設 計 自 動 化 83－12
1997. 2. 14)

密度推定に基づく全ネット同時配線手法: 端点成長法

佐々木 将央　　高橋 篤司　　梶谷 洋司

東京工業大学 電気・電子工学科
〒 152 東京都目黒区大岡山 2-12-1

接続すべき端子集合 (ネット) が多数指定されているメッシュで区切られている配線領域モデルにおいて，1ネットずつ順に経路決定していくことに対する弊害に対しては，従来から予測とか引き剥し再配線手法を含めて様々な対策が考案されてきた．本研究では各ネットの密度への影響を予測しながら，ネットの端点を両側から少しずつ伸ばすようにして経路を決定して行くことで，すべてのネットを同時に配線する新しいアルゴリズム『端点成長法』を提案する．また，ランダムに生成した実験データに対し実験を行ない，その有効性を確認した．

Net Simultaneous Prediction-Based Router:Terminal-Grow

Masachika SASAKI, Atsushi TAKAHASHI, and Yoji KAJITANI

Department of Electrical and Electronic Engineering,
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152, Japan

In meshed channel routing for many nets, conventional routing methods fix the routing net by net. In this paper, we note the bad effect of the net-wise routing, and propose a new algorithm, "terminal-grow" that routes all nets sirmultaneously by predicting the effect of each net grow to the density and growing each terminal from both sides. Experimental results show that our method constructs lower density global-routing than the conventional method.

1 序論

集積回路の高密度化，高速化，低電力化などにつれて，配線設計問題は今も重要なテーマである．集積回路の配線面積は，概略配線の段階で各領域の配線密度が小さく抑えられた優れた配線経路を得られなければ，詳細配線の段階で配線面積を小さくするのにも限界がある．このような概略配線問題はスタンダードセルやFPGAなど，多くのテクノロジに適用可能なもので，今までにも数多く提案されている．(例えば [1][2][3][4][5] など)．それらの多くはネット毎に経路決定するが，密度の予測を行い密度変化をできる限り抑える順序で経路決定する [1] などの工夫をする．しかし，これらの手法では配線を1本ずつ確定していくので，当然先に経路を決定した配線は，後から経路を決定する配線の密度に与える影響を十分には考慮する事が出来ず，最適解が得られにくい．そのため，引き剥し再配線などで対応することが多かった．したがって従来から多数ネットを同時に配線するアルゴリズムが求められていた．

本研究では，これに応えて (1) 予測密度を使う，(2) 各ネットについて両端子から配線する，という工夫によって，多数ネットを同時に配線するアルゴリズムを提案する．すなわち，提案手法では各ネットの各端子から経路を徐々に決定する．この決定された経路の先端を端点と呼び，端点を移動して，決定さ

れた経路を伸ばす．これをネットの成長と呼ぶ．端点はすべてのネットの両端に存在するが，最も密度に与える影響が少ないと予測される端点を成長させる．そしてすべての経路が決定するまで端点成長を繰り返すことにより全配線経路を得る．この工夫を捉えて本方法を端点成長法と呼ぶ．

本手法と従来手法の比較実験を行った結果，配線結果と計算時間に関し，良好な結果が確認された．

2 準備

スタンダードセルやFPGAなどの配置が与えられた時，配線領域を小領域に分割し，配線がどの小領域を通るかを決定する概略配線を行い，その後，各小領域内での詳細な配線を考える．

概略配線における最適化問題は，ある大きさの配線領域と，いくつかの接続すべき端子対が与えられた時に，各小領域での配線密度の最大値が最も小さくなるように配線経路決定する問題に抽象化できる．ただし，配線密度は各小領域を通過するネットの本数である．

本研究では，2端子ネットについてのみ考える．一般的な多端子ネットを対象とする場合は，スタイナー木を生成する既存のアルゴリズム ([6] など) を用いて端子及びスタイナー点間での2端子配線問題に直してから提案アルゴリズムを適用する．

