

実験計画法を用いたソフトウェアのテスト項目設定法

下川 浩樹 (富士通 銜)

1. はじめに

ソフトウェアのテストは、プログラムの内部構造に着目し、プログラムの中に存在する構造上の異常や誤りを検出する「ホワイト・ボックス・テスト」(構造テスト)とプログラムの内部構造はいっさい無視し、外部仕様書に基づき外部から見える機能だけを検査する「ブラック・ボックス・テスト」(機能テスト)の2つに分けることができる。これらのテストには、テスト実行時に最も重要な要素であるテスト項目の設定という問題がある。「ホワイト・ボックス・テスト」においては、プログラムのパス、エッジ等の網羅率を設定基準として計測性のあるテスト項目の設定法が数多く提案され、適用されている。一方、「ブラック・ボックス・テスト」においては、原因-結果グラフを用いた手法等が提案されているが、これらの手法を用いるには、種々の問題がある。例えば、「原因」と「結果」の因果関係を表す為に、対象機能の内部構造にまで熟知していなければならず、第三者の立場(検査部門)の人が外部仕様書を基に実施するテストには適さない。また、限られた工数の中で機能毎に原因-結果グラフを書く事は、膨大な工数を必要とし効率的なテスト手法とはいえない等々。

本論文では、「ブラック・ボックス・テスト」におけるテスト項目設定時に実験計画法で用いられている直交配列表を応用し、テスト項目を自動設定する手法について考えた。次章以下で、従来のテスト項目設定法とその問題点、実験計画法での直交配列表の適用法、適用プログラムによる検証及びその効果について述べる。

2. 従来のテスト項目設定法と問題点

テスト項目の設定は、検査対象製品の各機能毎に要因分析表(因子とその状態(取り得る値)で二次元的に表したもの)を作成し、その因子毎の状態の組合せを選択する事により行っている。図-1の手順例に従って以下に説明を行う。

- ① 検査対象製品の仕様把握及び要因洗い出し
  - ・外部仕様に基づきプログラム仕様及び、要因(動作環境、条件等)の洗い出しを行う。
- ② 要因分析表の作成
  - ・オペランドや外部要因を因子、因子の取り得る値を状態とし、2次元的に表す要因分析表を作成する。

- ③ テスト項目の設定
  - ・テスト担当者による組合せの選択(因子毎の状態の組合せの選択)及び設定テスト項目に対するグループ内レビュー実施。
- ④ テスト項目設定表の作成
  - ・組合せの選択によって得られるテスト項目をテスト項目設定表として作成する。

① (外部仕様書)

名札	命令	オペランド
	FID	TYPE = { OUT } PGM = { DSP, AIM, ACS } COPYLIB = { BATCH, YES, NO }

↓

② (要因分析表) 因 子

	A TYPE	B PGM	C COPYLIB
状態 1	OUT	AIM	YES
2	DSP	ACS	NO
3		BATCH	

↓

③ (テスト項目の設定)

[組合せの選択(思考)]  
[グループ内レビュー]

↓

④ (テスト項目設定表)

	A TYPE	B PGM	C COPYLIB
テスト項目 1	OUT	AIM	YES
2	OUT	ACS	NO
3	OUT	BATCH	YES
4	DSP	AIM	NO

図-1

③のテスト項目の設定時において、各因子毎の状態をすべて組合せて設定するのは、膨大な数となり不可能である。そこで、テスト担当者が組合せの選択を行い、テスト項目の設定を行っている。しかし、このようなテスト項目の設定には、以下の様な問題がある。

- ・設定基準がなく、テスト項目の設定が統一化できない
- ・設定されたテスト項目の網羅性に対する確認が困難
- ・多くの工数を要する

そこで、テスト項目の設定において、実験計画法を用いて組合せを行い、さらにそれを自動化する事により問題解決を計る事を考えた。

### 3. テスト項目設定のための手法

テスト項目設定において、因子間の組合せを考慮し、実現可能な数のテスト項目の設定が行える基準を、実験計画法における配置の構成問題を用いて設定することにした。

#### 3.1 実験計画法の利用

実験計画での直交配列の条件を定義し、それを有限体上で組合せ論的に考えてみる。

まず、テスト項目の設定基準を、次の様に設定する。

任意の2つの因子間において、すべての状態の組合せが必ず同一回数存在する。

これは、先の例(図-1)で言うと、因子AとBで(A1, B1), (A1, B2), (A1, B3), (A2, B1), (A2, B2), (A2, B3) が必ずどれかのテスト項目に存在し、また因子BとC, 因子AとCでも同様に各状態の組合せが存在するという事である。

