スタンダードセルレイアウト設計におけるセル配置改良をともなうタイミングドリブン端子割当てアルゴリズム

若林真一† 小出哲士†

本論文では，スタンダードセルレイアウト設計において，タイミングを考慮したうえで端子割当てとセル配置改良を同時に行うアルゴリズムを提案する．提案アルゴリズムの目的は，セル配置とセル間の概略配線経路が決定された後において，与えられたタイミング制約の下でチャネル密度と配線長を最小にする端子割当てとセル配置の改良を求めることである．提案アルゴリズムのレイアウトモデルにおいてはセル行は3行とし，中央のセル行を端子割当てと配置改良の対象とする．各セルに対する可能な端子割当てがたかだか k 種類で，配置改良後のセルは元の位置から相対的にたかだか r 個しか移動しないとするとき，提案アルゴリズムはセル数に比例する計算時間で最適解を求めることができる．さらに，提案アルゴリズムは簡単に一般のレイアウトモデルに対する端子割当てアルゴリズムに拡張できる．ベンチマークデータを用いた計算機実験の結果，提案アルゴリズムの有効性が示された．

A Timing-driven Pin Assignment Algorithm with Improvement of Cell Placement in Standard Cell Layout

SHIN'ICHI WAKABAYASHI† and TETSUSHI KOIDE†

In this paper, we propose a timing-driven pin assignment algorithm with improvement of cell placement in standard cell layout. The objective of the algorithm is to minimize the channel density as well as the total wire length by assigning nets to pins of cells under the given timing constraints. The layout model assumed in this paper consists of three cell rows, and the middle cell row is the one to which the proposed algorithm is applied. If the number of possible pin assignments for each cell is bounded by some constant k and the cell is allowed to be moved from its original position within r cells, then the proposed algorithm can solve the problem in linear time on the number of cells in the middle cell row. The proposed algorithm can easily be extended to a heuristic pin assignment algorithm to the general standard cell layout model. Experimental results with benchmark data show the effectiveness of the proposed algorithm.

1. はじめに

VLSIレイアウト設計の一方式であるスタンダードセルレイアウト設計においては，まずセルと呼ばれるあらかじめレイアウトパターンが設計された部分機能回路を複数の行を構成するように配置し，次に概略配線経路を求めるとともに配線に必要なフィードスルーの挿入を行い，最後に求めた概略配線経路に従ってセル行の間に確保されるチャネルと呼ばれる配線領域で詳細配線経路が決定される[10]．ここで，チャネルに面している各セルの端子の位置はチャネル配線を行う前に決定されている必要がある．一方，あるセルの複数の端子とそれらに接続されるネットの関係をレイアウト工程において変更することが許される場合が多いことが一般的である．たとえば，セルが2入力ANDゲートだったとすれば，2個の入力端子とそれらに接続されるネットの接続関係を入れ替えても論理的には等価なので，入力ゲート容量などの違いによるトランジスタ回路レベルでの不具合が生じない限り，これらの端子は入れ替え可能である．また，回路の論理的機能を変更することなく端子位置を変更できる別の場合として，セルの大きさや機能は同じで端子位置のみ違う複数のセルがセルライブラリから選択できる場合がある．特に，あるセルについて，そのレイアウトパターンを左右裏返し（フリッピングと呼ぶ）にしたものについては正しいレイアウトパターンと見なされることが一般的である．図1に2入力ANDゲートの場

† 広島大学工学部
Faculty of Engineering, Hiroshima University

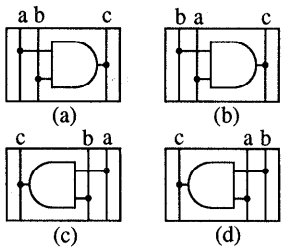

図1 2入力 AND ゲートの端子割当て
Fig. 1 Pin assignment of a 2-input AND gate.

合の端子入れ替え，およびフリッピングによるセル端子位置の変更例を示す．ここで図中の a，b，c はネットを示す．

チャネル配線の結果は各セルの端子位置に大きく依存する．このため，もし上述の理由によりチャネルに面する各セルの端子位置を変更できる場合，各セルの端子位置を最適に決定することによりチャネル配線の配線密度や配線長を最小化することができる．チャネル配線におけるセルの端子位置決定問題は一般にチャネル端子割当て（Channel Pin Assignment, CPA）問題と呼ばれ，これまでに多くの結果が報告されている．まず，端子位置の制約がない CPA 問題については文献 1)，2)，4)，9)，11) などの結果が知られている．しかし，これらの結果はセルの端子位置に制約のある場合には適用できない．一方，Hou と Chen はセルの端子位置は固定であるが，それらの端子とネットの接続関係は任意に変更可能である，という仮定の下で CPA 問題を解くアルゴリズムを提案した[7]．また，同じ著者により同じ仮定の下でセル上配線可能なチャネル配線に対する CPA 問題に対するアルゴリズムも提案されている[8]．しかしながらこれらのアルゴリズムは図1のように，あるセルに対して固定された数のレイアウトパターンのみの選択が許される場合の CPA 問題（このような問題をセルパターン選択問題と呼ぶ）には適用できない．

さて，チャネル配線においてはチャネルに面するセルの端子位置の変更だけでは配線長やチャネル密度の十分な改善を達成できない場合もある．このような場合，あらかじめ与えられるセルの配置を変更することによりさらなる改善を達成できる可能性がある．セルの配置改良とセルの端子割当ては密接な関係があるので，これらは同時に行うことが望ましい．さらに，近年のレイアウト設計においてはタイミングを考慮してセル配置を決定することが一般的なので，セル配置の改良や端子割当てを行う場合はタイミングを考慮することが望まれる．しかしながら，タイミングを考慮しながらセル配置の改良と端子割当てを同時に行う手法はこれまで知られていなかった．セルパターン選択問題に対しては Her と Wong[5] が多項式時間最適アルゴリズムを示しているが，このアルゴリズムはセル配置の改良は考慮しておらず，またタイミング制約も扱うことができない．同じく Her ら[6] はタイミングを考慮した端子割当て手法を提案しているが，やはりセル配置の改良は考慮されていない．

本論文では，スタンダードセルレイアウト設計に対し，概略配線終了後，タイミングを考慮しながらセル配置の改良と端子割当てを同時に行うアルゴリズムを提案する．本論文においては，各セルに対してセルに接続されるネットの端子への割当て方をセルパターンとして定義し，各セルにはあらかじめ定数個のセルパターンが与えられているものとする．また，セルに接続されるネットにはチャネル内における配線長の上限がタイミング制約として与えられているものとする．このとき，本論文で考察するタイミングドリブン端子割当て（Timing-Driven Pin Assignment, TDPA）問題とは，与えられたセル配置に対し，配線長とチャネル密度を最小とするようなセルパターンとセル位置を決定する問題とする．しかしながら，この問題は NP 困難である通常のセル配置問題を含み，効率良く解くことは非常に困難なので，ここでは以下の制約をおくことにより，問題を扱いやすくする．まず，与えられるセル配置のセル行数は3行とし，端子割当ておよびセル位置の改良は2行目のセル行のみを対象とする．また，任意のセルに対し，入力配置においてそのセルが左端から i 番目のセルとし，端子割当て後のセルの位置を j 番目とするとき，$i-r \leq j \leq i+r$ が成り立つものとする．ただし，r は0以上の定数とする．このように定義された端子割当て問題に対し，本論文ではセル数に比例する計算時間で効率良くセルの端子割当てとセル配置改良を求めるアルゴリズムを提案する．タイミング制約が与えられていない場合，この問題に対して提案アルゴリズムは最適な端子割当てとセル配置改良をセル数に比例する計算時間で求めることができる．提案アルゴリズムのレイアウトモデルは限定されているが，提案アルゴリズムを繰り返し適用することにより，一般のスタンダードセルレイアウトに対しても端子割当てと配置改良を実現できる．