配線領域は，図1のような横X個×縦Y個の小領域からなるとする．各小領域は座標で参照される．各ネットは端子の存在する小領域の組で与えられる．

ネットの集合を N とする．提案するアルゴリズムは，各ネット毎にそれぞれの最短経路を出力する．配線経路は小領域の集合として出力する．各ネットの経路は両端子から互いの距離が近付くように徐々に決定される．決定された経路 (既配線経路) の先端を端点と呼ぶ．端点の移動によって，既配線経路に新たに小領域を加える操作を成長 (grow) と呼ぶ．また，ネット $N^i(1 \leq i \leq |N|)$ について，ネット N^i の既配線経路を w^i，ネット N^i の端点間のマンハッタン距離を δ^i，端点を囲む最小矩形を端点矩形，端点矩形から，端点が属す小領域を除いた小領域の集合を配線候補領域 u^i と呼ぶ．また，ある端点に隣接するそのネットの配線候補領域に属する小領域をその端点の成長可能小領域と呼ぶ．また，2つの小領域 $(p_1, q_1), (p_2, q_2)$ を囲む最小矩形を，領域 $[(p_1, q_1), (p_2, q_2)]$ として参照する．

端点成長法は，配線の各端点を成長させた時の最終的な配線密度への影響を予測する．この予測のためには，配線経路の一部が決定している段階で，最終的な配線密度をある程度推定して知る必要がある．この最終的な配線密度を推定する値として以下の仮配線密度を用いる．

定義 1 ネット毎の仮配線密度 小領域 (p, q) のネット N^i による仮配線密度 $d^i_{p,q}$ は以下の通り．

$$d^i_{p,q} = \begin{cases} 1 & ((p,q) \in \mathbf{w^i}) \\ \frac{(\delta^i - 1)}{|\mathbf{u^i}|} & ((p,q) \in \mathbf{u^i}) \\ 0 & (otherwise) \end{cases}$$

定義 2 仮配線密度 小領域 (p, q) の仮配線密度 $d_{p,q}$ は以下の通り．

$$d_{p,q} = \sum_{1 \leq i \leq |N|} d^i_{p,q}$$

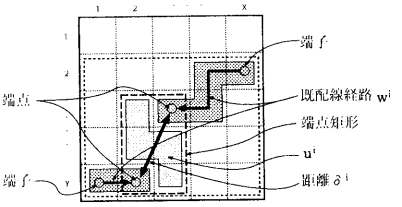

図 1: 概略配線問題

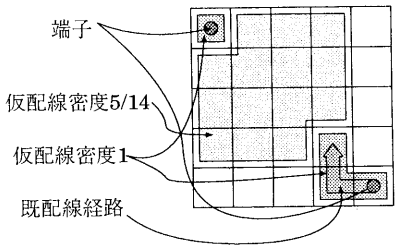

図 2: 仮配線密度

3 端点成長法

3.1 アルゴリズム

端点成長法はネットの端点を1小領域ずつ成長させる．経路の決定してないすべてのネットそれぞれ2つの端点について，各々2つの成長可能小領域がある (両端点のX座標またはY座標のいずれかが一致している時は成長させられる成長可能小領域は1つ)．これらの中から，全体の配線密度に悪い影響を与えないと予測される端点と成長可能小領域を選び，実際に成長させる．この選択は，後で詳しく述べる成長コストにより行なう．

端点成長法は，以下のように実行される．

step1 成長コスト最小端点計算 すべてのネットの両端点の成長可能小領域について成長コストを

求め，その中からコスト最小の端点と成長可能小領域の組を探す．

step2 端点成長 step1で求めた端点について成長を行なう．つまり，選択された成長可能小領域を既配線領域に加える．

step3 step1，step2をすべてのネットの経路決定が完了するまで繰り返す．

3.2 成長コスト

成長コストは各ネットの両端点とその各成長可能小領域について定義される．成長コストは密度平均コスト・進入コスト・縦横比コストの3つに適当な重みづけをした線形和からなる．

$$\begin{aligned}\text{成長コスト} &= \alpha(\text{密度平均コスト}) \\ &+\beta(\text{進入コスト})+\gamma(\text{縦横比コスト}) \qquad (1)\end{aligned}$$

成長コストの小さい成長の優先順位が高い．

3.2.1 密度平均コスト

本アルゴリズムは最短距離の配線をする．このため，ある2点間の配線が最大密度にどれだけ影響を及ぼさずに経路決定出来るかは，その2点を囲む最小矩形内の仮配線密度に関係する．矩形内の密度平均が高い程，そこに配線を伸ばした時の最大密度が高くなる可能性が高い．このため成長前の端点矩形と成長後の端点矩形を比べて，平均が小さいならば，密度の高いと予想される部分を配線候補領域から外すことになるので，このネットが密度の低い小領域を経由する可能性が高い．