ここで、大きさpのガロア体GF(p)上のn変数の一次独立な同次一次式をm個考える。(ガロア体である為にはpは素数の冪乗に限る)

$$\nu_i = \alpha_{i1} \theta_1 + \dots + \alpha_{in} \theta_n = \sum_{j=1}^n \alpha_{ij} \theta_j \quad (a)$$

(i = 1, \dots, m)

$$\nu_i, \alpha_{ij}, \theta_j \in GF(p)$$

ここで同次一次式が、どの一次式を取ってもその対が必ず存在する為には、その中のn変数 ( $\theta_1, \dots, \theta_n$ ) をガロア体GF(p)の各一点 (0, \dots, p-1) を取らせることにより得られる ( $\nu_1, \dots, \nu_m$ ) の値を求めればよい。

一般に、n変数の場合に存在する一次独立な同次一次式

は、 $\frac{p^n - 1}{p - 1}$  で与えられるから、m個の一次独立な同

次一次式を得るために必要な最小の変数nは、次の式で与えられる。

$$\frac{p^{n-1} - 1}{p - 1} < m \leq \frac{p^n - 1}{p - 1} \quad (b)$$

例として、p = 2, m = 3の場合を考える。

式(b)より必要な変数nは2であり、互いに一次独立な同次一次式は、最大3つ得られ、以下の式が考えられる。(これ以外の同次一次式は、この3つのどれかに従属する)

$$\begin{aligned} \nu_1 &= \theta_1 \\ \nu_2 &= \theta_2 \\ \nu_3 &= \theta_1 + \theta_2 \end{aligned}$$

( $\theta_1, \theta_2$ ) をガロア体GF(2)の各点を取らせる事により、次の組合せが得られる。

組合せNo.	$\nu_1$	$\nu_2$	$\nu_3$
No. 1	0	0	0
No. 2	1	0	1
No. 3	0	1	1
No. 4	1	1	0

これは、 $\nu$  の数 (mの値) を因子の数、pを因子の状態数とした場合、テスト項目設定基準を満足する。

これを一般化したものが、表-1である。

因子項目	1	2	3	4	5	6	7	8	9	10	11	12	13	14
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	1	0	1	0	0	1	1	0	1	0	1	0	1	0
T3	0	1	1	0	1	0	1	0	0	1	1	0	0	1
T4	1	1	0	0	1	1	0	0	1	1	0	0	1	1
T5	0	0	0	1	1	1	1	0	0	0	0	1	1	1
T6	1	0	1	1	1	0	0	0	1	0	1	1	0	1
T7	0	1	1	1	0	1	0	0	0	1	1	1	1	0
T8	1	1	0	1	0	0	1	0	1	1	0	1	0	0
T9	0	0	0	0	0	0	0	1	1	1	0	1	1	1
T10	1	0	1	0	0	1	1	1	0	1	1	1	0	1
T11	0	1	1	0	1	0	1	1	1	0	0	1	1	0
T12	1	1	0	0	1	1	0	1	0	0	1	1	0	0

表-1

### 3.2 要因分析表への組合せ表の適用

因子毎の状態数が素数の冪乗値の時の組合せ表をそれぞれ作成し、要因分析表をその組合せ表に適用させる事によって条件を満足したテスト項目の設定が可能となる。しかし、これには因子毎の状態数を素数の冪乗の値で均一にしなければならず、要因分析表の改造が必要となる（一般に、因子毎の状態数はバラバラであり、均一である事はほとんど無い）