本論文の構成を以下に示す．2章では準備としてレイアウトモデルと問題の定式化を行う．3章では提案アルゴリズムの詳細を述べる．4章では提案アルゴリズムに対する計算機実験による評価について述べる．最後に5章では本論文をまとめるとともに，今後の課

題についても触れる．

2. 準　　備

2.1 レイアウトモデル

スタンダードセルレイアウトにおいては，1つのチャネルは2つのセル行に挟まれた配線領域であり，逆にあるセル行は2つのチャネルに面している．このため，あるチャネルのチャネル密度とチャネル内の総配線長を最小化するためにはそのチャネルに面するセルのセル行内での配置，およびセルの端子位置を適切に決める必要がある．一方，セル行は一般に2つのチャネルに面しており，セルの端子に接続されたあるネットがこれらの2つのチャネルで配線される可能性がある．そこで本論文では1つのセル行とそれに隣接する2つのチャネルを対象とした端子割当て問題を定義する．

図2 に本論文におけるレイアウトモデルを示す．レイアウトは3行のセル行とその間にある2個のチャネルから構成されている．セル行は上から順に上部セル行，中央セル行，下部セル行と呼ぶ．また，上部セル行と中央セル行の間のチャネルを上部チャネル，中央セル行と下部セル行の間のチャネルを下部チャネルと呼ぶ．上部セル行，中央セル行，下部セル行に含まれるすべてのセルの集合を C で表す．また各セル $c \in C$ の幅を $width(c)$ で表す．セルとチャネルの配置は横方向を X 座標にとった XY 座標系で表すものとし，各セル c の左端と右端の x 座標をそれぞれ $lpos(c)$, $rpos(c)$ で表す．それぞれのセル行において，セルとセルの間に隙間はないものとする☆．3本のセル行の最左端のセルの中で，セルの左辺が最も左にあるセルを c' とするとき，$lpos(c') = 0$ とする．3本のセル行の最右端のセルの中で，セルの右辺が最も右にあるセルを c'' とするとき，$rpos(c'') = L$ とする．L をセル配置の配置幅と呼ぶ．

各セルはネットに接続される端子をセルの上辺もしくは下辺の境界上に持つものとする．端子の位置はセルの左端を0とした相対座標で表し，セルのレイアウトパターンで指定されるものとする．セルを実現するレイアウトパターンは複数ある場合があり，それぞれのレイアウトパターンで対応する端子位置が異なっていてもよいが，レイアウトパターンの幅と高さはそれぞれすべて同一とする．

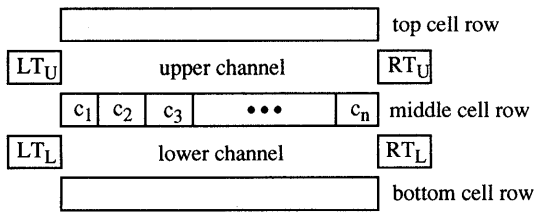

図2 レイアウトモデル
Fig. 2 The layout model.

☆ 与えられたセル配置のセル間に隙間がある場合は隙間と同じ幅で端子を持たない仮想的なセル（ダミーセル）を挿入する．ダミーセルも配置改良の対象となる．

セル $c \in C$ に接続されるネットの集合を $N(c)$ で表す．ここでセルに接続されるネットはセルの上に隣接するチャネル（上部チャネル）で配線されるか，セルの下に隣接するチャネル（下部チャネル）で配線されるかのいずれかであるとし，それぞれのネットの集合を $N_U(c)$, $N_L(c)$ で表す．また，中央セル行上を通過して上部セル行と下部セル行を直接接続するようなネットが存在した場合は，それらについては概略配線において中央セル行中にフィードスルーセルを挿入し，あらかじめ $N_U(c)$, $N_L(c)$ に属する2つのネットに分けておくものと仮定する．挿入されたフィードスルーセルも通常のセルと見なして提案アルゴリズムによるセル配置改良の対象とする．上部チャネルと下部チャネルの左端（$x = 0$）と右端（$x = L$）にはそれぞれ左外部端子セル（図2の LT_U, LT_L），右外部端子セル（図2の RT_U, RT_L）があり，それぞれのチャネルを左方向もしくは右方向に突き抜けるネットはこれらの端子セルに仮想的に接続されているものとする．

このとき，c を実現するあるレイアウトパターン M_c が存在して，与えられた回路の機能を正しく実現するという制約のもとに $N_U(c)$, $N_L(c)$ にそれぞれ含まれている各ネットをレイアウトパターン M_c の上辺もしくは下辺の境界に位置する各端子に割り当てることができるならば，そのようなネットの端子への割当て方をセルパターンと呼ぶ．すなわち，セル c のセルパターンはレイアウトパターン M_c と，そのレイアウトパターンの端子に対するネットの割当てを決めるマッピング f_c の2項組 $\langle M_c, f_c \rangle$ で表すことができる．一般にセルを実現するレイアウトパターンは複数あり，また1つのレイアウトパターンに対してネットから端子へのマッピングも複数ある場合もあるので（図1参照），セルパターンも複数ある場合が一般的である．セル c のセルパターンの集合を $CP(c)$ で表す．

2.2 チャネル密度と配線長

問題を定義するために，チャネル密度と配線長を定義する．以下ではすべてのセルについて，それぞれ1つのセルパターンが選択されていると仮定する．ここで Q を上部チャネルもしくは下部チャネルとし，$R1$

と $R2$ を Q の上辺と下辺に面するセル行とする．N_Q をチャネル Q で配線されるネットの集合とする．すべてのセルについてセルパターンが選択されているのでネットが接続される端子の位置も求めることができる．N_Q の各要素であるネット n について，ネット n のチャネル Q 内での最左端と最右端の端子の x 座標をそれぞれ $l(n)$，$r(n)$ とし，ネット n のスパン長 $span(n)$ を $span(n) = r(n) - l(n)$ と定義する．

このとき，任意の x 座標 h $(0 \leq h \leq L)$ に対して，座標 h における列密度 $d_Q(h)$ を

$$d_Q(h) = |\{n_i | n_i \in N_Q, \text{かつ}, l(n_i) \leq h \leq r(n_i), \text{かつ}, l(n_i) \neq r(n_i)\}|$$

と定義する．また，チャネル Q のチャネル密度を

$$D_Q = \max_{0 \leq h \leq L} d_Q(h)$$

と定義する．上部チャネルと下部チャネルのチャネル密度を D_U，D_L で表す．また，総チャネル密度 D_{total} を $D_{total} = D_U + D_L$ で定義する．