定義 3 密度平均コスト　端点矩形内の仮配線密度の平均を，成長可能小領域へ成長した後の端点矩形の平均から引いた値．ただし，仮配線密度は，成長前の値で評価する．

また，密度平均コストの小さい成長は，配線候補領域から外れる部分は成長後の端点矩形内に比べて仮配線密度が高い．そして，外れる領域の成長後の仮配線密度は下がり，成長後の端点矩形内の仮配線密度は上がるので，結局仮配線密度が均一化されることになる．したがって，それ以降の経路決定をより低い密度で行える可能性も高くなる．

3.2.2 進入コスト

端点矩形の仮配線密度平均を考えることで端点のおよその成長すべき方向を考えることができるが，最終的な評価基準が配線領域全体での最大密度である以上，仮配線密度の最大値を考慮する必要がある．

しかしながら，矩形領域内の密度平均は後述するアルゴリズムによって高速に計算できるのに対し，矩形領域内の最大値を計算するのはその矩形の面積に比例する時間がかかってしまう．そこで，端点成長時にその成長可能小領域の仮配線密度を調べ，出来るだけ仮配線密度の低い所から先に伸ばしていくようにする．そのために，端点成長時にその端点が進入する小領域の仮配線密度をコストとして加算する．

定義 4 進入コスト　成長可能小領域の仮配線密度．

3.2.3 縦横比コスト

進入コストは，本来ならば端点矩形全体で仮配線密度最大の点を考えたいところを，これから進入する1点のみに絞って密度を評価する．これは次回以降進入するかもしれない小領域については，仮配線密度の高い点でも，後の経路決定次第では避けられる可能性があるという考え方に基づく．

しかし，この評価を有効に作用させるためには，端点成長後のそのネットの経路決定の多様性が大きくなければならない．よって，端点の成長がある1方向に偏って端点矩形が細長く潰れてしまうことは好ましくない(本アルゴリズムでは最短配線のみ考えていることに注意)．そこで，端点矩形の縦横比と成長の方向のみによって定まるコストを考える．

定義 5 縦横比コスト　端点矩形の幅 h，高さ v の端点を成長させる時の縦横比コストは，

$$\text{縦横比コスト} = \begin{cases} \log_2 \frac{h}{v} & (\text{成長方向が縦の時}) \\ \log_2 \frac{v}{h} & (\text{成長方向が横の時}) \end{cases} \qquad (2)$$

これによって，もともとの端点矩形が正方形ならば縦横比コストは0，さもなければ正方形に近付く成長は負，遠ざかる成長は正となる．

また，式2中のlogの底の2は，縦横比が2倍変わった時にちょうど+1(あるいは−1)のコスト変化があることを示す．

3.2.4 成長コストの例

例えば，図3の状態まで経路が確定していて，図のように端点が成長可能小領域に成長しようとする時，密度平均コストは一点破線の矩形中の仮配線密度平均から点線の矩形中の仮配線密度平均を引いた値，進入コストは端点がこれから進入しようとする座標の仮配線密度の値，縦横比コストは成長前の端点矩形(図中の点線で示される矩形)の縦横比のlogを取った値である．

