並列局所探索法における近傍サイズの最適決定方式とその組合せ回路テスト生成への応用

夏目 幸一郎† 畠山 一実† 伊達 博†

局所探索法は組合せ最適化問題を解く発見的手法のひとつであり，可能解 x の近傍にあるより良い可能解 y を x に代入するという処理を行う．本論文では，特別な組合せ最適化問題を局所探索法で解く場合において，可能解 x をできるだけ速く最適解に近づけることを考える．その速度を理論的に求め，それを最大にする近傍サイズの近似式を導く．さらに，その結果を応用した組合せ回路の並列テストパターン生成手法を提案する．

Optimization of Neighborhood Size for Parallel Local Search and Its Application to Test Generation of Combinational Circuits

KOICHIRO NATSUME,† KAZUMI HATAYAMA† and HIROSHI DATE†

The local search is a heuristic approach to solve a combinational optimization problem. It includes the step of searching a better feasible solution y in the neighborhood of a feasible solution x and substituting y for x. In this paper, we consider an approach to bring a feasible solution x close to the optimum solution as fast as possible in the case of solving specific combinational optimization problem by the local search. We analyze the speed theoritically and present an approximation of optimal neighborhood size which maximize the speed. Moreover we apply the result to the parallel test generation of combinational circuits.

1. はじめに

局所探索法は組合せ最適化問題を解く発見的手法のひとつである．初期可能解 x_0 を生成して x に代入し，x の近傍 $N(x)$ 内により良い可能解 y があれば y を x に代入して同様の処理を繰り返す．x の近傍 $N(x)$ 内により良い可能解がない場合には x を出力して処理を終了する．

これを並列化する手法には，最良優先探索法とマルチ山登り法がある[1)]．最良優先探索法では，近傍探索を並列実行し，より良い可能解のうち最良のものを x に代入する．マルチ山登り法では，各プロセッサが独立に局所探索法を実行し，それらの結果のうち最良のものを最適解として出力する．

本論文では，可能解 x に対して決まる目的関数値から最適解とのハミング距離が分かるような組合せ最適化問題を扱う．このような問題を最良優先型局所探索法で解くことを想定し，可能解 x をできるだけ速く最適解に近づけることを考える．その速度を理論的に求め，それを最大にする近傍サイズを上記ハミング距離を用いて表す．さらに，その結果が文献 6) 記載の組合せ回路用並列テストパターン生成アルゴリズムに応用できると考えられる根拠を示す．文献 6) のアルゴリズムは近傍サイズを 1 に固定しているので，この結果を応用し，近傍サイズを最適決定することにより処理時間を短縮できると予想される．

† 株式会社日立製作所 日立研究所
Hitachi Research Laboratory, Hitachi, Ltd.

2. 組合せ最適化問題とそのためのハードウェア構成およびアルゴリズム

以下，本論文で議論の対象とする組合せ最適化問題と，それを解くためのハードウェア構成およびアルゴリズムを示す．なお，これらの仮定が組合せ回路の並列テストパターン生成用の最適化問題にあてはまることは，4 章で説明する．

(1) 組合せ最適化問題

目的関数

$$f(x) = f(x_1, x_2, \cdots, x_n) \rightarrow \text{最小}.$$

制約条件 $x \in \{0,1\}^n$.

問題に関する仮定

(a) 目的関数値 $f(x)$ を最小にする最適解 x はただ 1 つしかない．

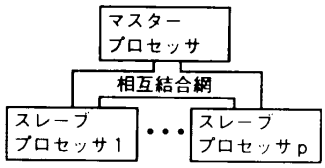

図1 ハードウェア構成
Fig. 1 Hardware composition.

(b) 可能解 $x=(x_1,x_2,\cdots,x_n)$ と最適解とのハミング距離 h（以下，FO 距離と呼ぶ）と，目的関数値 $f(x)$ との間には，次式の関係がある（g は単調非減少関数）．

$$h=g(f(x)). \tag{2.1}$$

(c) 処理時間の大部分は，目的関数値の計算が占めている．したがって，プロセッサ間の通信など，それ以外のことに要する時間は無視できる．

(2) ハードウェア構成

使用するプロセッサは全部で $(p+1)$ 台であり，そのうち1台は並列処理全体を管理するマスタープロセッサ，それ以外は近傍探索用のスレーブプロセッサである（**図1**）．スレーブプロセッサの数は，数台から数十台を想定する．

(3) アルゴリズム

(S1) マスタープロセッサは，初期可能解 $FS_0\in\{0,1\}^n$ をランダムに生成して $FS\leftarrow FS_0$ とする．FS の目的関数値 $f(FS)$ を計算して Min $\leftarrow f(FS)$ とする．

(S2) マスタープロセッサは，式(2.1)を用いて可能解 FS の FO 距離 h を計算する．$h=0$ ならば可能解 FS を出力し，終了．

(S3) マスタープロセッサは，近傍サイズ r を求める．可能解 FS と近傍サイズ r を各スレーブプロセッサに送信する．

(S4) 各スレーブプロセッサは，可能解 FS の成分をランダムに r 個選んで0になっているものを1に，1になっているものを0に変更した，変更解 CS を t 個生成する．それらの目的関数値を計算する．

