

第5分科会（第2グループ）

組み合わせテストを用いたソフトウェアテストとその限界

Software test that uses combinatorial test and its limit

主査	高橋 寿一	（ソニー株式会社）
副主査	増田 聡	（日本 IBM 株式会社）
リーダー	能瀬 正博	（株式会社富士通北陸システムズ）
サブリーダー	中子 勝晴	（株式会社日立システムアンドサービス）
研究員	金山 昇平	（株式会社ナナオ）
	那須 俊博	（株式会社セゾン情報システムズ）
	山岸 重幸	（ジャパンシステム株式会社）
	加藤 恒	（ヤマハ株式会社）

（敬称略 順不同）

概要

ソフトウェアの多機能化、肥大化に伴いテストすべき対象が膨大化する一方、テスト期間の短期化が求められている。組み合わせテストを効率的におこなう手法として直交表（実験計画法）がある。この直交表の有効性及びその限界について研究をおこなった。

Abstract

Software having becoming huge and rich in functions, the time we can spend on testing has been required to be shorter and shorter. And in such situations, orthogonal array (Design of Experiments) enables us to execute the combinatorial test effectively. We have researched two side of orthogonal array: its effectiveness, and the limit.

1. 背景

近年、ソフトウェアの多機能化、肥大化に伴いテストすべき項目は膨大化する一方となっているが、テストに投入できる工数及び時間は限られている。また、製品サイクルの短期化が進んでいるにも関わらず、品質に対する要求は以前にも増して高まっている状況である。

研究会メンバーは各社でソフトウェアのQA部門(品質保証部門)に携わり、テスト作業の改善への新しい手法を日頃から模索している。

研究テーマの選定に当たって、組み合わせテスト項目の作成や実施について改善したいという意見が多くあがった。具体的には次の点が挙げられた。

- ・ QA部門として、組み合わせテストをどこまでおこなえば十分なのか
- ・ 組み合わせられる全機能テストをおこなうと、テスト項目が膨大に増加する
- ・ 組み合わせテストの網羅率と実施コストの費用対効果が見えにくい

そこで組み合わせテストの一つとして存在する直交表を対象に、どのように利用すれば上記で求められている効果が得られるか研究することとなった。

2. 研究アプローチ

本研究の重点項目は次に示すものである。

直交表の理解

直交表を適用したことがなく詳細な使用方法も不明であるため、文献やシンポジウム等からの調査を図る。

直交表を用いた組み合わせテストの評価

実際に直交表を用いた組み合わせテスト項目の作成をおこない、その有効性と限界について以下の通り研究をおこなう。

- ・ テストに見合った最適な直交表の因子・水準の見積もり方法
- ・ 直交表を用いた場合のテスト網羅率及び、テスト効率
- ・ 直交表を用いた場合に不足するテスト内容及び、その補完方法

3. 直交表とは

直交表とは、実験計画法として代表的な直交表実験計画法で用いられる表のことで、コスト・スケジュールの制約によって、全てのケースで実験できないときなどに用いられる。因子とその水準が均等に現れる実験条件を決めるために、一つの列の各水準の中に、他の列の各水準がすべて同回数ずつ現れるように作られている。

ソフトウェアで組み合わせテストを実施する際には全ての機能の組み合わせをテストすることが理想であるが、限られた時間とコストで、それらのテスト項目を全て実施することは実質不可能である。しかし、直交表を使用することにより、組み合わせテストの効率化を図ることができると品質工学(タグチメソッド)の田口玄一氏は仮説されている。テストの効率化ができる理由は以下の通りである。

1 因子ずつの場合のバグの発見率が p である場合、直交表を利用して 2 因子の組み合わせの評価をすると、それに比較してほぼ p の 2 乗に期待される。具体的にバグの発生する確率を 1000 行あたり 4 件と仮定して計算した場合、機能組み合わせによるバグ発生率は下記となる。

$$\text{単機能バグ発生率} = 4/1000 = 0.004$$

$$2 \text{ 因子間バグ発生率} = (4/1000)^2 = 0.000016$$

$$3 \text{ 因子間バグ発生率} = (4/1000)^3 = 0.000000064$$

200 万行の製品とすると、

$$\text{単機能バグ数} = 4/1000 \times 2,000,000 = 8,000 \text{ 件}$$

$$2 \text{ 因子間バグ数} = (4/1000)^2 \times 2,000,000 = 32 \text{ 件}$$

$$3 \text{ 因子間バグ数} = (4/1000)^3 \times 2,000,000 = 0.128 \text{ 件}$$

これらのデータから以下のことが言える。

(a) 2 因子間のバグは漏れなく検出しなければならない。

(b) 多因子の組み合わせ (3 因子以上) バグは非常に件数が少ない。

また、3 因子間のバグを確実にテストするためには直交表 \times 最大因子の水準数のテスト項目数が必要であり、通常数 10 倍のテスト規模となる。これらを考慮すると、多因子間の組み合わせを全てテストすることは、それが 3 因子間であったとしても、テスト項目数が多い割に検出されるバグ数は概ね 2 因子間バグ数と変わらず、効率が悪いということになる [1]。