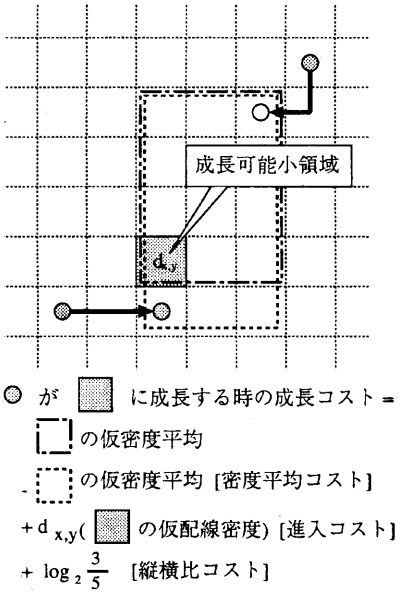

図 3: 成長コスト

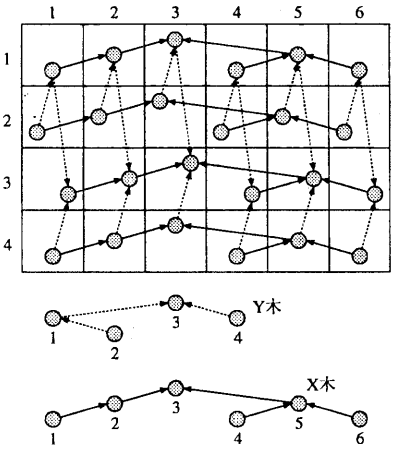

図 4: 2 重木の構成

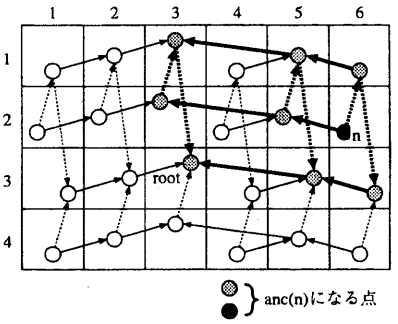

図 5: 点 $n_{6,2}$の先祖

4 密度総和の計算法

4.1 密度平均コスト計算

密度平均コストの計算手順は，その定義に従えば，各小領域の仮配線密度を保存し，求めたい領域内の全ての値を加算し，面積で割ることである．しかし，この計算には配線領域の面積に比例する時間を必要とする．また，仮配線密度の更新にも面積に比例する時間が必要である．

ここでは，特殊なデータ構造を用いることで，ある矩形領域に対する仮配線密度値の増減や，仮配線密度値の総和を，面積を A とするとき $O(\log^2 A)$ で計算するアルゴリズムを提案する．

4.2 データ構造とアルゴリズム

幅 X, 高さ Yの配線領域の各小領域 (i,j) に対して点 $n_{i,j}$を用意する．まず，各行について X座標で中央順 (in-order) になるように点間に辺を加え，同型な平衡木を構成する．これを X 木と呼ぶ．また，各列についても同様に Y 木を構成する．このような構造を 2 重木と呼ぶ．

例えば，6×4 の配線領域であれば，図4のようになる

2 重木において，点 n から，X 木または Y 木の親を辿って到達出来る n 自身を含むすべての点の集合を点 n の先祖 $anc(n)$ とする (図 5)．

また，各点 n は，x と yの多項式

$$f_n(x,y) = axy + bx + cy + d$$

を保存する．また，各点 n について，関数 $S_n(x,y)$ を以下のように定義する．

$$S_n(x,y) = \sum_{k \in anc(n)} f_k(x,y) \qquad (3)$$

ここでは，$S_n(x,y)$ にその点 n の座標 (p,q) を代入することにより，領域 $[(1,1),(p,q)]$ の仮配線密度合計となるように $f_n(x,y)$ を定める．

$$S_{n_{p,q}}(p,q) = \sum_{1 \le i \le p, 1 \le j \le q} d_{i,j} \qquad (4)$$

となる．ただしここで $n_{p,q}$は座標 (p,q) に対応する点を示す．

実際に必要な値は$S_n(x,y)$であるが，領域の密度が更新されるたびにすべての点の$S_n(x,y)$を更新するのは大変な作業である．そこで，2重木上で共通の先祖を持つ点は，その情報をその共通の先祖に$f(x,y)$として集約し，$f(x,y)$一つの更新で，それ以下の子孫の$S(x,y)$を一括して更新できるようにする(式3から，$f_n(x,y)$は点nより子孫すべてに関する情報であることが判る)．

つまり，この式4を満たし続けるように各点の保持する多項式$f_n(x,y)$を更新する．

任意の密度分布は，領域$[(1,1),(p,q)]$に対し密度を多項式回増減することにより得られる．また，$[(p_1,q_1),(p_2,q_2)]$に対する密度の増減は4回の操作で実現できる．また密度が0の場合，$f_n(x,y)=0$とすれば$S_n(x,y)$は実現できる．