(S5) 各スレーブプロセッサは，t 個の変更解のうち，目的関数値が最小のものを1つ選び，目的関数値とともにマスタープロセッサに送信する．

(S6) マスタープロセッサは，送信された p 個（p はスレーブプロセッサ数）の変更解のうち，目的関数値が最小のもの BS を1つ選ぶ．$f(BS)<$ Min ならば $FS\leftarrow BS$，Min $\leftarrow f(BS)$ とする．

(S7) (S2)へ．

3. 近傍サイズの最適値の決定

本章では，可能解 FS の FO 距離 h は既知であるものと仮定して，近傍サイズ r と可能解更新頻度 t の最適値を求める．本章の最後にこれらを最適決定したアルゴリズムを示す．

3.1 平均・FO 距離減少速度の定義

上記アルゴリズムにおいては，可能解 FS ができるだけ速く最適解に近づく，すなわち，(S2)と(S7)における FO 距離 h の差の期待値が，(S2)から(S7)までの所要時間の割に大きくなることが望ましい．(S2)から(S7)までの所要時間は，前章の仮定のもとでは可能解更新頻度 t に比例する．そこで，次のように定義する．

〈定義1〉

近傍サイズ r と可能解更新頻度 t は整数で，$1\le r\le n$，$t\ge 1$ とする．このとき，次式の値を平均・FO 距離減少速度と呼ぶ．

$$\begin{aligned}v&=v(r,t)\\&=((\text{S2})\text{における}\ FO\ \text{距離}\ h-(\text{S7})\text{における}\\&\quad FO\ \text{距離}\ h_{\text{NEW}}\ \text{の期待値})/t.\end{aligned}$$

以下，定義1による平均・FO 距離減少速度 v の値を最大にするような近傍サイズ r の近似値と可能解更新頻度 t の値を決定することを目標にして議論を進める．

3.2 平均・FO 距離減少速度の計算

本節では，前節で定義した平均・FO 距離減少速度 v の表現式を導く．ただし，$j<0$ または $j>i$ の場合には ${}_iC_j=0$ とする．また，$[x]$ は x を越えない最大の整数を表す．

まず，本アルゴリズムの(S4)において，可能解 $FS=(FS_1,FS_2,\cdots,FS_n)$ の成分をランダムに r 個選んで0になっているものを1に，1になっているものを0に変更することにより，変更解 $CS=(CS_1,CS_2,\cdots,CS_n)$ を生成したとする．このとき，最適解 $OS=(OS_1,OS_2,\cdots,OS_n)$ に対して，$FS_i=OS_i$ かつ $CS_i\neq OS_i$ である添字 i の個数を X_{CS} として $X_{CS}=k$ となる確率 $P(X_{CS}=k)$ を求める．

n 個（n は決定変数の個数）の添字の中から r 個を選び出す組合せは ${}_nC_r$ 通りある．$FS_i=OS_i$ である添字は $(n-h)$ 個あり，それらの中から k 個の添字を選び出す組合せは ${}_{n-h}C_k$ 通りある．また，$FS_i\neq OS_i$ である添字は h 個あり，それらの中から $(r-k)$ 個の添字を選び出す組合せは ${}_hC_{r-k}$ 通りある．したがって，求める確率は

$$P(X_{CS}=k)=\frac{{}_{n-h}C_k\cdot{}_hC_{r-k}}{{}_nC_r}$$

となる．また，このとき，変更解 CS の FO 距離は $h-(r-2k)$ となる．

次に，pt 個（p はスレーブプロセッサ数，t は可能解更新頻度）生成される変更解 CS のうち，目的関数値が最も小さいものを $BS=(BS_1,BS_2,\cdots,BS_n)$ とする．$FS_i=OS_i$ かつ $BS_i\neq OS_i$ である添字 i の個数を X_{BS} として，$X_{BS}=k$ となる確率 $P(X_{BS}=k)$ を求める．BS は，pt 個生成される変更解のうち FO 距離が最も小さいものでもあるので，

$$\begin{aligned}&P(X_{BS}=k)\\&=P(X_{BS}\geq k)-P(X_{BS}\geq k+1)\\&=\{P(X_{CS}\geq k)\}^{pt}-\{P(X_{CS}\geq k+1)\}^{pt}\\&=\left(\sum_{i=k}^{r}\frac{{}_{n-h}C_i\cdot{}_hC_{r-i}}{{}_nC_r}\right)^{pt}\\&\quad-\left(\sum_{i=k+1}^{r}\frac{{}_{n-h}C_i\cdot{}_hC_{r-i}}{{}_nC_r}\right)^{pt}\end{aligned}$$