表 3 . 1 L9 直交表

	P1	P2	P3	P4
Test Case 1	0	0	0	0
Test Case 2	1	1	1	0
Test Case 3	2	2	2	0
Test Case 4	0	1	2	1
Test Case 5	1	2	0	1
Test Case 6	2	0	1	1
Test Case 7	0	2	1	2
Test Case 8	1	0	2	2
Test Case 9	2	1	0	2

実際のテスト項目に適用すると

因子 4 個、水準 3 通りのとき、

【全ての組み合わせをテストする場合】

$3^4 = 81$ 通りのテスト項目が発生する。

【直交表を用いた場合】

9 通りのテスト項目となる。

上記の場合、9 分の 1 のテスト項目で 2 因子間網羅率 100% が確保でき、全ての組み合わせテストとほぼ同等のテスト結果が得られる。

なお、この田口玄一氏の仮説については、ソフトウェアテストの分野において、2 因子組み合わせの網羅テストにより効果のあるテストがおこなっていると実際に報告がなされている [2]。

これまでに述べた通り、直交表を利用すると効率的なテストがおこなえるということが分かるが、実際に直交表を使用して組み合わせテスト項目を作成しようとするとき以下のような問題があり難しいとされる。

- ・ 禁則処理、異常処理の因子・水準が組み込めない
- ・ 因子、水準及び、直交表の選択方法が分かりにくい

以上の条件を踏まえた場合の直交表の作成を次章に記載する。

4. 有効な直交表の作成方法

一般的な「路線情報検索プログラム」で組み合わせテストをおこなうことを想定し、直交表の作成方法を具体的に記述する。また、プログラムの仕様については「付録1 一般的な『路線情報検索プログラム』の仕様説明」に示す。

作成にあたっては、専用ツールがなくても作成できるよう Microsoft(R)Excel を使用した。

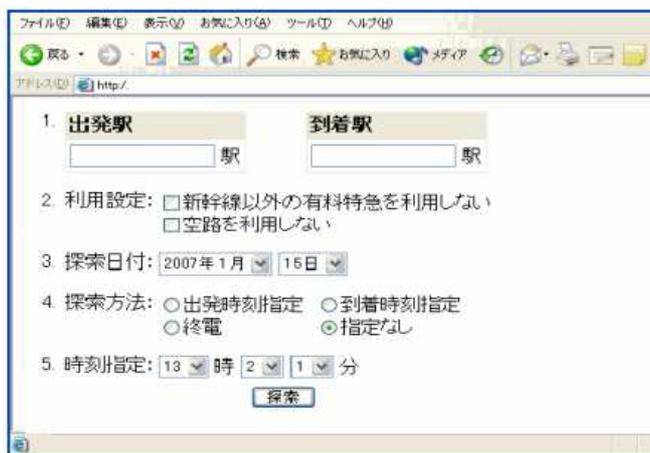


図 4 . 1 路線情報検索プログラム

4.1 因子と水準の決定

4.1.1 因子の洗い出し

図 4 . 1 の画面より、因子として妥当と考えられるものを洗い出す。考えられる因子と、それぞれの因子が保有する全水準数を表 4 . 1 . 1 . 1 に示す。

表 4 . 1 . 1 . 1 路線情報の因子と全組み合わせ数

#	因子	全水準数	水準数算出の根拠
1	出発駅		駅名として登録されているもの全て
2	到着駅		駅名として登録されているもの全て
3	利用設定 (新幹線利用)	2	ON あるいは OFF の 2 通り
4	利用設定 (空路利用)	2	ON あるいは OFF の 2 通り
5	探索日付	124	4 (月) × 31 (日) = 124 通り
6	探索方法	4	出発時刻指定、到着時刻指定、終電、指定なしの 4 通り
7	時刻指定 (時間)	24	0 ~ 23 の 24 通り
8	時刻指定 (分の 10 の位)	6	0 ~ 5 の 6 通り
9	時刻指定 (分の 1 の位)	10	0 ~ 9 の 10 通り
10	OS	8	一般的と思われるものを挙げると、Win95、Win98、WinNT、WinMe、Win2000、WinXP、MacOS9、MacOS の 8 種類
11	ブラウザ	4	一般的と思われるものを挙げると、Internet Explorer、Netscape Navigator、Safari、Firefox の 4 種類

