

ブロック結線モデルを用いた ペアワイズテストの原理に関する考察

古賀 明彦 加藤 正恭 小川 秀人

(株)日立製作所 横浜研究所

1. はじめに

近年、組込みシステムを始めソフトウェアで実現されたシステムが社会で多量に使われ、ソフトウェアの品質確保がますます重要になっている。ソフトウェアの品質確保のためには出荷前に十分なテストを行い、不具合が残らないようにする必要があるが、理論的に可能な入力の実組合せは莫大な数になり、すべてをテストすることは不可能である。この状況を解決するために、近年、オールペア法、あるいは、 n -way ペアワイズ法と呼ばれる方法がよく使われており [1, 2, 3]、我々もこの手法を用いて製品の品質確保に取り組んでいる [1]。この手法は因子（入力パラメータ等）の全組合せをテストするのではなく、任意の n 個 ($n=2, 3, \dots$) の因子間の組合せを網羅するようにテスト項目を生成する方法である。これは一般に不具合は少ない因子の組合せで発生するという経験則によるもので、例えば、それぞれ二つの値をとる 10 個の因子のシステムをすべてテストするためには $2^{10}=1024$ 通りのテストが必要であるが、 $n=2$ とした 2-way ペアワイズ法では 6 個のテストで済む。

このような特性から近年 n -way ペアワイズ法がよく使われているが、不具合を十分に取るためには n をいくつにすればよいかはあまり分かっていない。D. R. Kuhn ら [2] はいくつかの分野のプログラムで因子の組合せ数と発見される不具合の比率を比較しているが分野ごとに比率が異なっている。C. Song ら [3] は彼らの対象システムで対象ごとに $n=2\sim 7$ と大きく異なるため、 n を一律にするのではなく、大きな n で求めたテストケース集合から、ソースコードの網羅率を最大にする部分集合を求める方法を提案している。これらの状況から n -way ペアワイズ法で効率よく、要求された品質のテストを行うためには、対象システムの特性に合わせて適切な n を決定する手法の確立が必要となる。

これに対し、我々は、テスト対象システムのプログラムをランダムに生成して、組合せる因子数 n に対する不具合の発見率をシミュレートすることで対象と必要な因子数 n の関係が求められるのではないかと考え、シミュレーションを行ったのでこれを報告する。

2. 着想とシミュレーション方法

「プログラムをランダムに生成し、因子の組合せに対して、不具合が発見される比率を調べる」ことが可能な枠組みを検討するにあたり、

(1) 生成されるプログラムの枠組み

(2) 不具合発見の検出

をどのように表現するかを決める必要がある。上記 (1) については、原理的には、プログラミング言語の構文にあったプログラムを生成することは可能であるが、意味論的に正しいプログラムだけを生成することは難しい。また、さらに不具合を組み込んで、項目 (2) でそれを検出することも難しい。

我々は、プログラムを図 1 のように、近年のモデルベース開発などでよく使われるブロック・結線モデルで表現することを考えた。ここでブロックは機能を表し、結線はデータの流を表すものとする。このようなデータフローのモデルで任意の帰納的関数（即ち任意のプログラム）が表現できることは保障されている。

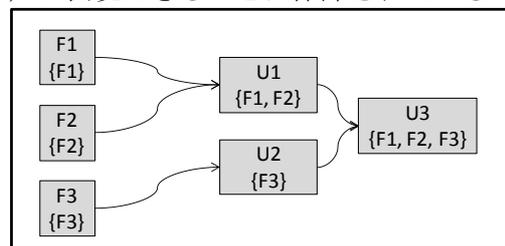


図 1 ブロック結線モデル

次に不具合の組込みと検出であるが、これは実際に不具合を組み込むのではなく、各ブロックがどの因子の組合せに依存しているかを集計する方法で近似した。図 1 の最も左側のブロック F1, F2, F3 は因子を表し、各ブロックの名前の下にはそのブロックが、直接、間接を含めて、

どの因子に依存しているかを書いてある。例えば、ブロック U1 は因子 F1 と F2 に依存しているため、ここに誤りがあれば、因子 F1 と F2 のすべての値の組み合わせを試すことで不具合が現れるはずである。従って、ランダムにブロック・結線モデルを生成したとき、全体のブロックのうち n 因子に依存するブロックの比率を集計することで、 n 因子の組合せで発見できる不具合の比率を求められることが期待できる（厳密には値の全組合せを試さなくても不具合が現れることがあるため不具合発見率の下限である）。