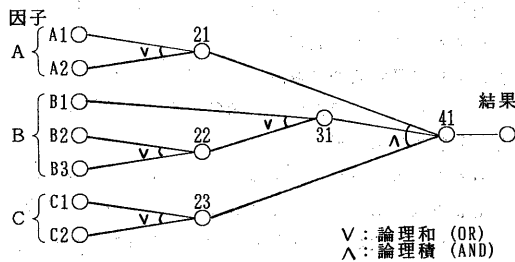
そこで、因子毎の状態を論理命令で組合せたブール・グラフを作成し、その各節点での適用を考えてみた。

ブール・グラフにおいて、最上位節点への同時入力となる節点数を因子数、各々の節点が入力状態数（状態数2固定）を因子の状態数と想定することにより、状態が2の組合せ表を適用する。

図-1の例を用いて項目設定までを説明する。

#### (1) ブール・グラフの作成

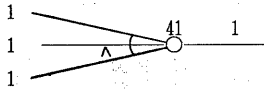
因子毎の状態を入力、結果を出力とし、その入力と出力を論理命令を用いて結合したブール・グラフを作成する。（各節点の入力は、2入力とする。）



#### (2) 各節点での組合せ表(表-1)の適用

##### ① 節点41(最上位節点)の入力

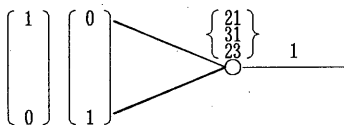
結果が1(存在する)となる時、節点41の出力は1であるから、入力はすべて1となる。



##### ② 節点21, 31, 23の入力

節点21, 31, 23の出力は、1であるから、入力

は  $\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$  となる。



注) 1つのテスト項目の中に同一因子の状態が、複数存在することはないので入力値  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$  は除く。

#### ③ 節点21, 31, 23での組合せ表適用

ここで、節点を因子(因子数=入力=3)、節点の入力となる値  $\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$  を状態(状態数

=入力数=2)と想定し、そこに組合せ表(図-2)を適用する。

まず、節点が3つあるので、因子数3の時の組合せを組合せ表から選んで適用させる。

組合せ数	節点21	節点31	節点23
T 1	0	0	0
T 2	1	0	1
T 3	0	1	1
T 4	1	1	0

これに、入力状態No.0に節点の入力となる値  $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$  入力状態No.1に  $\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$  を割当て、各々の節点番号を入力に分割する。

組合せ数	21	A1	A2	31	B1	B2	23	C1	C2
T 1	1	0	1	0	1	0	1	0	
T 2	0	1	1	0	0	0	1	1	
T 3	1	0	0	1	0	0	1	1	
T 4	0	1	0	1	1	1	0	0	

節点22に対しても同様に適用し、最終的に次の6項目のテスト項目が設定される。

テスト項目No.	A	B	C
1	A1	B1	C1
2	A2	B1	C2
3	A1	B2	C2
4	A1	B3	C2
5	A2	B2	C1
6	A2	B3	C1

これにより、要因分析表の改造なしで組合せ表を適用することが可能となった。

### 3.3 組合せ表の改造

表-1の組合せ表を適用した場合、因子数の増加による項目数の急増という問題がでてくる。

例えば、因子数が3から4になると、組合せ数が4項目から倍の8項目となる。同様に因子数7から8で、組合せ数が倍の16項目となる。

そこで、テスト項目の設定基準を次の様に変更した。

任意の2つの因子間において、すべての状態の組合せが必ず1つ以上存在する。

この基準に基づき、任意の2つの因子間のすべての状態の組合せが、均等（同一回数）に存在する組合せ（表-1）から、1つ以上存在する組合せに改造する事により、因子数増加による項目数増加の緩和を計る事が可能となる（項目数の増加を、従来の倍増（×2）から半減する事ができる）

改造した組合せ表を、表-2に示す。

因子節点	1	2	3	4	5	6	7	8	9	10	11	12	13	14
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	1	0	1	1	0	1	1	0	1	1	0	1	1	0
T3	0	1	1	0	1	1	0	1	1	0	1	1	0	1
T4	1	1	0	1	1	0	1	1	0	1	1	0	1	1
T5	0	0	1	1	1	0	0	0	1	1	1	0	0	0
T6	1	1	0	0	0	1	1	1	0	0	0	1	1	1
T7	0	1	1	1	0	1	1	0	0	0	1	0	0	1
T8	1	0	0	0	1	0	0	1	1	1	0	1	1	0
T9	0	1	0	0	1	0	1	1	0	0	1	0	1	0
T10	1	0	1	1	0	1	0	0	1	1	0	1	0	1