4.1.2 水準の絞込み

表 4 . 1 . 1 . 1 より、探索日付等の水準数が多い項目では、全水準に対してテストを実施した場合に膨大な作業量が発生する。そこで同値分割や限界値 / 境界値分析の手法を用いて水準数を絞る。例として時刻指定について考える。

時刻指定のうち、時間は 0 時から 23 時の 24 通り、分の 10 の位は 0 から 5 の 6 通り、分の 1 の位は 0 から 9 の 10 通りある (分については、画面上 10 の位と

1の位が分かれているため、別々に考えることとする)。それぞれについて同値分割および限界値/境界値分析をおこなった様子を表4.1.2.1に示す。

また、水準を選択する場合、異常(無効)値を水準に入れるとそのテスト項目の実行が失敗してしまい、他の水準間の組み合わせテスト結果を得ることができなくなる。そのため異常値は別途異常系テストを実施することとする。

表4.1.2.1 同値分割と限界値/境界値分析

#	項目	考えられる値	有効/無効	境界値	選択した値
1	時間	- ~ - 1	無効同値クラス	- 1	0、9、10、23
		0 ~ 23	有効同値クラス	0、23	
		24 ~	無効同値クラス	24	
2	分の10の位	- ~ - 1	無効同値クラス	- 1	0、5
		0 ~ 5	有効同値クラス	0、5	
		6 ~	無効同値クラス	0	
3	分の1の位	- ~ - 1	無効同値クラス	- 1	0、9
		0 ~ 9	有効同値クラス	0、9	
		10 ~	無効同値クラス	10	

以上のような分析を他の因子に対してもおこなう。その際、使用する直交表が2水準系直交表であることから、1つの因子に含まれる水準数を2の累乗(2、4、8・・・)としておくことが望ましい。

絞込みをおこない、最終的に決定した因子と水準を表4.1.2.2に示す。

表4.1.2.2 絞込んだ後の因子と水準

#	因子	水準数	水準
1	出発駅	3	津(最も短い駅名)
			東京
			南阿蘇水の生まれる里白水高原(ひらがなで最も長い駅名)
2	到着駅	3	津
			東京
			南阿蘇水の生まれる里白水高原
3	利用設定:新幹線利用	2	ON
			OFF
4	利用設定:空路利用	2	ON
			OFF
5	探索日付	4	2006年12月1日
			2007年1月1日
			2007年2月28日
			2007年3月31日
6	探索方法	4	出発時刻指定
			到着時刻指定
			終電
			指定なし
7	時刻指定(時)	4	0時
			9時
			10時
			23時
8	時刻指定(分)	4	00分
			09分
			50分
			59分
9	OS	4	WinMe
			Win2000
			WinXP
			MacOS
10	ブラウザ	2	InternetExplorer
			NetscapeNavigator

4.2 直交表の決定と作成

4.2.1 直交表サイズの決定

洗い出した因子と水準を元に、直交表のサイズを決定する。

2 因子間網羅率を 100%にするためには、組み合わせたい因子の中で最も大きな水準数をもつもの同士を掛け合わせた数だけ、テストをしなければならない。

表 4 . 1 . 2 . 2 を見ると、最も大きな水準数及び、次に大きい水準数はともに「4」である。ここから、最低限

$$4 \times 4 = 16 \text{ 項目のテスト}$$

が必要であるとわかる。

しかし、突出した水準数をもつ因子がない場合、どんなに因子が増えても上記計算式のテストすべき項目数が増えることがなく、妥当な直交表のサイズを決定することができない。このような場合は、自由度からの見積もりを実施する。自由度とは平均が与えられた場合に自由に定義できる変数の数をいう。

直交表のサイズを自由度から見積もるには、下記の計算式を利用する。自由度を求める式とともに示す。

$$\begin{aligned} \text{因子 A の自由度} &= \text{因子 A の水準数} - 1 \\ \text{直交表のサイズ} &= \text{自由度の総和} + 1 \\ &= (\text{各因子の水準数} - 1) + 1 \end{aligned}$$

この計算式を用い再度計算すると

$$\text{直交表のサイズ} = (4 - 1) \times 5 + (3 - 1) \times 2 + (2 - 1) \times 3 = 22$$

以上より、2 水準系で最も近い L32 直交表 を選択することとする。

4.2.2 直交表の作成

直交表の作成については「付録 2 直交表の作成方法」に示す。

4.3 禁則の回避

4.3.1 禁則マトリクスの作成

禁則とは、同時に選択できない関係のことである。路線情報検索プログラムでは出発駅と到着駅が該当する。同じ出発駅と到着駅を同時に選択できないからである。詳細は「付録 2 直交表の作成方法」に示す。

4.4 直交表の完成

これまでの流れを踏まえ、L32 直交表へ因子および水準を割り付ける。付録表 3 . 3 に完成した直交表を示す。

これで路線情報検索プログラムの組み合わせテストを実施するための項目が完成したが、2 因子間網羅率が 100%であるかどうか、つまり出現していない組み合わせがないかを確認しておく。確認するには Excel のピボットテーブルを使用する。

