線形計画法を用いたテスト集合圧縮アルゴリズムの一評価方法

3U－4

吉田清明　朱雀保正
久留米工業大学工学部

1．はじめに

著者らは先に，故障表から多項式時間内にサイズ（テスト入力数）の小さなテスト集合を生成するアルゴリズム[1]を提出した．このアルゴリズムにより生成されたテスト集合のサイズの評価は，これをその最小値と比較することにより行った．最小値は最初に与えられた故障表のサイズをm，近似アルゴリズムにより生成されたテスト集合のサイズをnとすると${}_mC_{n-1}$，${}_mC_{n-2}$…の組合せのテスト集合に対して充足解が存在するか否かを調べることにより求めた．しかしながらこの判定方法は，計算量のオーダーが階乗であるため現実的ではない．そこで本稿では，線形計画法（以下，ＬＰと略す）を用いたサイズの評価法を提出する．この評価法は，与えられた故障表の最小テスト集合のサイズの下界を与えるので，任意のアルゴリズムにより生成されたテスト集合サイズの定量的評価に応用できる．

2．故障表と最小テスト集合

故障表や最小テスト集合について簡単に説明しておく．表１のように入力パターンの印加に対して，単一縮退故障の発生している回路が正常信号値を出力するか故障信号値を出力するかの対応関係を，正常信号値の場合は0，故障信号値の場合は1を用いて示した表を故障表(fault table)と呼ぶ．どの故障の列にも少なくとも１つの１が存在するテスト入力の組み合わせのうち，その数が最小となるものを最小テスト集合という．表１の場合は，

表１ 故障表の例

テスト入力	代表故障										
	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11
T1	0	0	0	0	0	0	1	0	0	0	0
T2	0	0	0	0	0	1	0	1	1	0	1
T3	0	0	0	0	0	0	1	0	0	0	0
T4	0	0	0	1	0	1	0	1	1	0	1
T5	0	0	0	0	0	0	1	0	0	0	0
T6	0	0	0	0	1	1	0	1	1	0	1
T7	0	1	0	0	0	0	1	0	0	0	0
T8	0	1	1	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	1	1	0	0	0
T10	0	0	0	0	0	1	0	0	0	0	1
T11	0	0	0	1	0	0	1	1	0	0	0
T12	0	0	0	0	0	1	0	0	0	0	1
T13	0	0	0	0	1	0	1	1	0	0	0
T14	0	0	0	0	0	1	0	0	0	0	1
T15	1	0	1	0	0	0	0	0	0	1	1
T16	1	0	0	0	0	0	0	0	0	0	1

Application of LP Method to the Evaluation of the Test Sets Compression Algorithm
Kiyoaki YOSHIDA and Yasumasa SUJAKU
Kurume Institute of Technolog, 2228 Kamitsu-machi, Kurume, Fukuoka 830, JAPAN

｛T4, T6, T7, T15｝, ｛T6, T7, T11, T15｝,
｛T4, T7, T13, T15｝, ｛T4, T8, T13, T15｝,
｛T6, T8, T11, T15｝
の最小テスト集合が存在する．

3．テスト集合サイズの下界

表１のような故障表において，その最小テスト集合を求める問題は，一般に最小被覆問題と呼ばれ[2]，真を１偽を０とおくと

$$(\delta_{11}\wedge T_1\vee\delta_{21}\wedge T_2\vee\cdots\vee\delta_{i1}\wedge T_i\vee\cdots\vee\delta_{n1}\wedge T_n)$$
$$\wedge(\delta_{12}\wedge T_1\vee\delta_{22}\wedge T_2\vee\cdots\vee\delta_{i2}\wedge T_i\vee\cdots\vee\delta_{n2}\wedge T_n)$$
$$\vdots$$
$$\wedge(\delta_{1m}\wedge T_1\vee\delta_{2m}\wedge T_2\vee\cdots\vee\delta_{im}\wedge T_i\vee\cdots\vee\delta_{nm}\wedge T_n)=1 \quad (1)$$

ただしδ_{ij}は与えられた故障表のｉ行ｊ列の値，

を展開して得られる項の中のT_iの数の最小値を求める問題と等価である．この問題は「式(1)を満足し$\sum_{i=1}^{n}T_i$を最小にするようなT_iの集合を求める」と言い換えられる．一般に論理命題

$p : T_1\vee T_2\vee T_3\vee\cdots$

が真のとき，真を１偽を０とおくと

$q : T_1+T_2+T_3+\cdots\geq 1$

が成り立つ．また逆も成り立つ．このときT_iは０または１の値をとるが，これを$1\geq T_i\geq 0$に拡張することにより，表１のような故障表のテスト集合サイズの下界を求める問題を以下のようなＬＰの問題に帰着できる．

目的関数：$\sum_{i=1}^{n}T_i \rightarrow \mathrm{min}$

制約条件：$1\geq T_i\geq 0$，$\sum_{i=1}^{n}\delta_{ij}\cdot T_i\geq 1 \quad (j=1,m)$．

4．例題

表１の故障表のテスト集合サイズの下界を本稿の評価法で求める．目的関数zは

$$z=\sum_{i=1}^{16}T_i,$$

制約条件式は表１よりテスト入力間あるいは代表故障間の被覆情報を利用して圧縮することにより

$1\geq T_i\geq 0\,(i=1,16)$，$T_7+T_8\geq 1$，$T_{15}\geq 1$
$T_4+T_{11}\geq 1$，$T_6+T_{13}\geq 1$，$T_2+T_4+T_6\geq 1$，
$T_1+T_3+T_5+T_7+T_9+T_{11}+T_{13}\geq 1$

