots, n$ ,则矩阵 $A$ 为一致性矩阵,具有完全一致性。

#### 2.3.3 层次单排序及其一致性检验

对应于对比矩阵最大特征值 $\lambda_{\max}$ 的特征向量,经归一化后被记为 $U$ 。 $U$ 的元素为同一层次元素对于上一层某因素的相对重要性的排序权值,这一过程被称为层次单排序。

为了确保层次单排序的有效性,使得对比矩阵具有一致性,即使特征向量 $U$ 真实地反映元素在目标中所占的比重,引入一致性指标 $CI$ :

$$CI = \frac{\lambda_{\max} - n}{n - 1} \quad (2)$$

$CI$ 值的误差会随着对比矩阵维数 $n$ 的增大而增大。因此,引入平均随机一致性指标 $RI$ <sup>[15]</sup>,部分 $RI$ 值如表1所列。

表1 平均随机一致性指标

Table 1 Average random indexes

$n$	3	4	5	6	7	8	9	10
$RI$	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

定义 $CR$ 为一一致性比率,则 $CR = CI/RI$ ,当 $CR < 0.10$ 时,称对比矩阵具有满意的一致性,否则就需要调整对比矩阵,使之具有满意的一致性。

#### 2.3.4 层次总排序及其一致性检验

计算同一层次所有因素对于最高层(总目标)相对重要性

的排序权值,该值被称为层次总排序。若上一层次 A 包含  $k$  个因素  $A_1, A_2, \dots, A_k$ , 其层次总排序的权值分别为  $a_1, a_2, \dots, a_k$ , 下一层次 B 包含  $m$  个因素  $B_1, B_2, \dots, B_m$ , 对于因素  $A_j$ , 层次单排序权值分别为  $b_{1j}, b_{2j}, \dots, b_{mj}$ , 此时 B 层次总排序的权值为  $\sum_{j=1}^k a_j b_{mj}$ 。这一过程是从高层到低层进行的, 如果 B 层次的某些因素对于  $A_j$  单排序的一致性指标为  $CI_j$ , 相应地平均随机一致性指标为  $RI_j$ , 则 B 层次总排序一致性比率为:

$$CR = \frac{\sum_{j=1}^k a_j CI_j}{\sum_{j=1}^k a_j RI_j} \quad (3)$$

类似地,  $CR < 0.10$  时, 称对比矩阵具有满意的一致性, 否则须调整对比矩阵的元素取值, 使之具有满意的一致性。

### 3 基于 AHP 的回归测试用例优先级排序方法

#### 3.1 影响因素

本文将影响测试用例排序的 3 个因素<sup>[10]</sup>(缺陷检测率、缺陷检测比、严重性检测能力)作为准则。

(1) 缺陷检测率(RFT)。Kavitha 等<sup>[9]</sup>将其定义为测试用例单位时间内发现的平均缺陷数。对于测试用例  $T_j$ ,  $RFT_j$  由测试用例  $T_j$  发现的缺陷数量  $N_j$  以及  $T_j$  检测到缺陷所需的时间  $time_j$  计算得到。

$$RFT_j = \frac{N_j}{time_j} \times 10 \quad (4)$$

(2) 缺陷检测比(PFD)。Tyagi 等<sup>[10]</sup>将其定义为测试用例  $T_j$  发现的缺陷数量  $N_j$  占总缺陷数  $N$  的比例。为了方便, 并未乘以 100, 而是乘以 10。

$$PFD_j = \frac{N_j}{N} \times 10 \quad (5)$$

(3) 严重性检测能力(RDA)。Tyagi 等<sup>[10]</sup>将其定义为测试用例单位时间内检测严重缺陷的能力。

根据缺陷对软件的影响将缺陷严重性分为<sup>[9]</sup>: 小缺陷、一般缺陷、较大缺陷、严重缺陷以及致命缺陷。为了衡量缺陷的严重程度, 分别给出各种缺陷严重程度的定量值: 2, 4, 6, 8, 10。