そこで，$[(1,1),(p,q)]$へ仮配線密度をD加える時を考える．図6のように領域$[(1,1),(p,q)]$で密度変

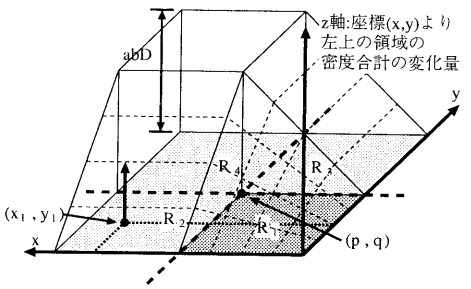

図 6: S_nの意味

化させる時，領域$[(1,1),(p,q)]$をR_1，領域$[(p+1,1),(X,q)]$をR_2，領域$[(1,q+1),(p,Y)]$をR_3，領域$[(p+1,q+1),(X,Y)]$をR_4と定義する．このとき，図6では，点$(x_1,y_1)\in R_2$について，領域$[(1,1),(x_1,y_1)]$の密度の合計の増加分はpy_1Dとなる．つまり，$S_n(x,y)$の変化分を$\Delta S_n(x,y)$とすると，R_2内のすべての点nについて，その座標を(x,y)とする時，領域$[(1,1),(x,y)]$の密度の増加分は$\Delta S_n(x,y)=pDy$である．ほかの領域についても同様のことが言えて，

$$\Delta S_n(x,y)=\begin{cases} Dxy & (n\in R_1)\\ pDy & (n\in R_2)\\ qDx & (n\in R_3)\\ pqD & (n\in R_4)\end{cases} \qquad (5)$$

となる．

また，式3から，

$$f_n(x,y)=S_n(x,y)-S_{p_{x(n)}}(x,y) -S_{p_{y(n)}}(x,y)+S_{p_{xy(n)}}(x,y) \qquad (6)$$

となる．ただし，$p_x(n)$を点nのX木の親の点，$p_y(n)$を点nのY木の親の点，$p_{xy}(n)$を点nのX木の親のY木の親の点とする．これは，図7に示すancの包含関係と，式3から明らかである(ただし親がいない時は$S=0$とする)．また，式6から，$f_n(x,y)$の

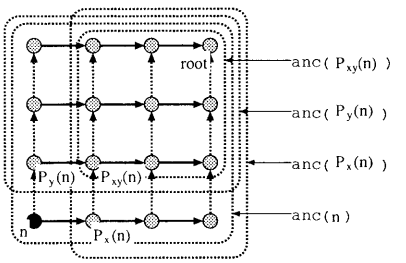

図 7: f_nの性質

変化量$\Delta f_n(x,y)$は式7のようになる．

$$\Delta f_n(x,y)=\Delta S_n(x,y)-\Delta S_{p_{x(n)}}(x,y) -\Delta S_{p_{y(n)}}(x,y)+\Delta S_{p_{xy(n)}}(x,y) \qquad (7)$$

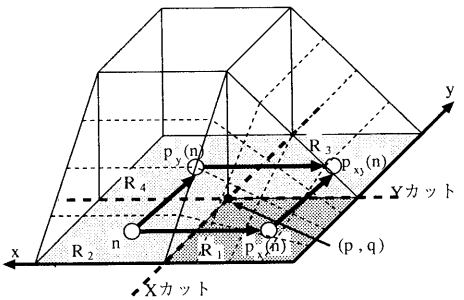

図 8: f_nの更新

例えば図8のように点が各領域に含まれていれば，$\Delta f_n(x.y)$は，

$$\Delta f_n(x,y)=pDy-Dxy-pqD+qDx \qquad (8)$$

となる．つまり，このように仮配線密度をD加算する操作をした時は，各ノードの保持する多項式に式7で示す式を加算すればよい．

ところでnと$p_x(n)$またはnと$p_y(n)$が同じ領域に含まれる場合，$\Delta f(x,y)=0$となり更新の操作は不要となる．実際には，次に定義するn_0の先祖集合$anc(n_0)$に属する点nに対してfの更新を行うだけで十分である．

領域 $[(1,1),(p,q)]$ の更新をおこなう時，$n_{p,q}$ のX木の右の子のY木の下の子を更新開始点 n_0 とする．ただし，$n_{p,q}$ がX木の右の子を持たない場合はY木の下の子を，Y木の下の子を持たない場合はX木の右の子を，両方持たない場合は $n_{p,q}$ を，n_0 とする．

以下，$anc(n_0)$ に対する更新だけで十分であることを示す．点 n_0 の対応する座標を (x_0,y_0) とおく．また，図8のように，R_1,R_3 と R_2,R_4 を分けるカットをXカット，R_1,R_2 と R_3,R_4 を分けるカットをYカットと呼ぶ．