さらに，チャネル Q に対して予測チャネル配線長を

$$W_Q = \sum_{n_i \in N_Q} span(n_i)$$

と定義する．上部チャネルと下部チャネルの予測チャネル配線長を W_U，W_L で表す．また，総予測配線長 W_{total} を $W_{total} = W_U + W_L$ で定義する．

2.3 タイミング制約

スタンダードセル配置においてはタイミング制約を考慮する場合は一般にネット単位もしくは信号経路単位で考慮され，与えられたタイミング制約を満足する配置が得られる．本論文では，チャネル単位でネットを扱うため，前段階であるセル配置工程の結果に基づいてタイミング制約がチャネル内のネットに関する配線長制約にあらかじめ変換されているものと仮定する．すなわち，各ネット n のスパン長 $span(n)$ に対し，その最大許容スパン長 $maxspan(n)$ が制約として与えられているものとする．端子割当てと配置改良の結果，すべてのネット n に対して $span(n) \leq maxspan(n)$ であれば，タイミング制約が満足されたものとする．

2.4 問題の定式化

以上の準備を基に，本論文で対象とするタイミングを考慮した端子割当て問題（問題 TDPA）を以下のように定式化する．

[問題 TDPA]

入力 (1) セル配置（配線に必要なフィードスルーを含む）．
(2) 各チャネルのネットリスト．
(3) 各セルの初期セルパターン．
(4) 中央セル行の各セルのセルパターン集合．
(5) 各ネット n の最大許容スパン長 $maxspan(n)$．
(6) 定数 $Dmax_U$，r．

出力 (1) 中央セル行の各セルの位置．
(2) 中央セル行の各セルのセルパターン．

目的 制約を満足し，3 項組 $\langle D_{total}, D_U, W_{total} \rangle$ を辞書式順序（第 1 項が最も優先度が高い）の下で最小化するように中央セル行の各セルの配置を改良し，セルパターンを決定する．

制約 (1) $D_U \leq Dmax_U$．(2) 中央セル行の各セル c について，初期配置において c がセル行の左端から i 番目に位置していたものとし，出力される配置において c が左端から j 番目に位置していたとすれば，$|i-j| \leq r$．(3) 中央セル行の最左端と最右端の x 座標は変化しない．(4) すべてのネット n に対し，$span(n) \leq maxspan(n)$．

この定式化においては，上部セル行と下部セル行のセル位置とセルパターンは固定とし，中央セル行のセル位置とセルパターンを初期配置から変更することにより上部チャネルと下部チャネルの総チャネル密度と総予測配線長を最小化することを目的としている．ただし，チャネル密度と配線長を同時に最小化できない場合があるので，チャネル密度の最小化を優先させた定式化となっている．また，チャネル密度が同じ場合は上部チャネル密度の最小化を下部チャネル密度より優先させる定式化となっている．制約 (1) は，本問題に対する手法を一般のレイアウトに適用する場合に上部のセル行から下に向かって順次適用していくことを想定して，上部チャネルのチャネル密度ばかり大きくなって，結果として全体のチャネル密度の減少につながらないようなことが起きないようにするために導入している．制約 (2) の r はセルの移動可能範囲と呼び，0 以上の整数である．制約 (3) は全体のレイアウト面積が増加することを防ぐために導入している．また，制約 (4) はタイミング制約である．ここで，与えられたセル配置と初期セルパターンによって決定される各ネットのスパン長は与えられたタイミング制約を満足しているものと仮定する．

3. 提案手法

3.1 定　　義

問題 TDPA に対する提案手法を説明するための準備として，まず用語を定義する．中央セル行のセルを左から順に c_1，c_2，c_3，...，とする．まず，セル c_i の上部インターバル密度 $id_U(i)$ を $id_U(i) = \max_{lpos(c_i) \leq l \leq rpos(c_i)} d_U(l)$ と定義する．ここで $d_U(l)$

表 1 配置タイプの総数
Table 1 The total number of placement types.

移動可能範囲 r	0	1	2	3	4	5	6	7
配置タイプの総数	1	3	12	50	210	882	3696	15444
配置グラフにおけるレベル間の総枝数	1	5	26	134	670	3262	15540	72732

は上部チャネルの $x = l$ における列密度である．セル c_i の下部インターバル密度 $id_L(i)$ も同様に定義する．このとき，セル c_i のインターバル密度 $id(i)$ を $id(i) = id_U(i) + id_L(i)$ と定義する．

次に部分チャネル密度を定義する．まず，セル c_i の上部サブチャネル密度 $sd_U(i)$ を $sd_U(i) = \max_{1 \le j \le i} id_U(j)$ と定義する．セル c_i の下部サブチャネル密度 $sd_L(i)$ も同様に定義する．このとき，セル c_i のサブチャネル密度 $sd(i)$ を $sd(i) = sd_U(i) + sd_L(i)$ と定義する．

次にインターバル配線長を定義する．前述のように $N_U(c_i)$ を上部チャネルにおいてセル c_i の端子に接続されるネットの集合とする．また，$PN_U(c_i)$ を上部チャネルにおいてセル c_i を横切るネットの集合とする．すなわち，$PN_U(c_i) = \{n | n \in N_U$，かつ，$l(n) \le lpos(c_i) < rpos(c_i) \le r(n)\}$ である．下部チャネルに対しても $N_L(c_i)$，$PN_L(c_i)$ を同様に定義する．このとき，セル c_i の上部インターバル配線長 $iw_U(i)$ を

$$iw_U(i) = \sum_{n \in N_U(c_i) \cup PN_U(c_i)} (\min\{r(n), rpos(c_i)\} - \max\{l(n), lpos(c_i)\})$$

と定義する．下部インターバル配線長 $iw_L(i)$ も同様に定義する．このとき，セル c_i のインターバル配線長 $iw(i)$ を $iw(i) = iw_U(i) + iw_L(i)$ と定義する．

最後にサブチャネル配線長を以下のように定義する．セル c_i の上部サブチャネル配線長 $sw_U(i)$ を $sw_U(i) = \sum_{1 \le j \le i} iw_U(j)$ と定義する．下部サブチャネル配線長 $sw_L(i)$ も同様に定義する．このとき，セル c_i のサブチャネル配線長 $sw(i)$ を $sw(i) = sw_U(i) + sw_L(i)$ と定義する．

3.2 セル配置改良の制約

問題 TDPA の定義より，初期配置で与えられた中央セル行のセルの並びは配置改良後はそれぞれ元の位置から r 個まで移動できる．たとえば $r = 2$ の場合，初期配置において左端から i 番目に位置していたセルは配置改良後は $i-2$，$i-1$，i，$i+1$，$i+2$ 番目のいずれかに位置することが可能である．このことを考慮すると，各セルがどのように配置改良されるかをいくつかの場合に分類することができる．以下ではそのように分類された配置改良の種類を配置タイプと呼ぶ．