表-2

この様に、実験計画法の適用により作成した組合せ表を用いる事で、テスト項目の設定基準を満足した形で実作業にそのまま適用する事が可能となった。

#### 4. 適用プログラムによる検証

検査対象製品プログラムに対して、本技法（実験計画法を用いたテスト項目設定法）を適用し、その検証を行った。次に、検証項目毎の結果を示す。

##### 4.1 テスト項目数及び検出障害数

従来のテスト項目設定法と本技法とで設定されたテスト項目数、及び検出障害数との比較を行った。プログラム毎の比較図を図-2に示す。

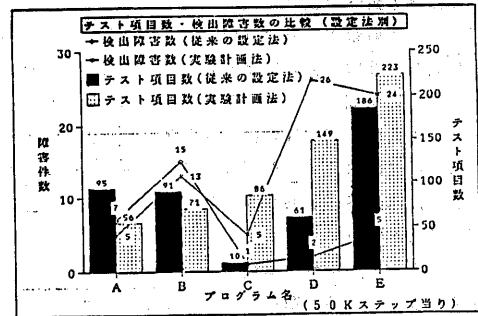


図-2

一般的に本技法を用いた方が、従来の設定法よりもテスト項目は多く設定される。また、内容面からのテスト項目設定範囲においても、全範囲にわたり均一な設定がなされる。

従来の設定法においては、テスト項目数、設定範囲が各プログラム毎にバラバラで不均一な設定となっている。従来の設定法を用いたテスト項目の分析結果を次に示す。

##### ・テスト項目設定不足 ~ プログラムC

（機能全般からテスト項目は設定されているが、項目数不足の為、障害検出数が少ない。）

##### ・テスト項目設定範囲の偏り ~ プログラムD, E

（テスト項目の設定範囲が、機能のある部分に偏っており、障害検出力が低い。）

プログラムA, Bについては、テスト項目数、設定範囲ともに全般にわたって設定されており、本技法での設定項目と同様に良好なテスト項目設定がなされている。

#### 4.2 検出バグ分析

プログラムの検査及び利用時に検出された障害に対して、原因分析（単一原因（一因子の状態によるもの）、複合原因（因子間の状態の相互作用によるもの））を行うことにより、組合せによって設定されたテスト項目の有効性を計ってみる。原因分析の結果を表-3に示す。

プログラム名	単一原因 検出障害	複合原因 検出障害 (2因子間)	複合原因 検出障害 (3因子間以上)
a	13	18	37
b	0	1	2
c	1	3	5
d	8	12	21
e	3	9	11
f	5	10	18
合計	30	53	94

表-3

検出障害原因は、ほとんどが（全体の83%）が複合原因によるものであり、要因分析表での因子間の状態の組合せ選択が重要であることがわかる。

ここで表-3の複合原因（3因子間以上）による検出障害が本技法を用いたテスト項目で、どの程度検出できるかを調べてみた。その結果を表-4に示す。

プログラム名	複合原因 3因子間以上	検出可能	検出不可
a	37	3	34
b	2	2	0
c	5	1	4
d	21	1	20
e	11	3	8
f	18	2	16
合計	94	12	82

表-4

3因子間以上の複合原因による障害の検出率は、13%あり、本技法のテスト項目設定基準でもかなりの障害が検出できる。また、全障害177件中、95件の障害が検出でき、障害検出率54%とかなり高い値を示している。これは、テスト項目を全組合せ設定した場合と本技法を用いて設定した場合の項目数の比較を行うことにより、効率の良いテスト項目が設定されている事がわかる。

#### 5. 運用法及び適用状況

この手法を用いたテスト項目の設定は、端末から入力情報として要因分析表の各因子毎の状態数、組合せの禁止事項である制約条件を与える事により、組合せによって設定されたテスト項目を、テスト項目設定表の帳票として得ることができる。運用法を図-3に示す。