すると，以下の性質が成り立つ．

性質 1 各 X 木において，ある点とその X 木の親とを結ぶ辺が X カットを越えるような点は，x_0 に対応する点から R_X（X 木のルート）までのパス上ある．

各 Y 木においても同様である．

証明 図9より明らか． ∎

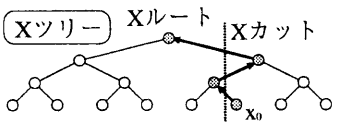

図 9: 性質 1

性質 2 $\forall n \notin anc(n_0)$ について，

$$\Delta f_n(x,y) = 0 \tag{9}$$

が成り立つ．

証明 点 n のX木の親への辺がXカットと交差し，かつY木の親への辺がYカットと交差すると仮定する．性質1から，n は，X木の x_0 から R_X に至るパス上にあり，かつY木の y_0 から R_Y に至るパス上にある．

各X木と各Y木がそれぞれ同型であることと，点の先祖の定義から $n \in anc(n_0)$ であることがわかる．

したがって，$n \notin anc(n_0)$ であれば，(1)n のX木の親への辺はXカットと交差しない，(2)n のY木の親への辺はYカットを交差しない，の少なくとも一方は成り立つ．

(1)の場合，n と $p_x(n)$ は同じ領域 R_i に属し，$p_y(n)$ と $p_{xy}(n)$ は同じ領域 R_j に属する（X木は同型であるから）．すると，式5と，式7から，

$$\begin{aligned}\Delta f_n(x,y) &= \Delta S_{p_{xy}(n)}(x,y) - \Delta S_{p_x(n)}(x,y) \\ &\quad -\Delta S_{p_y(n)}(x,y) + \Delta S_n(x,y) \\ &= 0\end{aligned} \tag{10}$$

(2)の場合も同様． ∎

性質2から，$anc(n_0)$ に関してのみ $f_n(x,y)$ の更新を行えばよいことがわかる．

この更新は式7に従っておこなわれ，式8がその例である．

4.3 初期化・更新・参照操作

以上のことから初期化・更新・参照の各操作は次のように実現できる．

初期化操作 全ての点 n に対して，$f_n(x,y)=0$ とする．

更新操作 $anc(n_0)$ の各点 n に対し，$f_n(x,y)$ に式7で示す多項式を加算する．

参照操作 領域 $[(1,1),(x,y)]$ の仮配線密度の合計は，

$$\sum_{n \in anc(n_{x,y})} f_n(x,y)$$

で参照する．

任意の矩形の密度の合計の参照操作は，原点を含む矩形の合計の計算を高々4回行うことで実現できる．

配線領域の面積を A とした時に，X木とY木の高さはそれぞれ $O(\log A)$ である．したがって，$|anc(n)| = O(\log^2 A)$ であるので，更新操作と参照操作は $O(\log^2 A)$ で実行できる．

また，初期化操作には $O(A)$ 必要である．

4.3.1 例

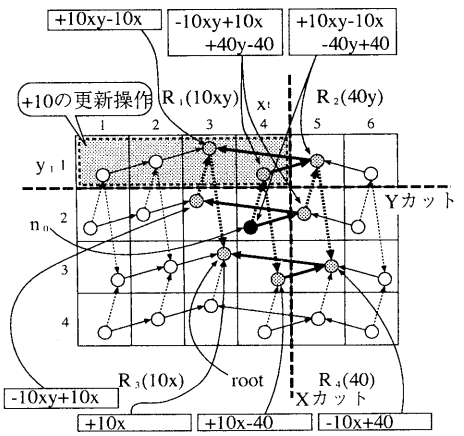

図 10: 更新操作の例

図10に示されるように，領域 $[(1,1),(4,1)]$ の仮配線密度を 10 加算する更新をしたとする．この時，点 $n_{4,1}$ には下の子がいるので更新開始点は $n_{4,2}$ となる．そして，太線上の 9 つの頂点で示される $anc(n_{4,2})$ について，図に示す多項式の更新を行う．