配置タイプは以下のように形式的に定義される．今，初期配置において中央セル行のセルの並びは左から順に c_1，c_2，c_3，...，だったとし，各セルの移動可能範囲を $r \ge 0$ とする．配置改良後の中央セル行において，左から i 番目に位置するセルを c_{i+k} とする．ただし，$-r \le k \le r$ である．配置改良後の中央セル行においてセル c_{i+k} の左に位置しているセルの集合を $C_L(i)$ で表す．このとき，以下の添え字の集合を定義する．

$$I_L(i) = \{j | -r \le j \le r, c_{i+j} \in C_L(i)\}$$

このとき，セル c_{i+k} の配置タイプを 2 項組 $\langle k, I_L(i) \rangle$ で表す．定義より明らかに，配置タイプの総数はセルの移動可能範囲 r に依存し，r を定数と見なす場合は配置タイプの総数も定数となる．r が与えられたとき，$(2r+1)$ 個のセルの並びを考え，セルの並びのすべての順列の中で各セルの移動可能範囲の制約を満たすセルの並びを選択することにより，配置タイプの総数は簡単に求めることができる．**表 1** に r が 0 から 7 までの場合の配置タイプの総数を示す．表中の 3 行目の数値については後述する．

配置改良後の中央セル行のセルの並びにおいて隣り合う 2 つのセルの配置タイプを考えると，配置タイプの定義から 2 つのセルの配置タイプは互いに関係があり，独立ではないことが分かる．たとえば，$r = 2$ の場合，12 種類の配置タイプがあるが，配置改良後の中央セル行の左から h 番目のセルがある配置タイプを持つとき，$(h+1)$ 番目のセルは 12 通りの配置タイプの中で許されるのはたかだか 3 通りの配置タイプのみである．**表 2** に $r = 2$ の場合の配置タイプの定義と，右に隣り合うセルの可能な配置タイプを示す，なお，以下では便宜上，配置タイプに番号を付加し，タイプ 1～タイプ 12 のように呼んでいる．

配置タイプの定義より，配置改良後の中央セル行の可能なすべてのセルの並びを非巡回有向グラフで表現することができる．このような有向グラフを配置グラフと呼ぶ．配置グラフ $PG = (PV, PE)$ は以下のように定義される．今，初期配置の中央セル行は n 個のセルがあり，左から順に c_1，c_2，c_3，...，c_n と並んでいたとする．このとき配置グラフの節点集合 PV は $PV = \{\langle i, t, h \rangle | 1 \le i \le n$，$t$ は配置タイプ，$i-r \le h \le i+r$，ただし，セル c_i は配置タイプ t で中

表 2 配置タイプの例 ($r=2$)
Table 2 An example of placement types ($r=2$).

タイプ番号	タイプの定義	右隣のセルのタイプ番号
1	$\langle -2,\{-1,0\}\rangle$	6, 9, 11
2	$\langle -2,\{0,1\}\rangle$	1
3	$\langle -2,\{-1,1\}\rangle$	4, 8, 10
4	$\langle -1,\{-2,0\}\rangle$	6, 9, 11
5	$\langle -1,\{-2,1\}\rangle$	4, 8, 10
6	$\langle 0,\{-2,-1\}\rangle$	6, 9, 11
7	$\langle 0,\{-2,1\}\rangle$	1
8	$\langle 1,\{-2,0\}\rangle$	1
9	$\langle 1,\{-2,-1\}\rangle$	4, 8, 10
10	$\langle 2,\{-2,0\}\rangle$	3
11	$\langle 2,\{-2,-1\}\rangle$	5, 7, 12
12	$\langle 2,\{-2,1\}\rangle$	2

央セル行の h 番目に配置可能 } と定義される．ここで，「セル c_i は配置タイプ t で中央セル行の h 番目に配置可能」という条件がついているのは，たとえば，あるセルを 1 番目のセルとして配置する場合，そのセルの左にさらに別のセルが配置されることを仮定している配置タイプは選択できないので，このような組合せを排除するために導入されている．h 番目に配置されたセルに対応する節点をレベル h の節点と呼ぶ．次に配置グラフの有向枝集合 PE は $PE=\{(u,v)|u,v\in PV$, かつ $u=\langle i,t,h\rangle$, $v=\langle i',t',h'\rangle$ とするとき，セル c_i を配置タイプ t で h 番目に，セル $c_{i'}$ を配置タイプ t' で $h'=h+1$ 番目に配置可能 } と定義する．有向枝の定義より配置グラフは非巡回有向グラフとなる．配置グラフの節点で，入力枝次数が 0 の節点を最左端節点，出力枝次数が 0 の節点を最右端節点と呼ぶ．レベル h とレベル $h+1$ の節点集合の間の枝の総数を表 1 に示す．配置グラフの例として $n=6$, $r=2$ の場合の配置グラフを図 3 に示す．

3.3 タイミング制約のない場合のアルゴリズム

配置グラフの定義より，配置改良後の中央セル行の可能なセルの並びは配置グラフにおいて最左端節点から最右端節点までの経路に対応する．この事実を利用して，配置改良と各セルの端子割当てを同時に行うアルゴリズムを以下のように構成できる．ここではまずタイミング制約がない場合についてアルゴリズムを説明し，その後，次節でタイミング制約を考慮した手法を述べる．

タイミング制約がない場合（すなわち，すべてのネットについて最大許容スパン長が配置幅 L の場合），総チャネル密度，上部チャネル密度，総予測配線長の 3 項組を辞書式順序で最小化する配置改良と各セルの端子割当てを求めることになる．配置改良については配置グラフのすべての経路を考慮することで最適解を見

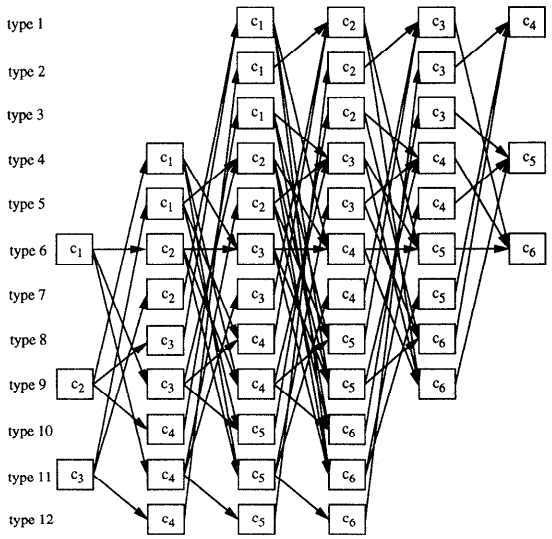

図 3 配置グラフの例 ($r=2$)
Fig. 3 An example of the placement graph ($r=2$).