3. シミュレーションの内容

ここでは次のような方法でサイクルの無いブロック・結線モデルの生成を行った場合の挙動を詳しく報告する。

1. N を因子の数とする
2. M を生成するブロックの数とする
3. 因子に 1 から N までの番号を振り、ブロックに $N+1$ から $N+M$ までの番号を振る。
4. ブロック $N+1$ から $N+M$ までの各ブロックに対して自分より小さいブロックあるいは因子を 1 個から K 個選び、それらから自分への結線を結ぶ

このような方法でブロック・結線モデルをランダムに多数生成し、 n 因子に依存するブロックの生成比率を調べた。

4. 結果の考察

図 2 に因子数=40 とし、ブロック数を 40 個（因子を含まず）から 200 個まで 40 ずつ増やしながら、全体のブロック数に対する n 因子以下に依存するブロックの比率をプロットしたグラフを示す。各ブロックには 2 個以下の結線が入るとし、ブロック・結線モデルの生成はそれぞれ 1,000 回試行して平均をとった。

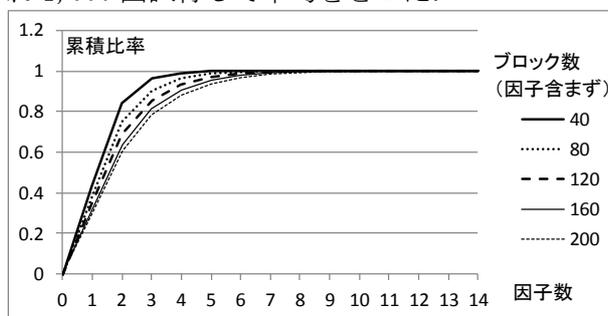


図2 n 因子以内に依存するブロック数の比率

ブロック数=40, 因子数=2 での平均は 0.844, 標準偏差は 0.073, 因子数=3 での平均は 0.961, 標

準偏差は 0.039 であり、通常、ペアワイズテストに期待される常識的な値がでている。

図 2 のグラフから、組合せる因子数 n が増えると、 n 以下の因子に依存するブロック数が急速に 1.0 に近くなる（飽和する）という、ペアワイズテストの経験則と合致する結果が得られた。また、ブロック数を多くすると、飽和が遅くなり、必要な因子数は次第に多くなっていくことから、複雑なソフトウェアのテストでは多くの因子の組合せが必要なことが分かる。

我々はここで報告した結果以外にも、(1) サイクルを許した場合のシミュレーション、(2) 結線の本数を増やした場合のシミュレーションを行った。サイクルを許した場合は、少ない結線の本数でブロックが依存する因子が増えるのでグラフの飽和は遅くなった。サイクルは、再帰呼出、ループ、結果のフィードバックなどに対応するので、これらが多用されているシステムでは多くの因子の組合せを考慮する必要があると思われる。また、結線の本数を増やした場合も、ブロックが依存する因子の個数が急速に増えるため、グラフの飽和が遅くなる。

5. おわりに

本報告ではブロック・結線モデルで表現されたプログラムをランダムに生成することで、ペアワイズテストにおける、少ないテストケースで不具合が発見できるという効果をシミュレーションで示した。

今後、本手法を適切な因子数の決定に利用するために、例えば、ブロック数と結線数の第一次近似としてプログラム行数、各行における変数や関数の参照数を考察し、本シミュレーション結果と実際のデータの照合を行うことで、本手法の検証と見積もりのための指標の検討を行っていく所存である。

参考文献

- [1] 川上真澄, 他: "デジタル家電に対する All-Pair テストの改善と適用", JaSST2006, 2006
- [2] D. R. Kuhn et. al.: "Software Fault Interactions and Implications for Software Testing", IEEE Trans. Software Eng., vol. 30, pp. 418-421, 2004.
- [3] C. Song, et. al., "iTree: Efficiently Discovering High-Coverage Configurations Using Interaction Trees", Proceedings of the 2012 International Conference on Software Engineering, pp903-913, 2012