となる．

次に，平均・FO 距離減少速度 v を求める．(S7) における Min の値が (S2) におけるものを下回らない場合（この場合，$X_{BS}\geq r/2$）には可能解 FS が更新されないので $h_{NEW}=h$ となり，そうでない場合（この場合，$X_{BS}\leq r/2$）には可能解 FS が更新されるので $h_{NEW}=h-(r-2X_{BS})$ となる．ゆえに，平均・FO 距離減少速度 v の値に寄与するのは $X_{BS}<r/2$ すなわち $X_{BS}\leq[(r-1)/2]$ の場合だけなので，

$$\begin{aligned}&v=v(r,t)\\&=\frac{1}{t}\sum_{k=0}^{[(r-1)/2]}(r-2k)P(X_{BS}=k)\\&=\frac{1}{t}\sum_{k=0}^{[(r-1)/2]}(r-2k)\left\{\left(\sum_{i=k}^{r}\frac{{}_{n-h}C_i\cdot{}_hC_{r-i}}{{}_nC_r}\right)^{pt}\right.\\&\quad\left.-\left(\sum_{i=k+1}^{r}\frac{{}_{n-h}C_i\cdot{}_hC_{r-i}}{{}_nC_r}\right)^{pt}\right\}.\end{aligned}\tag{3.1}$$

ここに，

v：平均・FO 距離減少速度，

r：近傍サイズ，

t：可能解更新頻度，

n：決定変数の個数，

p：スレーブプロセッサ数，

h：現在解の FO 距離，

となる．

3.3 最適近傍サイズに関する考察の準備

以下，近傍サイズ r と可能解更新頻度 t の最適値について理論的な考察を行うのに必要な，記号を定義し補題を述べる．

〈**定義 2**〉 n, p, h を整数の定数とし，$n\geq 1$, $p\geq 1$, $1\leq h\leq n$ とする．このとき，x と y の 2 変数関数 $A(x,y)$, $B(x,y)$, $C(x,y)$ を次式で定義する．ただし，x と y は整数で，$1\leq x\leq n$, $y\geq 0$ とする．また，$y>x$ の場合には $B(x,y)=0$ とする．

$$A(x,y)=\frac{{}_{n-h}C_y\cdot{}_hC_{x-y}}{{}_nC_x}.\tag{3.2}$$

$$B(x,y)=\sum_{z=y}^{x}A(x,z).\tag{3.3}$$

$$C(x,y)=1-\{B(x,y)\}^p.\tag{3.4}$$

〈**補題 1**〉 $M\geq 0$ とする．$X_m(0\leq m\leq M)$ と $Y_m(0\leq m\leq M)$ は $(M+1)$ 項から成る有限数列で，次の条件を満たすものとする．

$$X_m>0\quad(0\leq m\leq M).\tag{3.5}$$

$$\sum_{j=0}^{M}X_j=\sum_{j=0}^{M}Y_j=S.\tag{3.6}$$

$$\frac{Y_0}{X_0}\geq\frac{Y_1}{X_1}\geq\cdots\geq\frac{Y_M}{X_M}.\tag{3.7}$$

(a) このとき，

$$\sum_{j=m}^{M}X_j\geq\sum_{j=m}^{M}Y_j\quad(1\leq m\leq M)\tag{3.8}$$

が成り立つ．

(b) さらに，$Y_0/X_0>Y_M/X_M$ ならば，

$$\sum_{j=m}^{M}X_j>\sum_{j=m}^{M}Y_j\quad(1\leq m\leq M)\tag{3.9}$$

が成り立つ．

3.4 可能解更新頻度の最適値

本節では，可能解更新頻度 t（t は整数で $t\geq 1$）の最適値（平均・FO 距離減少速度 v を最大にする値）$t_{opt}=1$ であることを理論的に示す．

〈**定理 1**〉 r（$1\leq r\leq n$ とする）を定数と見た場合，$v(r,t)$ は $t=1$ のときに最大値をとる，すなわち，$t_{opt}=1$ となる．

〈**略証**〉

(a) $1\leq r<h$ の場合

平均・FO 距離減少速度 v は

$$v(r,t)=2\sum_{k=1}^{[(r-1)/2]}\frac{1-\{B(r,k)\}^{pt}}{t}$$

$$+\{r-2[(r-1)/2]\}\frac{1-\{B(r,[(r-1)/2]+1)\}^{pt}}{t}$$

と表示できる．$1 \le k \le [(r-1)/2]+1$ であるすべての k に対して $0 \le B(r,k) < 1$ なので，この式は $(1-s^t)/t$ $(0 \le s < 1)$ の形をした項の重み付きの和となっている．各項は $t=1$ のときに最大値をとるので，v も $t=1$ のときに最大値をとる．

(b) $h \le r \le n$ の場合

平均・FO 距離減少速度 v は

$$v(r,t) = 2\sum_{k=1}^{[(2h-r-1)/2]}\frac{1-\{B(r,k+r-h)\}^{pt}}{t}$$
$$+\{r-2[(r-1)/2]\}\frac{1-\{B(r,[(r-1)/2]+1)\}^{pt}}{t}$$

と表示できる．$r-h+1 \le k \le [(r-1)/2]+1$ であるすべての k に対して $0 \le B(r,k) < 1$ なので，この式は $(1-s^t)/t$ $(0 \le s < 1)$ の形をした項の重み付きの和となっている．各項は $t=1$ のときに最大値をとるので，v も $t=1$ のときに最大値をとる．(証明終)