つけることができる．次にチャネル密度と配線長についてであるが，これについては次に示す 4 つの重要な事実があり，これに基づいて提案手法は構成されている．

事実 1 中央セル行のあるセル c のインターバル密度とインターバル配線長は c の左に配置されたセルの集合と右に配置されたセルの集合，およびセルの端子割当て（セルパターン）のみから計算できる．

すなわち，セルのインターバル密度とインターバル配線長の計算にはそのセルの左と右にどのようなセルが配置されているかとセルパターンだけが必要であり，セルの左や右に他のセルがどのような順序で並んでいるかとか他のセルがどのセルパターンで配線されているかという情報は必要ない．この事実 1 の正当性はインターバル密度とインターバル配線長の定義より簡単に示すことができる．事実 1 より，配置グラフの各節点に対して，各節点に許されるセルパターンごとにインターバル密度とインターバル配線長を計算することができる．

事実 2 チャネル配線長を無視した場合の問題 TDPA の解の上部チャネル密度と下部チャネル密度をそれぞれ D_U, D_L とする．このとき，D_U, D_L を上部チャネル密度と下部チャネル密度の上限と見なして，この制約の下にチャネル配線長を最小とする問題 TDPA の解を求めれば，その解は元の問題 TDPA の解となる．

この事実 2 は問題 TDPA の目的関数が総チャネル密度，上部チャネル密度，チャネル配線長の 3 項組であり，辞書式順序により総チャネル密度と上部チャネル密

度が配線長より優先されて評価されることから自明である．事実2よりアルゴリズムにおいては，まず総チャネル密度と上部チャネル密度の2項組 $\langle D_{total}, D_U \rangle$ の最小値を求めてから，その値を制約として総予測配線長を最小とする端子割当てとセル配置改良を求めればよいことが分かる．

事実3 問題TDPAの解は配置グラフにおけるある最左端節点 vl_1 からある最右端節点 vr_n までの経路 P における節点とその節点に対して選択されたセルパターンの2項組の系列に対応する．すなわち，P 上の節点を vl_1 から順に $vl_1 = v'_1, v'_2, v'_3, \ldots, v'_n = vr_n$ とし，それぞれの節点に対して選択されたセルパターンの系列を $p'_1, p'_2, p'_3, \ldots, p'_n$ とすれば，問題TDPAの解は $\langle v'_1, p'_1 \rangle, \langle v'_2, p'_2 \rangle, \ldots, \langle v'_n, p'_n \rangle$ で表される．

配置グラフ上の最左端節点から最右端節点までの経路と問題TDPAにおける可能なセル配置の間に1対1対応関係があることに注意すれば事実3は自明である．

事実4 問題TDPAの解の上部チャネル密度と下部チャネル密度をそれぞれ D_U，D_L とする．このとき，配置グラフのすべての節点 v に対し，配置グラフの最左端節点から節点 v までの任意の経路とそれに対するセルパターンの割当ての中で，上部チャネル密度と下部チャネル密度がそれぞれ D_U，D_L 以下で，サブチャネル配線長が最小となる経路と経路上の節点に対するセルパターンの割当てを節点 v に対する部分解と呼び，部分解に対応するサブチャネル配線長を節点 v の最適サブチャネル配線長と呼ぶ．このとき，節点 v に対する最適サブチャネル配線長は以下のように表される．

$$sw_v = \min_{(u,v)\in PE} sw_u + \min_{p'_j \in P'_v} iw'_j$$

ここで，PE は配線グラフの有向枝集合，sw_u は節点 u の最適サブチャネル配線長，P'_v はチャネル密度の制約を満足する v のセルパターン集合，iw'_j は v のセルパターン p'_j に対するインターバル配線長を表す．

事実4もインターバル配線長とサブチャネル配線長の定義より簡単に示すことができる．事実4より，問題TDPAにおける最適解の計算は，非巡回有向グラフにおける最小経路の探索と同じ構造を持つことが分かる．

事実1〜4に基づいてタイミング制約の考慮をしない問題TDPAに対するアルゴリズムを以下に示す．まず，アルゴリズムの基本となる配置グラフのデータ構造をPASCAL言語のレコード記述で図4に示す．次に，このデータ構造に基づくアルゴリズムの概略を以下に示す．

```
type
 DensityLength = array[1..MaxPatterns, 1..2] of
                 integer; { 密度/配線長配列 }
 VertexData = record { 配置グラフの節点データ }
   CellID: integer;
 { 節点に対応するセルのネットリストにおける識別番号 }
   LPos:  integer; { セルの左端の x 座標 }
   RPos:  integer; { セルの右端の x 座標 }
   NumPatterns:  integer; { 有効なセルパターン数 }
   PDensity:  DensityLength;
          { 各パターンに対するインターバル密度 }
   PLength:  DensityLength;
          { 各パターンに対するインターバル配線長 }
   Pattern:  integer; { 選択したセルパターン }
   Density:  integer; { サブチャネル密度 }
   Length:  integer; { サブチャネル配線長 }
   Precede:  integer { バックトラック用のポインタ }
 end;
```

図4 配置グラフの節点のデータ構造
Fig. 4 Data structure of a vertex of the placement graph.

ステップ1 配置グラフを構成する．

ステップ2 各節点の各セルパターンについて，上部インターバル密度および下部インターバル密度を求める．まず，ステップ2の準備として初期配置の中央セル行の各セルについて，上部チャネルと下部チャネルにおいて各セルのインターバルを完全に通過するネット数を求める．ここでネット n があるセル c のインターバルを完全に通過するとは，$l(n) < lpos(c) < rpos(c) < r(n)$ が成り立つことをいう．

ステップ2.0 まず，各ネットの $l(n)$ と $r(n)$ を求める．次に中央セル行を左端から右に向かって走査することにより，各セルのインターバルを完全に通過するネット数を求めることができる．

次に配置グラフの各節点について，各節点についてステップ2.1〜2.4の手順を繰り返してインターバル密度を計算する．

ステップ2.1 インターバル密度を求めようとしている節点のレベルを i とする．ステップ2.0の結果を利用して，初期配置の中央セル行の c_{i-r} から c_{i+r} までのインターバルを完全に通過する上部チャネルと下部チャネルのネット数を求める．求めたネット数を du_1，dl_1 とする．

ステップ2.2 節点の配置タイプに対応して，中央セル行の c_{i-r} から c_{i+r} のセルの並びを入れ替えた $(2r+1)$ 個のセルから構成されるセル行を作成する．

ステップ 2.3 ステップ 2.2 で作成したセル行に対し，インターバル密度を求めようとしている節点のインターバルを完全に通過する上部チャネルと下部チャネルのネット数を求める．求めたネット数を du_2，dl_2 とする．

ステップ 2.4 インターバル密度を求めようとしている節点の各セルパターン P について，そのセルの端子が $l(n)$ もしくは $r(n)$ となるようなネットに注目して，それらのネットのみから定義される上部および下部インターバル密度を求める．求めた密度を du_3，dl_3 とすれば，セルパターン P に対応する上部インターバル密度，下部インターバル密度はそれぞれ $du_1+du_2+du_3$，$dl_1+dl_2+dl_3$ となる．このステップ 2.4 の操作をすべてのセルパターンについて繰り返す．

ステップ 3 総チャネル密度と上部チャネル密度の 2 項組 $\langle D_{total}, D_U\rangle$ を最小とする最左端節点から最右端節点までの経路を求める．求めた経路の上部チャネル密度を D_U，下部チャネル密度を D_L とする．

ステップ 4 各節点に対し，上部チャネル密度，下部チャネル密度がそれぞれ D_U，D_L 以下となるセルパターンに対してインターバル配線長を計算し，最小インターバル配線長となるセルパターンを求める．

ステップ 5 配置グラフの各節点について，最左端節点からトポロジカルソートの順序で，上部チャネル密度，下部チャネル密度がそれぞれ D_U，D_L 以下となる条件の下に各節点の最適サブチャネル配線長を求める．