Kavitha 等<sup>[9]</sup>讨论了测试用例  $T_j$  检测到的各个缺陷的严重程度值之和  $S_j$ , 计算公式如下:

$$S_j = \sum_{k=1}^t SV_k \quad (6)$$

其中,  $t$  表示测试用例  $T_j$  检测到的缺陷数,  $SV_k$  表示检测到的单个缺陷的严重程度值。

测试用例  $T_j$  的严重性检测能力 RDA 的计算公式如下<sup>[10]</sup>:

$$RDA_j = \frac{S_j \times N_j}{time_j} \quad (7)$$

其中,  $S_j$  表示测试用例  $T_j$  的严重性值,  $N_j$  表示测试用例  $T_j$  发现的缺陷数量,  $time_j$  表示  $T_j$  检测到缺陷所需的时间。

#### 3.2 测试用例优化层次结构模型

测试用例优化采用层次分析法, 该方法以优化测试用例序列为目标层, 以 3 个影响因素: 缺陷检测率、缺陷检测比、严重性检测能力为准则层, 以待排序的测试用例为方案层, 如图 2 所示。

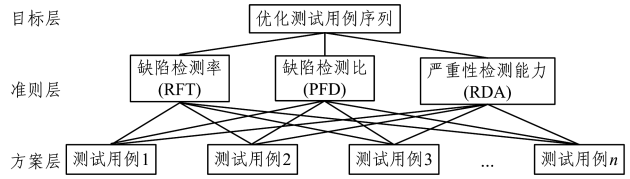


图 2 测试用例优化的层次结构模型

Fig. 2 Hierarchical structure model of test case prioritization

#### 3.3 对比矩阵的构造

构造准则层对于目标层的对比矩阵时, 可以根据实际需求给各个影响因素分配权值, 并且可以扩展影响因素的个数, 根据定义得出相应的对比矩阵。例如, 准则层 3 个影响因素的权值分别为 1, 2, 3, 则对比矩阵为:

$$A = \begin{bmatrix} 1 & 1/2 & 1/3 \\ 2 & 1 & 2/3 \\ 3 & 3/2 & 1 \end{bmatrix}$$

构造方案层对于准则层的对比矩阵时, 本文无需依据 1—9 比例标度凭借经验给各元素赋权值, 而是结合定义根据测试集相关数据来两两比较得到对比矩阵。

本文的对比矩阵严格按照其定义构造, 并非凭借经验所得, 因此单排序、总排序均能通过一致性检验。

#### 3.4 基于 AHP 的回归测试用例优先级排序算法

本节结合层次分析法的步骤与测试用例优化应用给出了一个详细的最优的算法求解排序序列, 算法总体流程如图 3 所示。

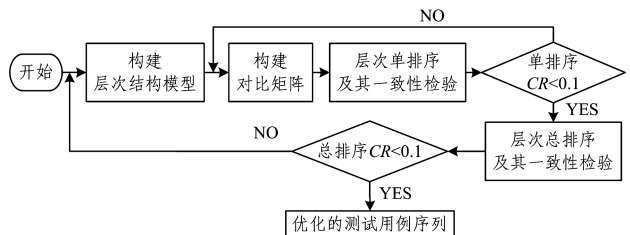


图 3 基于 AHP 的回归测试用例优先级排序方法的流程

Fig. 3 Process of regression test case prioritization method based on AHP

首先, 以优化测试用例序列为目标层要素, 以各测试用例为方案层要素, 以根据实际测试需求确定的影响因素为准则层要素, 构建层次结构模型。然后, 根据影响因素与对比矩阵定义构造准则层对于目标层的对比矩阵, 以及方案层对于准则层的对比矩阵。随之, 计算各对比矩阵的最大特征值及其对应的特征向量, 并将其归一化, 从而进行单排序, 计算一致性比率  $CR$ , 并对一致性比率  $CR$  进行一致性检验。若不通过一致性检验, 则重新构造对比矩阵。最后, 进行总排序, 计算一致性比率  $CR$ , 并对一致性比率  $CR$  进行一致性检验。若不通过一致性检验, 则重新考虑模型; 若通过一致性检验, 则将总排序值递减排列, 得到的对应的测试用例序列即为优化测试用例序列。