3.5 最適近傍サイズについての理論的検討

本節では，可能解更新頻度 $t=1$ としたときの近傍サイズ r（r は整数で $1 \le r \le n$）の最適値（平均・FO 距離減少速度 v を最大にする値）r_{opt} についての理論的結果を示す．ただし，最適な近傍サイズがただ1つに確定しない場合には，それらのうち最小のものを選ぶものとする．

〈**定理 2**〉 n と p を任意の正の整数とする．$1 \le h \le n/2$ であるすべての h に対して，$r_{opt} \le h$ となる．

〈**略証**〉 h と r $(1 \le h \le n/2,\ h < r < 2h)$ を定数とするとき，$\min(n-r,h)$ を M，$A(r,i+r-h)$ を X_i，$A(2h-r,i)$ を Y_i と見なすと，これらは補題1の (a) の仮定において，j を i と読みかえたものを満たすことを示す．

式 (3.5) と (3.6) が満たされることは容易に分かる．

$0 \le i \le \min(n-r,h)$ として，i の関数 $D(i)$ を次式で定義する．

$$D(i) = \frac{A(2h-r,i)}{A(r,i+r-h)}.$$

すると，計算により，

$$D(i) = \begin{cases} \dfrac{(2h-r)!(n+r-2h)!}{r!(n-r)!}\cdot\displaystyle\prod_{j=1}^{r-h}\frac{2h-r-i+j}{n-r-i+j} \\ \qquad\qquad (0 \le i \le 2h-r), \\ 0 \quad (2h-r < i \le \min(n-r,h)), \end{cases}$$

となる．

$D(i)$ は $0 \le i \le 2h-r$ の範囲では正の値をとる単調非増加関数 $(r-h)$ 個と正の定数との積の形をしており，$2h-r < i \le \min(n-r,h)$ の範囲では0である．ゆえに，$0 \le i \le \min(n-r,h)$ の範囲において単調非増加であり，式 (3.7) が満たされることが分かる．

ゆえに，補題1の (a) より，$1 \le k \le \min(n-r,h)$ である k に対して，

$$\sum_{i=k}^{\min(n-r,h)} A(r,i+r-h) \ge \sum_{i=k}^{\min(n-r,h)} A(2h-r,i)$$

となり，さらに，

$$C(r,k+r-h) \le C(2h-r,k)$$

となることも容易に分かる．

一方，$h < r < 2h$ である r に対して $2h-r < h < r$ であり，

$$v(r,1) = 2\sum_{k=1}^{[(2h-r-1)/2]} C(r,k+r-h) + \{r-2[(r-1)/2]\} \times C(r,[(2h-r-1)/2]+1+r-h),$$

$$v(2h-r,1) = 2\sum_{k=1}^{[(2h-r-1)/2]} C(2h-r,k) + \{r-2[(r-1)/2]\} \times C(2h-r,[(2h-r-1)/2]+1),$$

と表示できるので，そのような r に対しては $v(r,1) \le v(2h-r,1)$ である．

また，$r \ge 2h$ のときには $v(r,1)=0$ となる．

ゆえに，r_{opt} の定義より，$h < r_{opt} < 2h$ とも $r_{opt} \ge 2h$ ともなりえないので，$r_{opt} \le h$ である．

以上より，定理は証明された．(証明終)

〈**定理 3**〉 n と p を任意の正の整数とする．$n/2 < h \le n$ であるすべての h に対して，$r_{opt} \ge h$ となる．

〈**略証**〉 h と r $(n/2 < h \le n, 2h-n \le r < h)$ を定数とするとき，$\min(n-h,r)$ を M，$A(r,i)$ を X_i，$A(2h-r,i+h-r)$ を Y_i と見なすと，これらは補題1の (b) の仮定において，j を i と読みかえたものを満たすことを示す．

式 (3.5) と (3.6) が満たされることは容易に分かる．

$0 \le i \le \min(n-h,r)$ として，i の関数 $E(i)$ を次式で定義する．

$$E(i) = \frac{A(2h-r,i+h-r)}{A(r,i)}.$$

すると，計算により，

$$E(i)=\begin{cases}\dfrac{(2h-r)!(n+r-2h)!}{r!(n-r)!}\cdot\displaystyle\prod_{j=1}^{h-r}\frac{n+r-2h-i+j}{r-i+j} & (0\le i\le n+r-2h),\\ 0 \quad (n+r-2h<i\le \min(n-h,r)), & \end{cases}$$

となる．

$E(i)$ は $0\le i\le n+r-2h$ の範囲では正の値をとる単調非増加関数 $(h-r)$ 個と正の定数との積の形をしており，$n+r-2h<i\le\min(n-h,r)$ の範囲では 0 である．ゆえに，$0\le i\le\min(n-h,r)$ の範囲において単調非増加であり，式 (3.7) が満たされることが分かる．