ステップ 6 最右端節点の中で上部チャネル密度，下部チャネル密度がそれぞれ D_U，D_L 以下となる条件の下にサブチャネル配線長が最小となる節点を見つけ，その節点より最左端節点までバックトラックすることにより各セルの配置位置とセルパターンを求める．

上記の提案アルゴリズムの正当性と計算量について以下の定理が成り立つ．

定理 1 提案アルゴリズムはタイミング制約の付かない問題 TDPA を最適に解く．また，中央セル行のセル数を n，セルの移動可能範囲を r とし，各セルの端子数の上限を t，各セルのセルパターンの上限を p とする．r，t，p を定数とすれば，アルゴリズムの時間計算量と空間計算量はともに $O(n)$ となる．

証明 アルゴリズムの正当性は事実 1～4 より簡単に示すことができるが，ここでは紙面の都合上，詳細は省略する．次に時間計算量について考察する．ステップ 1 では配置グラフ $PG=(PV,PE)$ を構成している．配置グラフの節点数 $|PV|$ は配置タイプの総数を T とすれば，$|PV|<n\times T$ である．ここで簡単な考察より $T<(2r+1)\begin{pmatrix}2r-1\\ r-1\end{pmatrix}$ となる．また，レベル h の節点からレベル $h+1$ の節点への枝の総数を A とすれば，配置グラフの枝数 $|PE|$ は，$|PE|<n\times A$ である．ここで，$A<(r+1)(2r+1)\begin{pmatrix}2r-1\\ r-1\end{pmatrix}$ を示すことができる．したがって，r を定数とすれば $|PV|=O(n)$，$|PE|=O(n)$ となるので，ステップ 1 の時間計算量も $O(n)$ となる．

次にステップ 2 について考える．ネットの総数を $|N|$ とするとき，仮定より $|N|\leq nt$ となる．バケツソートを用いればステップ 2.0 は $O(|N|+nt\log t)$ の時間計算量で実行できる．各節点に対するステップ 2.1～2.4 の計算時間の総和は $|PV|\times O(pt\log t)$ となる．したがって，仮定よりステップ 2 の時間計算量は $O(n)$ となる．ステップ 3 については，非巡回有向グラフ $G=(V,E)$ の最短経路は一般に $O(|V|+|E|)$ の時間計算量で求めることができるので[3]，仮定よりステップ 3 の時間計算量は $O(n)$ となる．同じ理由によりステップ 4，5 の時間計算量は $O(n)$ である．ステップ 6 は明らかに $O(n)$ の時間計算量である．したがって，定理が成り立つ．アルゴリズムの空間計算量は明らかに $O(|PV|p+|PE|)$ となるので，仮定より $O(n)$ となる．（証明終）

3.4 タイミング制約を考慮したアルゴリズム

前節で提案したアルゴリズムをタイミング制約をともなう問題 TDPA のアルゴリズムに拡張する．しかし，前節で提案したアルゴリズム上で直接タイミング制約を考慮しながら端子割当てとセル配置改良を行うことは困難である．このため，ネットに重みを導入し，重み付きネットに対する配線長最小化問題としてタイミング制約を扱うことにする．

問題 TDPA の定義から，各ネット n に対し，タイミング制約としてネットの最大許容スパン長 $maxspan(n)$ が与えられている．ネット n のスパン長は端子割当てと配置改良により初期配置におけるスパン長から変化するが，セルの移動可能範囲の制約があるので，端子割当てと配置改良の結果においてスパン長が長くなっても，その上限はあらかじめ計算できる．この上限を $longest\_span(n)$ で表す．このとき，$maxspan(n)<$

$longest\_span(n)$ となるようなネットをクリティカルネットと呼び，その他のネットをノーマルネットと呼ぶ．定義よりクリティカルネットはタイミング制約を考えずに前節の提案アルゴリズムで配置改良や端子割当てを行ってしまうとタイミング制約違反を生じる可能性のあるネットであり，ノーマルネットはそのような可能性のないネットである．タイミング制約を扱うために以下ではネットに重みを導入する．ネット n の重みを $w(n)$ で表す．クリティカルネットに対しては $w(n) = u \geq 1.0$ の重みを与え，u の値はタイミング制約の条件が厳しいほど大きな値とする．また，ノーマルネットに対しては $w(n) = u' = 1.0$ の重みを与える．このような重みを与えたうえで，前節の提案アルゴリズムの配線長の計算を重み付き配線長に変更して，セル配置改良と端子割当てを求める．すなわち，前節のアルゴリズムではチャネル Q の予測チャネル配線長を

$$W_Q = \sum_{n_i \in N_Q} span(n_i)$$

で定義していたのに対し，この節では

$$W_Q = \sum_{n_i \in N_Q} span(n_i) \times w(n_i)$$

として求める．

本論文ではクリティカルネット n に対する重み $w(n)$ の与え方として，ネット n のスパン長 $span(n)$ が $\alpha \times maxspan(n)$ 未満の場合は $w(n) = 1.0$，そうでなければ

$$w(n) = \frac{\beta - 1}{(1-\alpha) maxspan(n)} span(n) + \frac{1 - \alpha\beta}{1-\alpha},$$

とする．ただし，α と β は $0.0 < \alpha < 1.0$，$\beta > 1.0$ を満足する定数である．この式において，$span(n) = maxspan(n)$ のとき，$w(n) = \beta$ となる．

タイミング制約を考慮した場合の提案手法の概略を以下に示す．まず，重み付き配線長の総和が最小となるセル配置改良と端子割当てを求める．求めた配置と端子割当てに対してタイミング制約を違反していないかどうかの検証を行い，違反しているネットがあれば，そのネットに対する重みを増加して再度アルゴリズムを実行する．この操作をタイミング制約違反のない解が求まるか，あるいは規定の回数繰り返したらアルゴリズムを終了する．なお，このアルゴリズムはタイミング制約がない場合は前節のアルゴリズムと同じ動作となる．

ステップ 1〜3 前節のアルゴリズムのステップ 1〜3 と同じ．

ステップ 4 各ネットをクリティカルネットとノーマルネットに分類する．ノーマルネットの重みは 1 とし，クリティカルネットの重みは上述の式より計算される値とする．*LoopCount* を 1 とする．

ステップ 5〜7 前節のステップ 4〜6 と同じ．ただし，配線長の計算はすべて重み付きで行う．

ステップ 8 すべてのクリティカルネット n に対してタイミング制約を満足しているかどうかを調べる．満足していないネットについてはネットの重み $w(n)$ に γ を加えたものを新たにネット n の重み $w(n)$ とする．

ステップ 9 すべてのクリティカルネットについて，タイミング制約を満足している場合は直前に実行したステップ 7 の結果をアルゴリズムの結果として出力してアルゴリズムを終了する．そうでなければ *LoopCount* に 1 を足す．*LoopCount* があらかじめ定められた定数 *MaxLoopCount* に達していれば初期配置をアルゴリズムの結果として出力して終了する．そうでなければステップ 5 に戻る．

このアルゴリズムは問題 TDPA の許容解（ヒューリスティック解）を出力するが，最適解の出力は保証されないので，問題 TDPA のヒューリスティックアルゴリズムと見なすことができる．また，時間計算量と空間計算量は前節のアルゴリズムと同じである．