#### 算法 1 基于 AHP 的回归测试用例优先级排序方法

输入: 准则层影响因素权值, 测试用例对应的 RFT 值  $r_i$ 、PFD 值  $p_i$ 、RDA 值  $d_i$

输出:优化后测试用例序列  $x$

Step1 构建层次结构模型

确定目标层、准则层、方案层的要素。

Step2 构造对比矩阵

1. 准则层对于目标层的对比矩阵:

$$A = (a_{ij})_{n \times n} = (w_i/w_j)_{n \times n}$$

2. 方案层对于 RFT 的对比矩阵:

$$B_1 = (b_{ij})_{n \times n} = (r_i/r_j)_{n \times n}$$

3. 方案层对于 PFD 的对比矩阵:

$$B_2 = (b_{ij})_{n \times n} = (p_i/p_j)_{n \times n}$$

4. 方案层对于 RDA 的对比矩阵:

$$B_3 = (b_{ij})_{n \times n} = (d_i/d_j)_{n \times n}$$

Step3 层次单排序及其一致性检验

计算对比矩阵最大特征值及其对应特征向量,并将向量归一化。

1. 对比矩阵  $A$  最大特征值对应特征向量归一化为:

$$W = (w_1, w_2, w_3)^T$$

2. 对比矩阵  $B_1$  最大特征值对应特征向量归一化为:

$$E = (e_1, e_2, \dots, e_n)^T$$

3. 对比矩阵  $B_2$  最大特征值对应特征向量归一化为:

$$F = (f_1, f_2, \dots, f_3)^T$$

4. 对比矩阵  $B_3$  最大特征值对应特征向量归一化为:

$$G = (g_1, g_2, \dots, g_3)^T$$

一致性检验:计算各对比矩阵的一致性指标 CT 和一致性比率 CR,若一致性检验不通过,则跳转至 Step2,重新构造对比矩阵。

Step4 层次总排序及其一致性检验

$$C = (E, F, G), Q = C \cdot W = (x_1, x_2, \dots, x_3)^T$$

一致性检验:计算总排序一致性比率 CR,若一致性检验不通过,则跳转至 Step 1,重新考虑模型。

Step5 获得最优测试用例序列

将  $x_i$  值递减排列,得出对应测试用例  $T_i$  的序列  $x$ ,即为优化测试用例序列。

### 3.5 影响因素的比例优化

由于准则层影响因素的比例的设置对排序效果的影响较大,此处采用单一变量法和主成分分析的思想<sup>[16]</sup>分析最优影响因素比例,并运用 Matlab 求解,步骤如下:

(1)设置影响因素的初始比例为  $w_1 : w_2 : w_3 = 1 : 1 : 1$ ;

(2)依次调整其中一个因素的权值,其余因素保持权值不变,分析得出影响测试用例优化评价标准 APFD 值的主要影响因素;

(3)只调整主要影响因素的权值,其余因素权值不变,直至 APFD 值保持不变,取最小整数权值构成最优比例。

## 4 实验分析

为了验证基于 AHP 的测试用例优先级排序方法的合理性和有效性,本文以文献[9]中的测试用例集及其相关属性数据为例,寻求以下问题的解答。

Q1:在多影响因素前提下,基于 AHP 方法以最优影响因素比例优化测试用例序列的缺陷检测能力是否超过单因素排序的缺陷检测能力?多影响因素排序是否更合理?

Q2:与其他方法相比,基于 AHP 方法以最优影响因素比

例优化测试用例序列是否更具有效性?