となり，目的関数zは最小値$z_{min}=4$の値をとる．

このときT_iの値はT_1からT_{16}まで左から順に{0,0,0,1,0,0,1,0,0,0,0,0,1,0,1,0}の整数解となり，$z_{min}=4$は表1のテスト集合サイズの下解であると同時に下限であることが判る

5．実験結果

表2のような数十ゲートの小規模な組合せ回路に同時故障シミュレーション法を適用し，表3の故障表1を生成した．シミュレーション回数は1000回（ただし故障検出率100%が得られたらそこで打ち切る）とし，与える入力パターンは1発生確率0.5の擬似ランダムパターンにより生成した．また同一の入力パターンが生成された場合は，これを除外した．同時故障シミュレーションは，故障表を作成するため既検出故障を除外せずに行った．表4において「同時」は表3の回路データに通常の同時故障シミュレーション法を適用して得られたサイズである．「近似」は表3の故障表1に[1]の近似アルゴリズムを適用して得られたサイズである．「下限」は故障表2に対して，テスト集合の初期値をm，「近似」で得られたサイズをnとして${}_mC_{n-1}$，${}_mC_{n-2}$...の組合せのテスト集合に対して充足解が存在するか否かを調べることにより得られた下限である．「下限」において故障表1をそのまま用いたのでは手に負えないので，それをテスト入力や故障の被覆の関係で圧縮した故障表2を用いた．「ＬＰ」は本稿の方法に従って，シンプレックス法[3]を故障表2に適用して得られたサイズの下界である．表3，表4ともに（）内は必要とした処理のCPU時間（単位：sec）の合計である．また「－」は3日で計算を打ち切ったことを示す．計算機はsun4/1000（主記憶256MB）を使用した（ただしマルチスレッドは不使用）．表4より例えば#7の回路の場合，下界が10.7であるから下限は11以上の22（故障表1の行の値）以下の整数値であることが判る．事実，「近似」によりサイズ11のテスト集合が得られている．

表2 ベンチマーク用小規模回路の特性

回路名	ゲート数（個）	外部入力端子数（個）	外部出力端子数（個）	内部信号線数（個）	代表故障数（個）
#1	31	11	3	120	137
#2	19	9	5	75	83
#3	31	9	4	85	83
#4	31	8	2	83	96
#5	35	6	8	115	122
#6	46	9	2	129	132
#7	63	14	8	200	237
#8	21	15	1	64	66
#9	70	9	5	177	182
#10	77	8	5	158	152
#11	33	11	3	111	123
#12	19	9	5	75	83
#13	29	9	5	97	103
#14	36	9	5	109	128
#15	42	15	14	163	172

表3 故障表の特性 ()はCPU時間，単位はsec

回路名	検出率（%）	故障表1（行×列）	故障表2（行×列）
#1	100.00	193×137(1.1)	25×25(14.3)
#2	100.00	222×83(0.5)	11×11(15.8)
#3	100.00	271×83(0.9)	15×15(14.5)
#4	65.63	251×63(1.2)	14×13(6.6)
#5	95.90	64×117(0.5)	12×12(2.2)
#6	100.00	22×132(0.1)	14×14(0.5)
#7	100.00	136×237(0.9)	74×50(15.9)
#8	86.36	989×57(2.4)	20×20(113.2)
#9	100.00	75×182(0.4)	18×18(7.9)
#10	96.05	251×146(1.5)	193×31(18.5)
#11	100.00	123×123(0.4)	24×24(5.9)
#12	100.00	222×83(0.5)	11×11(15.7)
#13	100.00	403×103(1.3)	18×18(38.2)
#14	100.00	93×128(0.3)	13×13(3.7)
#15	97.67	989×168(4.6)	251×65(218.1)

表4 実行結果 ()はCPU時間，単位はsec

回路名	同時	近似	下限	ＬＰ
#1	35(0.2)	25(0.7)	25(60.1)	25(14.4)*
#2	27(0.2)	11(0.4)	11(15.8)	11(15.8)*
#3	26(0.2)	16(0.6)	15(14.5)	15(14.5)*
#4	16(0.4)	11(0.8)	11(6.7)	11(6.6)*
#5	18(0.2)	12(0.4)	12(2.2)	12(2.2)*
#6	15(0.1)	12(0.1)	11(0.6)	10.7(0.5)
#7	34(0.2)	17(0.7)	-	15.9(18.1)
#8	30(3.6)	20(2.1)	20(114.7)	20(113.3)*
#9	28(0.1)	19(0.4)	18(8.4)	18(7.9)*
#10	17(0.6)	11(1.3)	-	11(20.3)*
#11	30(0.1)	24(0.4)	24(28.9)	24(6.0)*
#12	27(0.2)	11(0.3)	11(15.8)	11(15.7)*
#13	28(0.5)	18(1.1)	18(38.6)	18(38.2)*
#14	17(0.1)	13(0.3)	13(3.9)	13(3.7)*
#15	44(2.0)	17(4.3)	-	15.25(237.7)

*は下限（整数解）が求まったことを示す．

6．おわりに

与えられた故障表における最小テスト集合サイズの下界をＬＰを用いて求める方法を提案した．これにより故障表を用いる任意のテスト集合圧縮アルゴリズムから生成されたテスト集合のサイズの定量的評価が可能となった．現在，どのような条件が整った場合に下限（整数解）が得られるかを研究中である．

文献

[1] 吉田，朱雀，元石：故障検出数による重み付けを用いたテスト集合圧縮法に関する研究，九州大学大型計算機センター計算機科学研究集会報告第12号，pp.31-39(1995)．

[2] M.Garey and D.Jonson: *Computers and Intractability, A Guide to the Theory of NP-completeness*, W.H.Freeman and Co. (1979).

[3] 茨木，福島：FORTRAN77 最適化プログラミング，岩波書店(1983)．