完成した直交表に、DATA という列を 1 行追加し、表全体を対象としてピボットテーブルを作成する。作成されたピボットテーブルの行と列に確認したい因子をドロップすると、その因子間の組み合わせが何回テスト項目に現れているかが数字で示される。空欄になっている箇所がテスト項目に現れていない組み合わせである。図 4 . 4 . 1 に、時間指定・時間と探索方法とで確認した例を示す。

合計 : DATA	探索方法				
	時間指定・時間	指定なし	終電	出発時刻指定	到着時刻指定
0時	2	2	2	2	8
10時	2	2	2	2	8
23時	2	2	2	2	8
9時	2	2	2	2	8
総計	8	8	8	8	32

図 4 . 4 . 1 2 因子間網羅率が 100%であるかどうかの確認

図 4 . 4 . 1 を見ると空欄となっている箇所はないため、この 2 因子間網羅率は 100%である [3]。

その結果、路線情報検索プログラムでは絞り込んだ因子、水準を総当りでテスト実施した場合 73,728 件（全水準数を掛け合わせた件数）のテスト項目となっていたが、直交表を使用することにより 32 項目で効果的なテストが実施できることになるといえる。

5. 問題点とその改善案

本章まで直交表の使用方法及び、有効性を述べてきたが、直交表によるテストの十分性については検討をおこなってこなかったとともに、その必要十分性について既知の研究もされていない。そのため直交表における、テストの不足していると思われる点及び、改善案を以下に記載する。

単体テストレベルの品質が事前に確保されている必要がある。

個々の単体機能でバグが出る場合は、本来直交表を用いてテストをおこなう組み合わせテストのバグを検出しきれない。一旦差し戻し、単体テストでバグが存在する因子に着目した全水準の再テストをおこなうべきである。異常系項目が含まれていない。

別途、異常系テストを実施する必要がある。4 章の有効な直交表の例として、ありえない日付の 2 月 31 日がある。

仕様書にあらわれない条件でのバグが漏れることがある。

一般的にバグは偏在するものであり、経験則からテストすべき領域を絞り込むことにより新たなバグを検出できる [4]。過去の経験をもとにした推測テストの併用をすべきである。

機能性以外のテストが不十分である。

ソフトウェアの品質要素には非機能部分も存在する。組み合わせテスト以外にも以下に挙げるような非機能テストを必要に応じて実施すべきである。（セキュリティテスト、性能テスト等）

6. まとめ

QA 部門における直交表の活用について以下のような有効性があると考えられる。

- ・組み合わせの網羅率を定量的に評価できる。

テストの組み合わせは 2 因子間網羅率 100%でほぼ問題ない。また直交表では因子・水準が平均的に現れるため、3 因子間網羅率も高まる。

- ・テスト項目の効率化ができる。

テスト内容をほぼ維持しながらテスト項目を減らすことができる。具体的

に 4 章では 73,728 項目が 32 項目に効率化できた。

- ・ 経験のない人でも正確な組み合わせテストがおこなえる。

組み合わせ手順は確立しており、誰でも 2 因子間網羅率 100% のテスト項目を作成できる。但し、因子と水準の選択を誤るとテスト項目の品質が悪くなるので、因子と水準を決めるときには十分な検討を加えることが必要である。

5 章で示した問題点と改善案を踏まえて直交表を活用することで、効果的な組み合わせテストを実施することができる。

7. 今後の課題

今回の研究では文献からの直交表の理解、作成方法の調査からスタートしたため、既存の情報、メンバーのテスト経験からの討論が主であった。

直交表の有効性、テスト項目作成方法について今回研究をおこなったが、直交表によるテストが浸透していない理由は直交表への理解、作成の難しさだけなのか、他にも課題が存在するのか、今後実際に現場で直交表を使用して調査をしていく必要がある。

参考文献

- [1] 秋山浩一, “直交表を活用したソフトウェアテストの効率化 ~ HAYST 法の活用 ~”, ソフトウェアテストシンポジウム大阪 (JaSST' 05 大阪) 講演資料, URL: www.jasst.jp/jasst05w/pdf/S4-1.pdf
- [2] 多田敏彦, “組み合わせテスト生成・網羅率評価ツールの開発”, 第 25 回ソフトウェア品質シンポジウム発表報文集 (日科技連), pp.305-310
- [3] 秋山浩一, “ソフトウェア・テスト PRESS,” 直交表による組み合わせ入門, vo2, pp.89-107, Jan.2006.
- [4] 西康晴, “テストの改善、テストによる改善”, 2006 年度ソフトウェア品質管理研究会 第 6 回例会 講演資料