さらに，$i=0$ のとき $E(i)>0$，$i=\min(n-h,r)$ のとき $E(i)=0$ である．

ゆえに，補題 1 の (b) より，$1\le k\le\min(n-h,r)$ である k に対して，

$$\sum_{i=k}^{\min(n-h,r)}A(r,i)>\sum_{i=k}^{\min(n-h,r)}A(2h-r,i+h-r)$$

となり，さらに，

$$C(r,k)<C(2h-r,k+h-r)$$

となることも容易に分かる．

また，$\min(n-h,r)<k\le[(r-1)/2]+1$ である k に対しては，

$$C(r,k)=C(2h-r,k+h-r)=1$$

となることを示すのもたやすい．

一方，$2h-n\le r<h$ である r に対して $r<h<2h-r$ であり，

$$v(r,1)=2\sum_{k=1}^{[(r-1)/2]}C(r,k)+\{r-2[(r-1)/2]\}\times C(r,[(r-1)/2]+1),$$

$$v(2h-r,1)=2\sum_{k=1}^{[(r-1)/2]}C(2h-r,k+h-r)+\{r-2[(r-1)/2]\}\times C(2h-r,[(r-1)/2]+1+h-r),$$

と表示できるので，$[(r-1)/2]+1\le\min(n-h,r)$ の場合にも $[(r-1)/2]+1>min(n-h,r)$ の場合にも $v(r,1)<v(2h-r,1)$ である．

また，容易に分かるように，$n/2<h\le n$ ならば $v(n,1)=2h-n$ であり，さらに $1\le r<2h-n$ の場合には $v(r,1)\le r$ となる．したがって，$n/2<h\le n$ かつ $1\le r<2h-n$ ならば $v(r,1)<v(n,1)$ である．

ゆえに，r_{opt} の定義より，$2h-n\le r_{opt}<h$ とも $1\le r_{opt}<2h-n$ ともなりえないので，$r_{opt}\ge h$ である．

以上より，定理は証明された．(証明終)

本論文ではスレーブプロセッサの数は数台から数十台を想定しているが，以下では，スレーブプロセッサ数 p が十分大きい場合における r_{opt} の値を調べる．

〈**定理 4**〉 n を任意の正の整数とする．p が十分大きい場合には，$1\le h\le n$ であるすべての h に対して $r_{opt}=h$ となる．

〈**略証**〉 まず，$B(x,y)$ の値を調べる．容易に分かるように，

(a) $x\ge h$ の場合

$0\le y\le x-h$ のときには $B(x,y)=1$，$x-h<y\le\min(n-h,x)$ のときには $0<B(x,y)<1$，$y>\min(n-h,x)$ のときには $B(x,y)=0$ である．

(b) $x<h$ の場合

$y=0$ のときには $B(x,y)=1$，$0<y\le\min(n-h,x)$ のときには $0<B(x,y)<1$，$y>\min(n-h,x)$ のときには $B(x,y)=0$ である．

この結果を用いて，$p\to\infty$ のときの $v(r,1)$ の値を調べる．$v(r,t)$ の定義より，

(c) $h\le r\le n$ の場合

$r-h\le[(r-1)/2]$ すなわち $r<2h$ のとき，$v(r,1)\to 2h-r$ となる．

$r-h>[(r-1)/2]$ すなわち $r\ge 2h$ のとき，任意の正の整数 p に対して $v(r,1)=0$ となる．

(d) $1\le r<h$ の場合

$v(r,1)\to r$ となる．

したがって，$p\to\infty$ としたときに $v(r,1)$ が近づく極限関数 $g(r)$ が定義できて，

$$g(r)=\begin{cases}r & (1\le r<h),\\ 2h-r & (h\le r<2h),\\ 0 & (2h\le r\le n),\end{cases}$$

と表示できる．

$g(r)$ は $r=h$ のときに最大値 h をとり，$r\ne h$ であるどんな h に対しても，$g(r)\le h-1$ である．

したがって，h に依存して決まる値である $p_0(h)$ をうまく選ぶことにより，次のことが成り立つようにできる．

$p\ge p_0(h)$ を満たすすべての p に対して，$v(r,1)$ は $r=h$ のときかつそのときに限り最大値をとる．

ゆえに，$p\ge\max p_0(h)$ (h の範囲は $1\le h\le n$) であるすべての p に対して，$r_{opt}=h$ $(1\le h\le n)$ となる．

ゆえに，定理は証明された．(証明終)

3.6 最適近傍サイズの近似式

本節では，数値実験に基づいて最適近傍サイズ r_{opt} の近似式を導く．

図2に，決定変数の個数 $n=30$，スレーブプロセッサ数 $p=2$ および15，可能解更新頻度 $t=1$ として FO 距離 h と最適近傍サイズ r_{opt} との関係を示した．黒丸が $p=2$ の場合を，白丸が $p=15$ の場合を示している．ただし，二重丸は黒丸と白丸が重なっていることを示している．n，p および t を一定とした場合，r_{opt} の値については次のような傾向が観察される．

(1) r_{opt} は $h=n/2$ の付近で最も急激に変化する．

(2) h が $n/2$ に上から近づくときよりも下から近づくときのほうが，r_{opt} は $h=n/2$ のときの値（n が奇数の場合には $h=(n-1)/2$ のときと $h=(n+1)/2$ のときの値との平均値）に速く近づく．