### 4.1 实验设置

运用 AHP 优化测试用例序列,可根据提高缺陷检测能力的目标调整影响因素比例,灵活性较强。本文采用文献[9]中的测试用例集及其相关属性数据,具体如表 2 所列。

表 2 测试用例的属性

Table 2 Properties of test cases

缺陷	测试用例									
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
F1								*	*	
F2		*	*		*					
F3				*		*				*
F4		*	*							
F5								*		
F6								*	*	
F7				*	*		*			
F8	*					*				
F9			*			*				*
F10	*							*		
缺陷数	2	2	2	3	2	3	1	4	2	2
运行时间/s	9	8	14	9	12	14	11	10	10	13
总严重性	6	6	6	10	8	10	4	20	12	6

实验 1 将测试用例按照各单影响因素进行排序,将其缺陷检测能力与基于 AHP 方法以最优影响因素比例排序的缺陷检测能力作对比,验证基于 AHP 的回归测试用例优先级排序方法的合理性。

实验 2 将基于 AHP 方法以最优影响因素比例排序的缺陷检测能力与文献[9]和文献[10]所提出的方法作对比,验证基于 AHP 的回归测试用例优先级排序方法的有效性。

由式(4)一式(7)和表 2 的数据计算缺陷检测率 RFT、缺陷检测比 PFD、严重性检测能力 RDA,具体的数据如表 3 所列。

表 3 测试用例的 RFT, PFD, RDA

Table 3 RFT, PFD and RDA of test cases

测试用例	缺陷检测率 (RFT)	缺陷检测比 (PFD)	严重性检测能力 (RDA)
T1	2.220	2	1.333
T2	2.500	2	1.500
T3	1.428	2	0.857
T4	3.330	3	3.333
T5	1.660	2	1.333
T6	2.142	3	2.142
T7	0.900	1	0.364
T8	4.000	4	8.000
T9	2.000	2	2.400
T10	1.538	2	0.923

### 4.2 最优影响因素比例

(1)设置初始影响因素比例为  $1 : 1 : 1$ ,即构造准则层对于目标层的对比矩阵如下:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

根据算法设定的步骤求解,层次总排序结果如表 4 所列。将总排序值降序排列,得到优化测试用例序列:T8, T4, T6, T9, T2, T1, T5, T10, T3, T7。

表 4 层次总排序结果

Table 4 Results of overall ranking levels

准则层	RFT	PFD	RDA	总排序值
方案层	0.333	0.333	0.333	
T1	0.102	0.087	0.060	0.083
T2	0.115	0.087	0.068	0.090
T3	0.0657	0.087	0.039	0.064
T4	0.153	0.130	0.150	0.144
T5	0.076	0.087	0.060	0.074
T6	0.099	0.130	0.097	0.109
T7	0.041	0.043	0.016	0.033
T8	0.184	0.172	0.361	0.239
T9	0.092	0.087	0.108	0.096
T10	0.071	0.087	0.042	0.067

(2)运用单一变量原则更改 3 种影响因素的比例(RFT:PFD:RDA),根据算法设定的步骤进行排序,并计算出 APFD 值。

由表 5 可知,缺陷检测率 RFT 对测试用例优化评价标准 APFD 值的影响最大,因此将其作为主要影响因素。

表 5 不同比例下测试用例的优化结果

Table 5 Optimization results of test cases with different ratios

影响因素比例	优化测试用例序列	APFD/%
1:1:1	T8,T4,T6,T9,T2,T1,T5,T10,T3,T7	82
1:1:2	T8,T4,T6,T9,T2,T1,T5,T10,T3,T7	82
1:2:1	T8,T4,T6,T9,T2,T1,T5,T10,T3,T7	82
1:2:2	T8,T4,T6,T9,T2,T1,T5,T10,T3,T7	82
2:1:1	T8,T4,T6,T2,T9,T1,T5,T10,T3,T7	84