3.5 一般のスタンダードセルレイアウトへの適用

前節で提案したアルゴリズムを一般のスタンダードセルレイアウトに適用する場合は，上の行から順に中央セル行と見なし，その上下に位置するセル行もしくは外部端子列を上部セル行，下部セル行と見なして順次アルゴリズムを適用することによりヒューリスティック解を得ることができる．また，最下位セル行までアルゴリズムの適用を終了したら，再び最上位セル行からアルゴリズムの適用を繰り返すことにより，さらに配置改良を行うこともできる．

4. 実験結果

提案アルゴリズムの有効性を示すため，ワークステーション上に 3.4 節で示したアルゴリズムをプログラムとして実現し，ベンチマークデータを用いた計算機実験により評価した．プログラムは PASCAL で記述し，UltraCOMPstation 170 上で実行した．計算機実験では ISCAS ベンチマークデータの一部をテストデータとして用いた．テストデータは論理合成システム SIS により論理合成とテクノロジマッピングを行い，スタンダードセルレイアウトシステム TimberWolfSC4.2c によりセル配置と概略配線およびフィードスルーセル挿入を行った．このようにして求めたセル配置と概略

表 3 テストデータ
Table 3 Test data.

Data	$\#cells$	$\#nets$	$\#rows$	D_{init}	$W_{init}(\lambda)$
C1	622	838	8	108	159739
C2	837	1192	9	123	209418
C3	897	1331	10	173	361556
C5	1802	2484	13	295	802788
C7	4036	4954	18	335	1042243

配線のデータを提案アルゴリズムの入力とした．**表 3** にテストデータの概要を示す．表中の Data，$\#cells$，$\#nets$，$\#rows$ はそれぞれデータ名，セル数，ネット数，セル行数である．チャネル数は $\#rows+1$ となる．また，ネット数は各チャネルで配線されるネット数の総和を示す．D_{init} と W_{init}（単位は λ）は与えられた初期セル配置における各チャネルのチャネル密度とチャネル配線長の総和を示す．各セルのセルパターンはライブラリで与えられているセルレイアウトとそれをフリッピングしたものの 2 種類のみとした．上部チャネル密度の上限 $Dmax_U$ は各セル行にアルゴリズムを適用して配置改良を行う前の上部チャネル密度とした．

アルゴリズム中の各定数については予備実験により適切と考えられる値をあらかじめ求めて，それらの値を実験では使用した．まず，タイミング制約を扱うためのネットの重み付けに用いる定数 α，β，γ はそれぞれ $\alpha=0.9$，$\beta=2.0$，$\gamma=0.3$ とした．また，タイミング制約を満たさなかった場合のアルゴリズムの繰返し回数（ネットの重みの更新回数）の上限である MaxLoopCount は 20 とした．さらに，レイアウト全体に対しては 3.5 節で示した方法で提案アルゴリズムを適用しているが，最上位セル行から最下位セル行までの提案アルゴリズムの適用を 1 ループとし，20 ループまで繰り返した．ただし，あるループにおいていずれのセル行においても配置改良ができなかった場合はそのループでアルゴリズムを終了させた．

テストデータに対する正確なタイミング制約の生成が困難だったので，実験で用いたタイミング制約については以下の手法によりランダムに生成した．まず，レイアウト中の各セルの幅の平均を求める．これを $Width_{ave}$ とする．次に与えられた初期配置における各ネット n のスパン長を $init\_span(n)$ とし，タイミング制約である最大許容スパン長を $maxspan(n)$ とする．このとき，$maxspan(n)$ を以下の式により生成した．

$$maxspan(n) = init\_span(n) + slack(n)$$

ただし，$slack(n)$ は $min\_slack \times Width_{ave} \leq slack(n) \leq max\_slack \times Width_{ave}$ を満たすランダムに生成された整数である．また，min_slack，max_slack は $0 < min\_slack \leq max\_slack$ を満たす自然数である．min_slack，max_slack が小さいほど，厳しいタイミング制約が生成される．以下では表 6 の実験結果を除いて，つねに $min\_slack = 2$，$max\_slack = 10$ でタイミング制約を発生させた．また，表 6 の実験結果を除いて，すべての実験で同じデータについては同じタイミング制約を用いた．

セルの移動可能範囲 r を 3 としたときの提案アルゴリズムの実験結果を**表 4** に示す．D_{final}，W_{final} は出力されたセル配置におけるチャネル密度とチャネル配線長の総和を示す．カッコ内の数値は初期配置におけるそれらの値を 1 としたときの比を示す．$\#loop$ はセル配置全体に対する提案アルゴリズムのループ回数を示す．前述のようにループ回数の上限は 20 とした．CPU はアルゴリズムの計算時間（単位は秒）である．実験結果より提案アルゴリズムはチャネル密度を最大 13.0%，最小 6.5%，平均 9.0%，またチャネル配線長を最大 8.8%，最小 3.9%，平均 6.6% 改善していることが分かり，アルゴリズムの有効性が確認できた．

次にセルの移動可能範囲 r と得られる解の質および計算時間の関係について調べた．データ C5 と C7 に対して r を 0 から 4 まで変化させたときの実験結果を**表 5** に示す．C7 に対しては $r=4$ の場合，CPU 時間が 5 万秒を超えても解を求めることができなかったので計算を途中で打ち切った．実験結果より r が大きくなるほど一般に解は良くなるが，計算時間も増大することが分かる．$r=1$ の場合でも短い計算時間で配置が改良されていることが分かる．一般に，従来のセル配置手法においても隣接するセルどうしを対交換することである程度の配置改良を実現しているが，配置改良結果は対交換の場所と順序に依存する．また，セルパターンの選択は対交換とは独立に行われる．これに対し，提案アルゴリズムにおいては最適な対交換とセルパターン選択を保証しているので，従来手法と少なくとも同等かより良い配置改良を実現している．ここで，C5 については $r=4$ の場合，$r=3$ より解が悪くなっているが，これはアルゴリズムがヒューリスティックなためにたまたま起こった現象であると考えられる．提案アルゴリズムはタイミング制約がなく，かつ 2.4 節で示した 3 セル行のレイアウトモデルに対しては最適解の出力を保証しており，r が大きくなると少なくとも解が悪くなることはないことも保証されるが，一般のタイミング制約を持つレイアウトモデルに対しては最適解を出力する保証がない．この実験で

表 4 タイミング制約を考慮した実験結果 ($r = 3$)
Table 4 Experimental results considering timing constraints ($r = 3$).

Data	D_{final}	$W_{final}(\lambda)$	$\#loop$	CPU (sec)
C1	94 (0.870)	145647 (0.912)	11	202
C2	115 (0.935)	201301 (0.961)	9	250
C3	154 (0.890)	333188 (0.922)	9	390
C5	262 (0.888)	741696 (0.924)	20	2099
C7	306 (0.913)	992897 (0.953)	20	5400

表 5 セルの移動可能範囲に関する実験結果 (C5, C7)
Table 5 Experimental results on the movable range of cells (C5, C7).