定理2～4および上記の性質(1)と(2)を満足させるため，

$$h_{st}=\frac{2h}{n}-1 \tag{3.10}$$

により FO 距離 h を (-1) 以上 $(+1)$ 以下の値に変換し，それを用いた

$$r_{a-opt}=\begin{cases}\dfrac{n}{2}\left\{1-(-h_{st})^{\frac{1}{\alpha_1 n^{\beta}p^{-\gamma}+1}}\right\} & (h<\frac{n}{2}),\\[2ex] \dfrac{n}{2}\left(1+h_{st}^{\frac{1}{\alpha_2 n^{\beta}p^{-\gamma}+1}}\right) & (h\geq\frac{n}{2}),\end{cases} \tag{3.11}$$

という式を r_{opt} の近似式として用いることにした．

式(3.11)に含まれる α_1，α_2，β，γ の値は，式(3.12)で与えられる r_{a-opt} の誤差の二乗和 S が最も小さくなるように決めた．ただし，式(3.12)において，n の範囲は $5\leq n\leq 35$ を満たす5の倍数，p の範囲は $2\leq p\leq 20$ を満たす整数，h の範囲は $1\leq h\leq n$ を満たす整数とする．

$$S=\sum_n\sum_p\sum_h(r_{a-opt}-r_{opt})^2. \tag{3.12}$$

最初は各定数 α_1，α_2，β，γ の変域を十分広くとって比較的粗い格子を設定し，その中から S を最小にするものを選択した．その後，徐々に変域を狭め，格子を細かくしていくことにより精度を上げていった．その結果，$\alpha_1=0.46$，$\alpha_2=0.53$，$\beta=0.67$，$\gamma=0.40$ と求まった．

式(3.11)にこれらの値を代入した

$$r_{a-opt}=\begin{cases}\dfrac{n}{2}\left\{1-(-h_{st})^{\frac{1}{0.46n^{0.67}p^{-0.40}+1}}\right\} & (h<\frac{n}{2}),\\[2ex] \dfrac{n}{2}\left(1+h_{st}^{\frac{1}{0.53n^{0.67}p^{-0.40}+1}}\right) & (h\geq\frac{n}{2}),\end{cases} \tag{3.13}$$

による h と r_{a-opt} の関係を図2の直線と曲線で示す．実線の曲線が $p=2$ の場合を，点線の曲線が $p=15$ の場合を，破線の直線が $p\to\infty$ とした場合を示している．

以上より，最適なアルゴリズムは2章(3)で述べたアルゴリズムの(S3)，(S4)および(S5)を，次のように変更したものになる．

(S3) マスタープロセッサは，式(3.10)と式(3.13)を用いて近傍サイズ r を求める．可能解 FS と近傍サイズ r を各スレーブプロセッサに送信する．

(S4) 各スレーブプロセッサは，可能解 FS の成分をランダムに r 個選んで0になっているものを1に，1になっているものを0に変更した，変更解 CS を1個生成する．その目的関数値を計算する．

(S5) 各スレーブプロセッサは，変更解とその目的関数値をマスタープロセッサに送信する．

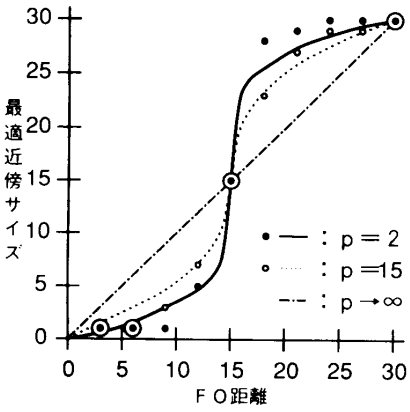

図2 FO 距離と最適近傍サイズ（$n=30$）
Fig. 2 FO metric and optimal neighborhood size.

4. 本手法の組合せ回路テスト生成への応用

図3に示した2つの組合せ回路C1とC2を考える．C1は通常の組合せ回路であるが，C2のNOTゲート（論理ゲートの種類については**図4**を参照）の出力線には故障があり，どのような場合にも信号値が1になってしまう．これらの2つの回路の間で出力信号値に違いが生じるような入力信号の組合せ（テストパターン）

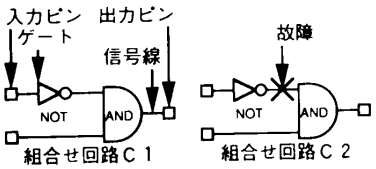

図3 組合せ回路の例
Fig. 3 Example of combinational circuit.

ゲート	出力信号値
NOT	入力値が0の場合 1 入力値が1の場合 0
AND	全入力値が1の場合 1 その他の場合 0
OR	全入力値が0の場合 0 その他の場合 1

図4 論理ゲートの種類と出力信号値
Fig. 4 Logic gate and output value.