(3)调整主要影响因素 RFT 的权值至 APFD 值不变。

由图 4 可知,当 RFT 的权值大于 4 时,APFD 值均为 85%且保持不变,因此最小整数比 5:1:1 为最优影响因素比例。

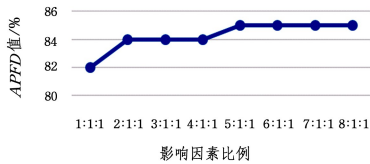


图 4 不同影响因素比例的 APFD 值

Fig. 4 APFD values for different ratios of influencing factors

### 4.3 多影响因素排序的合理性

为了验证多影响因素排序的合理性,以最优影响因素比例排序与 3 个影响因素单独排序结果进行对比。测试用例按表 3 中各影响因素值分别降序排序,从而得到优化序列,如表 6 所列。

表 6 不同准则下测试用例的优化结果

Table 6 Optimization results of test cases with different criteria

影响因素	优化测试用例序列	APFD/%
RFT	T8,T4,T2,T1,T6,T9,T5,T10,T3,T7	85
PFD	T8,T4,T6,T2,T9,T1,T5,T10,T3,T7	84
RDA	T8,T4,T9,T6,T2,T1,T5,T10,T3,T7	81
AHP(5:1:1)	T8,T4,T2,T6,T9,T1,T5,T10,T3,T7	85

### 4.4 层次分析法的有效性

为了验证基于层次分析法的测试用例优先级排序方法的有效性,本实验将该方法与文献[9]和文献[10]所提方法作对

比,结果如表 7 所列。

表 7 不同方法的测试用例优化结果

Table 7 Optimization results of prioritizing test cases with different methods

方法	优化测试用例序列	APFD/%
文献[9]	T8,T4,T9,T6,T5,T2,T1,T10,T3,T7	80
文献[10]	T8,T4,T6,T9,T2,T1,T5,T10,T3,T7	82
层次分析(5:1:1)	T8,T4,T2,T6,T9,T1,T5,T10,T3,T7	85

### 4.5 实验数据分析

根据实验结果,可逐一回答前文提到的 Q1 和 Q2。

Q1:在多影响因素排序前提下,以最优的影响因素比例 5:1:1,运用基于 AHP 的测试用例优先级排序方法来优化序列,APFD 值为 85%。在单影响因素前提下,由图 5 可以看出,按照缺陷检测比、严重性检测能力排序的 APFD 值均低于 85%,按照缺陷检测率排序的 APFD 值与以最优影响因素排序的 APFD 值一样,但按缺陷检测率排序方法只考虑了缺陷检测率,未考虑缺陷严重性缺乏对测试用例的综合评价。因此,考虑多影响因素可使排序效果更具合理性。

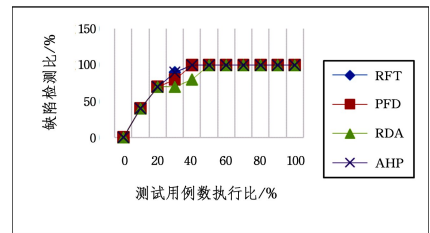


图 5 实验一中各准则的 APFD 值

Fig. 5 APFD values of different criterias in experiment 1

Q2:文献[9]与文献[10]所提方法的 APFD 值均小于 85%。由图 6 可知,文献[9]中影响因素只有两个,因此适当扩充影响因素的数量可使排序效果更具合理性;文献[10]中影响因素虽有 3 个,但比例固定,不一定能满足实际需求或达到最优效果,缺乏灵活性。因此,基于 AHP 的测试用例优先级排序方法更具有效性。

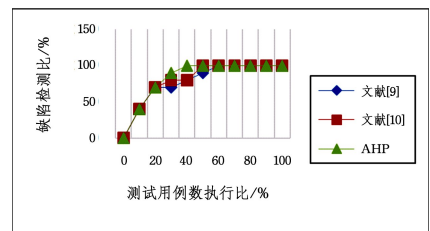


图 6 实验二中各方法的 APFD 值

Fig. 6 APFD values for different methods in experiment 2

**结束语** 本文提出的基于 AHP 的测试用例优先级排序方法可用于提高回归测试的缺陷检测率,具有灵活性。构造层次分析结构模型时可考虑多种影响测试用例的因素,并将其作为准则,优化各个影响因素权值以达到实际测试的需求。然而,影响回归测试用例优先级排序的因素众多,本文只考虑了 3 种,未来进一步的研究方向是寻找实例,扩大测试用例规模,以确定更多重要的影响因素,根据实际测试需求优化影响因素比例,经过层次分析法得到最优测试用例序列。