Case	D_{final}	$W_{final}(\lambda)$	$\#loop$	CPU (sec)
C5, $r = 0$	295 (1.000)	796972 (0.993)	3	5
C5, $r = 1$	278 (0.942)	767612 (0.956)	20	137
C5, $r = 2$	271 (0.919)	749172 (0.933)	20	557
C5, $r = 3$	262 (0.888)	741696 (0.924)	20	2138
C5, $r = 4$	267 (0.905)	750776 (0.935)	14	6037
C7, $r = 0$	335 (1.000)	1037151 (0.995)	3	16
C7, $r = 1$	320 (0.955)	1011315 (0.970)	20	344
C7, $r = 2$	311 (0.928)	993881 (0.954)	20	1359
C7, $r = 3$	306 (0.913)	992897 (0.953)	20	5419
C7, $r = 4$	—	—	—	> 50000

表 6 タイミング制約に関する実験結果 (C7)
Table 6 Experimental results on timing constraints (C7).

Case	D_{final}	$W_{final}(\lambda)$	$\#loop$	$\#vio$	CPU (sec)
Tight	306 (0.913)	992897 (0.953)	20	0	5419
Loose	296 (0.884)	970341 (0.931)	20	0	6776
No	296 (0.884)	966815 (0.928)	20	182	7370

は $r = 4$ の場合，$r = 3$ の場合より悪い局所解で解が改善できなくなったものと考えられる．

この実験結果より，各セルごとに適切なセルパターンの選択を行う（$r = 0$ の場合に相当）だけではほとんど配置改良を行うことができず，セルパターンの選択と配置改良を同時に行う本アルゴリズムが有効であることが分かる．また，実用上は $r = 2$ もしくは $r = 3$ で提案アルゴリズムを実行することが適当だと考えられる．

最後に，タイミング制約の厳しさと得られる解の関係について考察するための実験を行った．**表 6** にデータ C7 に対する実験結果を示す．この表において，Tight となっているのはこれまでの実験結果と同じ場合であり，前述のタイミング制約の生成方法において，$min\_slack = 2$，$max\_slack = 10$ とした場合の結果である．Loose は $min\_slack = 5$，$max\_slack = 25$ の場合であり，Tight の場合と比較して平均して 2.5 倍，タイミング制約が緩い．No はタイミング制約を与えなかった場合の実験結果である．表 6 の結果より，タイミング制約が厳しい場合は配置改良の度合いが下がることが分かる．表中の $\#vio$ は出力されたセル配置において満足されなかったタイミング制約の総数である．No の場合，タイミング制約を考慮していないので，結果として $min\_slack = 2$，$max\_slack = 10$ で生成したタイミング制約に対して 182 個の違反があった．これより提案アルゴリズムはタイミング制約を満たしながら効率良く配置改良を実現できることが分かる．実験結果においてタイミング制約が緩くなるほど計算時間が増加しているのは，一般のレイアウトモデルに対する提案アルゴリズムの適用においては，配置改良の見込みのないセル行，すなわち前回のループにおいてそのセル行と上下のセル行がいずれも配置改良されていなかった場合はそのセル行についてはアルゴリズムの適用を行わずに次のセル行の処理に移ることにしているので，タイミング制約が厳しくなるほど実際に配置改良されるセル行の総数が減少するためだと考えられる．

5. おわりに

本論文ではチャネル配線領域を持つスタンダードセルレイアウトに対するセル改良と端子割当てを同時に行う手法を提案した．本手法は各チャネルで配線され

るネットの配線長（スパン長）の上限という形でタイミング制約を考慮することができる．類似の問題に対する従来の結果の大半が解の最適性の保証がないのに対し，本論文の提案アルゴリズムは対象とするレイアウトモデルはセル行が3行でセルの移動範囲に制約があるという条件があるが，タイミング制約が与えられない場合はセル数に比例する計算時間での最適解の生成を保証する，という大きな特長を持つ．また，提案アルゴリズムはヒューリステイックアルゴリズムとして一般のレイアウトモデルへの適用も簡単である．計算機実験の結果，満足すべき結果が得られた．

今後の課題としては，実際のデータとタイミング制約に対して提案アルゴリズムを適用し，詳細配線まで行った場合の評価を行うことがある．また，多項式時間での最適解の生成という提案アルゴリズムの特長を保ったまま，レイアウトモデルをより広いものにすることがある．

参考文献

1) Cai, Y. and Wong, D.F.: Optimal channel pin assignment, *IEEE Trans. CAD*, Vol.10, No.11, pp.1413–1424 (1991).
2) Chang, K.-E.: Efficient algorithms for wiring channels with movable terminals, *IEEE Trans. CAD*, Vol.12, No.7, pp.1059–1063 (1993).
3) Cormen, T.H., Leiserson, C.E. and Rivest, R.L.: *Introduction to Algorithms*, MIT Press, Cambridge (1990).
4) Gopal, I.S., Coppersmith, D. and Wong, C.K.: Optimal wiring of movable terminals, *IEEE Trans. Comput.*, Vol.C-32, No.9, pp.845–858 (1983).
5) Her, T.W. and Wong, D.F.: Optimal module implementation and its application to transistor placement, *Proc. ICCAD*, pp.98–101 (1991).
6) Her, T.W., Wang, T.-C. and Wong, D.F.: Performance-driven channel pin assignment algorithms, *IEEE Trans. CAD*, Vol.14, No.7, pp.849–857 (1995).
7) Hou, C.Y. and Chen, C.Y.R.: A hierarchical methodology to improve channel routing by pin permutation, *Proc. ICCAD*, pp.440–443 (1991).
8) Hou, C.Y. and Chen, C.Y.R.: A pin permutation algorithm for improving over-the-cell channel routing, *Proc. Design Automation Conf*, pp.594–599 (1992).
9) Koide, T., Wakabayashi, S. and Yoshida, N.: Optimal channel pin assignment with multiple intervals for building block layout, *Proc. EURO-DAC '92*, pp.348–353 (1992).
10) Sait, S.M. and Youssef, H.: *VLSI Physical Design Automation*, IEEE Press, Piscataway (1995).
11) Tragoudas, S. and Tollis, I.G.: River routing and density minimization for channels with interchangeable terminals, *Integration: the VLSI journal*, Vol.15, pp.151–178 (1993).

(平成10年9月24日受付)
(平成11年2月 8 日採録)

若林 真一（正会員）

昭和31年生．昭和59年広島大学大学院工学研究科システム工学専攻博士課程後期修了．同年日本アイ・ビー・エム（株）入社．東京基礎研究所副主任研究員．昭和63年7月より広島大学工学部助教授．VLSI CAD，組合せ最適化，遺伝的アルゴリズムに関する研究に従事．工学博士．IEEE，ACM，電子情報通信学会各会員．

小出 哲士（正会員）

昭和42年生．平成4年広島大学大学院工学研究科システム工学専攻博士課程前期修了．同年広島大学工学部第二類（電気系）助手．平成11年3月広島大学工学部助教授．平成11年4月より東京大学大規模集積システム設計教育研究センター（VDEC）助教授．VLSIレイアウト設計，組合せ最適化，遺伝的アルゴリズムに関する研究に従事．博士（工学）．IEEE，ACM，電子情報通信学会各会員．