を求めることを考える．ただし，テストパターンが複数個ある場合には，そのうち1つだけを求めればよいものとする．

この例題では，**表1**に示すように4通りの入力ベクトル（入力信号の組合せ）に対する出力信号値の異同を全部調べることにより，入力ベクトル (1,1) だけがテストパターンになっていることが分かる．

組合せ回路のテストパターンを生成する問題はNP完全なので[2]，テストパターンがあれば必ず求められる厳密解法[3),4)]には回路規模の多項式時間で終了する保証がなく，また，プログラムが複雑になる．そこで，入力ベクトルの *FO* 距離と相関のある目的関数を定義し，その値を最小にするものを発見的に探索する方法が提案されるようになった[5]．我々が以前に提案した手法もその1つであり，組合せ回路C1およびC2における実数値の出力値に基づいて目的関数値を計算し，それを $r=1$, $t=1$（r は近傍サイズ，t は可能解更新頻度）の局所探索法により最小化する逐次処理型の手法となっている[6]．

C1およびC2における実数値の出力値の計算手順を説明する．まず**図5**(a)のように，(i) 入力ピンにおいて，対応する入力ベクトルの成分が0の場合には実数値 ε を，1の場合には実数値 $1-\varepsilon$ を与え（ただし $0.0<\varepsilon<0.5$），(ii) 故障のある信号線には，その故障が信号値を0に固定する場合には実数値0を，1に固定する場合には実数値1を与える．次に，各論理ゲートに対して入力ピンから順番に図5(b)に示すゲート演算規則を適用して実数出力値を計算する．

表1 入力ベクトルと目的関数値
Table 1 Input vector and object function value.

入力ベクトル	出力信号値 C1	出力信号値 C2	テストパターンか？	目的関数値	*FO* 距離
(0,0)	0	0	NO	10000.00	2
(0,1)	1	1	NO	101.01	1
(1,0)	0	0	NO	101.01	1
(1,1)	0	1	YES	1.02	0

実数値を与える箇所		与える実数値
入力ピン	入力ベクトルの成分が0	ε
	入力ベクトルの成分が1	$1-\varepsilon$
故障箇所	信号値を0に固定する	0
	信号値を1に固定する	1

(a) 実数値出力値計算のために与える値

ゲート	演算規則
NOT	$1-a$
AND	$\prod_{i=1}^{m} a_i$
OR	$1-\prod_{i=1}^{m}(1-a_i)$

(b) ゲート演算規則

図5 実数値出力値計算規則
Fig. 5 Computation of real output value.

入力ベクトルの目的関数値は，その結果を用いて次式で求められる．ただし，出力値とは実数値出力値のことである．

$$\text{目的関数値} = \frac{1}{\sum_{\text{全出力ピン}} |\text{C1の出力値} - \text{C2の出力値}|}. \tag{4.1}$$

$\varepsilon=0.01$ として，図3の回路C1およびC2を考えた場合における，各入力ベクトルに対して決まる目的関数値と *FO* 距離を表1に示した．たとえば，入力ベクトル (0,0) に対しては，C1の実数値出力値は $(1-0.01)\times 0.01 = 0.0099$，C2の実数値出力値は $1\times 0.01 = 0.01$ と計算され，目的関数値は式 (4.1) より10000.00と求められる．また，*FO* 距離は2である．

組合せ回路の並列テスト生成に対する2章の条件 (a), (b) および (c) は，テスト生成の研究者から見ればとても自然なものである．以下，組合せ回路の並列テスト生成がこれらの条件を満たしていると思われる根拠を示す．

実用上のテスト生成は，前処理として多数（数千個程度）の入力ベクトルをランダムに発生して出力信号値の異同を調べ，それらの中にテストパターンが1つ

もない場合に限り，テスト生成用のアルゴリズムを適用するという手順をふむ．これは，出力信号値の計算には0および1の2値しか現れないので非常に精密な工夫を施すことができ，実数値出力値の計算時間に比べて無視できるほどの時間しかかからないからである．

このため，本手法の適用対象となるのはテストパターンが（複数個あるとしても）少ない回路ペアに限られる．テストパターンが少ないということは，式(4.1)を最小にする最適解が（複数個あるとしても）少数であるということである．したがって，仮定(a)は近似的に成り立つと考えられる．

また，表1を見ると，目的関数値が小さい入力ベクトルはテストパターンであり，目的関数値が大きい入力ベクトルはテストパターンでないので，目的関数値を最小化することによりテストパターンを求めることができることが分かる．また，FO 距離 h と目的関数値 c との間には近似的に，

$$h \fallingdotseq \log_{1/\varepsilon} c \tag{4.2}$$

の関係があり（ただし，ε は実数値出力値を計算する際に，対応する入力ベクトルの成分が0の場合に入力ピンに与える値），図3の回路ペアについて仮定(b)が成り立っていることも分かる．

さらに，我々は文献6)において式(4.1)を用いたテスト生成のプログラム実験を行い，数千個のゲートを含む回路に対しても実用時間内でテスト生成が可能なことを確認している．この結果より，仮定(b)は実用上の大規模回路に対しても近似的に成り立っていると推測される．