## 参考文献

- [1] CHEN X, CHEN J H, JU X L, et al. Survey of test case prioritization techniques for regression testing[J]. Journal of Software, 2013, 24(8):1695-1712. (in Chinese)  
陈翔, 陈继红, 鞠小林, 等. 回归测试中的测试用例优先排序技术述评[J]. 软件学报, 2013, 24(8):1695-1712.
- [2] YOO S, HARMAN M. Regression testing minimization, selection and prioritization: A survey[J]. Software Testing, Verification & Reliability, 2012, 22(2):67-120.
- [3] HARROLD M, ORSO A. Retesting software during development and maintenance[C] // Proceedings of the Frontiers of Software Maintenance. Beijing: IEEE Press, 2008: 99-108.
- [4] WONG W E, HORGAN J R, LONDON S, et al. A Study of Effective Regression Testing in Practice[C] // Proceedings of IEEE International Symposium on Software Reliability Engineering. New Mexico: IEEE Press, 1997: 264-274.
- [5] ROTHERMEL G, UNTCH R H, CHU C, et al. Test case prioritization: An empirical study[C] // Proceedings of IEEE International Conference on Software Maintenance. Oxford: IEEE Press, 1999: 179-188.
- [6] ELBAUM S, MALISHEVSKY A G, ROTHERMEL G. Prioritizing test cases for regression testing[C] // Proceedings of the 2000 ACM SIGSOFT international symposium on Software Testing and Analysis. Oregon: IEEE Press, 2000: 102-112.
- [7] MAHESWARI R U, JEYAMALA D. A novel approach for test case prioritization[C] // Proceedings of IEEE International Conference on Computational Intelligence and Computing Research. Enathi: IEEE Press, 2013: 1-5.
- [8] WANG Y, ZHAO X, DING X. An effective test case prioritization method based on fault severity[C] // Proceedings of IEEE International Conference on Software Engineering and Service Science. Beijing: IEEE Press, 2015: 737-741.
- [9] KAVITHA R, SURESHKUMAR N. Test Case Prioritization for Regression Testing based on Severity of Fault[J]. International Journal on Computer Science and Engineering, 2010, 2(5):1462-1466.
- [10] TYAGI M, MALHOTRA S. An approach for test case prioritization based on three factors[J]. International Journal of Information Technology and Computer Science (IJITCS), 2015, 7(4):79-86.
- [11] MAHMOOD H, HOSAIN S. Improving test case prioritization based on practical priority factors [C] // Proceedings of IEEE International Conference on Software Engineering and Service Science. Beijing: IEEE Press, 2017: 899-902.
- [12] CHANG L H, MIAO H Q, XIAO L. Self-adaptive Test Case Prioritization Based On History Information [J]. Computer Science, 2015, 42(9):154-158. (in Chinese)  
常龙辉, 缪淮扣, 肖蕾. 基于历史信息的自适应测试用例优先级技术[J]. 计算机科学, 2015, 42(9):154-158.
- [13] ROTHERMEL G, UNTCH R H, CHU C, et al. Prioritizing testcases for regression testing[J]. IEEE Transactions on software engineering, 2001, 27(10):929-948.
- [14] XU J, LIU N. The basic idea and practical application of analytic hierarchy process [J]. Information Research, 2008(12): 113-115. (in Chinese)  
徐俊, 刘娜. 层次分析法的基本思想与实际应用[J]. 情报探索, 2008(12):113-115.
- [15] 姜启源, 谢金星, 叶俊. 数学模型(第四版)[M]. 北京: 高等教育出版社, 2011: 249-269.
- [16] 何晓群. 多元统计分析(第四版)[M]. 北京: 中国人民大学出版社, 2015: 113-141.