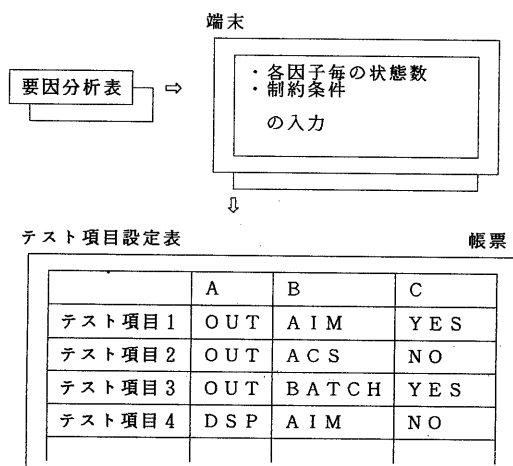


図-3

また、このテスト項目設定法の適用状況を、従来の方法（手作業）によるテスト項目設定との工数比較と合せて表-5に示す。

適用機能	設定項目数	本手法を用いた 設定工数	従来の方法を用いた 設定工数*
プログラムA	3877	5.5 H	176 H
プログラムB	1729	3.5 H	80 H
合計	5606	9.0 H	256 H

表-5

\*機能の量（プログラムの複雑度）、プログラムの規模等によって設定工数は異なるが、経験的に推定した。

#### 6. 効果

ソフトウェアのテスト作業におけるテスト項目の設定には、豊富な経験と創造力が必要であると言われている。本技法においては、上記の有無によるテスト項目の質・量の格差をなくし、誰にでも一定の質・量のテスト項目が設定できる事を目差した。

数種のプログラムに対して本技法を適用した結果、それらから期待した結果を得ることができた。以下に、そのいくつかの効果について示す。

##### ① 網羅性のあるテスト項目の設定

網羅度というのは、基本的には機能テストの様なブラックボックス指向のテストにおいて云々できるものではない。しかし、少なくとも2因子間のすべての状態の組合せは必ず存在するというテスト項目が設定された事で、ホワイトボックス指向のテストと言う条件網羅の度合の向上が見られたと言える。

##### ② テスト項目量のめやすの設定

各因子毎の状態の組合せすべてをテスト項目として設定したとしても、テスト工程・工数上から現実には実施不可能である。

そこで、テスト工程・工数を考慮し、実現可能で、かつ、①で述べた意味での網羅性のあるテスト項目の設定によって、十分性に対する目安の設定が可能となった。

##### ③ テスト項目の均一化

従来の手作業によるテスト項目設定に比べ、担当者 の恣意を排除した均一なテスト項目の設定が可能となる。

#### ④ 作業工数削減

工数が大幅削減となることは、表-5に示したとおりである。また、従来の手作業によるテスト項目設定よりは単純で、設定者の限定もなくなるので作業者の負荷の分散を計ることも可能となる。

### 7. 今後の課題

#### (1) 因子・状態の重要度を考慮した項目設定

テスト項目として設定する場合の因子毎の状態における重要度（要因分析時に、PMBCA法、BMBCA法の利用）、因子間の寄与率を考慮した組合せの設定を行い、テスト項目の設定法をより充実させていく必要がある。

#### (2) 要因分析範囲の絞り込み

テスト項目の質的向上には、要因分析における因子及び状態の追加が考えられる。しかし、現状での要因分析における因子の抽出には限界があり、対象機能の規模により大きく左右される。そこで、要因分析の対象範囲の絞り込み（FTAの活用）を行う事により、より有効な要因分析が可能となる。

### 8. おわりに

「ブラック・ボックス・テスト」におけるテスト項目設定法について述べた。実験計画法の直交配列表を応用し、要因分析表にそのまま適用できる様にする事で、効率的なテスト項目が簡単に設定できる様になった。今後は、要因分析の手法をさらに改良していく事により、テスト項目の質的向上、障害検出力の増加を計ることが可能であると考えられる。

#### [参考文献]

- (1) J. Myers (長尾, 松尾訳) ソフトウェアテストの技法  
近代科学社 (1980)
- (2) 田口玄一: 実験計画法  
丸善 (1976)
- (3) 高橋碧郎: 「ガロア体の離散データへの応用」  
数理科学 (1979.11)
- (4) 高橋碧郎: 「シュミレーションの直交実験による  
効率化」  
数理科学 (1972.12)