さらに，大規模な回路の場合には実数値出力値の計算に時間がかかると考えられるので，仮定(c)も成り立つと思われる．

以上より，最適なテスト生成アルゴリズムは次のようになる．

(S01) マスタープロセッサは，テスト生成対象の回路ペアの情報を読み込み，各スレーブプロセッサに送信する．

(S02) 各スレーブプロセッサは，多数(数千個程度)の入力ベクトルをランダムに発生して出力信号値の異同を調べ，テストパターンの有無をマスタープロセッサに送信する．テストパターンがあればそれも送信する．

(S03) マスタープロセッサは，テストパターンの有無を総合判定する．テストパターンがあれば出力し，終了．テストパターンがなければ(S04)へ．

(S04) マスタープロセッサは，初期入力ベクトル $FS_0 \in \{0,1\}^n$ をランダムに生成して $FS \leftarrow FS_0$ とする．式(4.1)を用いて FS の目的関数値 $f(FS)$ を計算して $\mathrm{Min} \leftarrow f(FS)$ とする．

(S05) マスタープロセッサは，式(4.2)を用いて入力ベクトル FS の FO 距離 h を計算する．$h=0$ ならば入力ベクトル FS を出力し，終了．

(S06) マスタープロセッサは，式(3.10)と式(3.13)を用いて近傍サイズ r を求める．入力ベクトル FS と近傍サイズ r を各スレーブプロセッサに送信する．

(S07) 各スレーブプロセッサは，入力ベクトル FS の成分をランダムに r 個選んで0になっているものを1に，1になっているものを0に変更した，変更ベクトル CS を1個生成する．式(4.1)を用いてその目的関数値を計算する．

(S08) 各スレーブプロセッサは，変更ベクトルとその目的関数値をマスタープロセッサに送信する．

(S09) マスタープロセッサは，送信された p 個（p はスレーブプロセッサ数）の変更ベクトルのうち，目的関数値が最小のもの BS を1つ選ぶ．$f(BS) < \mathrm{Min}$ ならば $FS \leftarrow BS$，$\mathrm{Min} \leftarrow f(BS)$ とする．

(S10) (S05)へ．

5. むすび

本論文では，可能解に対して決まる目的関数値から最適解とのハミング距離が分かるような組合せ最適化問題を扱い，最適近傍サイズを上記ハミング距離を用いて表した．さらに，その結果を組合せ回路の並列テストパターン生成に応用できると考えられる根拠を示した．

もし，2章(1)の仮定を満たす最適化問題がテストパターン生成用のもの以外にもあれば，本論文の議論はそれに対しても適用できる．

今後は，プログラム実験により本手法の高速化効果を確認する予定である．

参考文献

1) 石川幹人，十時　泰，戸谷智之，星田昌紀，広沢誠：並列反復改善法によるタンパク質の配列解析，情報処理学会論文誌，Vol.35, No.12, pp.2816-2830 (1994).

2) Ibarra, P.H. and Sahni, S.K.: Polynomially Complete Fault Detection Problems, *IEEE Trans. Comput.*, Vol.C-24, No.3, pp.242-249 (1975).

3) Roth, J.P., Bouricius, W.G. and Schneider, P.T.: Programmed Algorithms to Compute Tests to Detect and Distinguish between Fail-

ures in Logic Circuits, *IEEE Trans. Electron. Comput.*, Vol.EC-16, No.5, pp.567–579 (1967).
4) Goel, P.: An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits, *IEEE Trans. Comput.*, Vol.C-30, No.3, pp.215–222 (1981).
5) Cheng, K.-T. and Agrawal, V.D.: A Simulation-Based Directed-Search Method for Test Geneation, *Proc. Int. Conf. Comp. Des. (ICCD'87)*, pp.48–51 (1987).
6) 池田光二，畠山一実，林 照峯：バックトラック処理不要な組合せ回路テスト生成手法，信学技報，FTS89-35, pp.53–57 (1989).

(平成7年6月26日受付)
(平成8年3月12日採録)

夏目幸一郎（正会員）

平成元年筑波大学で数学を専攻して卒業．同年（株）日立製作所入社．以来，VLSI・CADの研究開発に従事．研究の必要上，計算量理論，アルゴリズム理論，組合せ最適化などに関心を寄せるに至る．現在，同社日立研究所所属．

畠山 一実（正会員）

昭和28年生．昭和51年京都大学工学部数理工学科卒業．昭和57年同大学院博士課程修了．工学博士．昭和57年4月（株）日立製作所に入社．以来，論理回路に対するテスト設計自動化の研究に従事．現在，同社日立研究所情報制御第1研究部主任研究員．IEEE，電子情報通信学会，日本オペレーションズリサーチ学会会員．

伊達 博（正会員）

昭和36年生．昭和60年九州大学理学部数学科卒業．昭和62年同大学院理学研究科修士課程（数学専攻）修了．同年（株）日立製作所入社．現在，(株) 日立製作所日立研究所所属．平成2～5年（財）新世代コンピュータ技術開発機構へ出向．工学博士．電子回路の設計自動化，特に並列処理を利用したVLSI-CADの研究開発に従事．電子情報通信学会，人工知能学会，ACM，IEEE各会員．