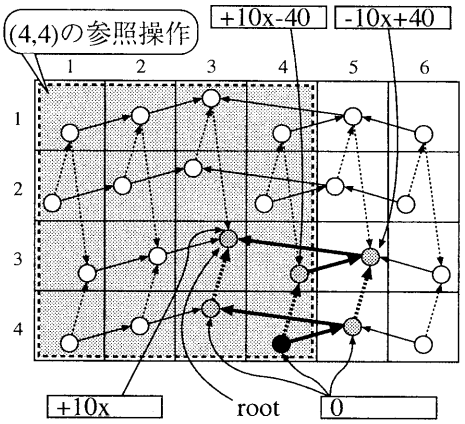

図 11: 参照操作の例

また，領域 $[(1,1),(4,4)]$ の合計を求める参照操作をしたいときには，図11中の太線上の 6 つの頂点の多項式を合計して $(10x)$，最後に $(x,y)=(4,4)$ を代入すればよい (40 となる)．

5 端点成長法の計算量

縦横比コストの計算は明らかに定数オーダ時間で実行可能である．進入コストと密度平均コストは，仮配線密度の矩形合計参照操作で計算できるので，計算量は面積 A に対して $O(\log^2 A)$ となる．したがって，成長コスト全体の計算時間は，$O(\log^2 A)$ となる．

成長させるべき端点とその方向を決定したあとで，実際にその端点を成長させる．具体的には，成長の履歴を記録し，端点の座標を移動して，仮配線密度を更新するが，計算量の観点からは仮配線密度の更新操作が支配的となる．

例えば図12の左に示すような成長が起こって，端点矩形への仮配線密度の影響が D_1 から D_2 になったとすると，図12右に示すような高々5 回の矩形領域更新操作で，仮配線密度の更新を実現出来る．このことから，端点成長操作は面積 A に対して $O(\log^2 A)$ の時間で実行可能である事がわかる．

端点成長法のアルゴリズム全体は前述した通り，$|N|$ 本の各ネットの各端点の各成長可能方向についての成長コスト計算と，1 回の端点成長操作を交互に繰り返す．ネットの本数 $|N|$ 本，ネットの平均長

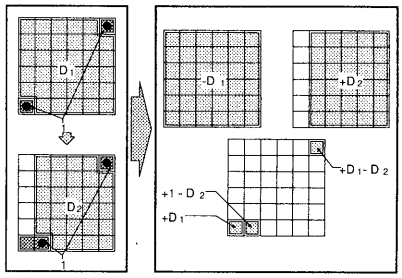

図 12: 端点成長と密度矩形更新

L とすれば，この繰り返しを $|N|L$ 回行うことになるので，結局最終的な計算量は $O(|N|^2 L\log^2 A)$ となる．

6 実験

6.1 配線結果

端点成長法の配線結果に対する効果を調べるために，次のような実験を行った．

10×10 のグリッドを用意し，ここに 200 対の端子をランダムに配置する．このような配線問題を 100 個作成して，以下の 7 種類のアルゴリズムを用いて配線を行う．そして総当り戦で各データ毎の勝ち負けを調べ，その勝敗数を数える．ただし，配線領域全体での配線密度の最大値の大小で勝敗を決める．実験したアルゴリズムは以下の通りである．この中で，α,β,γの値は，式 1のコスト計算に使われる重みの値である．

TG1:1:1 端点成長法 $(\alpha=1,\beta=1,\gamma=1)$

TG0:1:1 端点成長法 $(\alpha=0,\beta=1,\gamma=1)$

TG1:0:1 端点成長法 $(\alpha=1,\beta=0,\gamma=1)$

TG1:1:0 端点成長法 $(\alpha=1,\beta=1,\gamma=0)$

M1 単純な迷路法 (短いネットから配線)

M2 単純な迷路法 (長いネットから配線)

Conv 従来手法 [1]

この結果を表 1に示す．各項目でその項目に対応する行のアルゴリズムが対応する列のアルゴリズムに対し，勝ったデータ数が左側，負けたデータ数が右側に示される．例えば，TG1:1:1 は TG0:1:1 に 75 勝 2 敗 23 分けしていることになる．太文字で書かれている部分の通り，端点成長法が従来手法に比べて良好な配線結果を出力することがわかる．また，3

	TG1:1:1	TG0:1:1	TG1:0:1	TG1:1:0	M1	M2	Conv	合計
TG1:1:1	0/ 0	75/ 2	100/ 0	35/ 11	77/ 1	84/ 0	**43/ 7**	414/ 21
TG0:1:1	2/ 75	0/ 0	100/ 0	7/ 58	29/ 16	33/ 12	6/ 48	177/209
TG1:0:1	0/100	0/100	0/ 0	0/ 99	1/ 98	0/ 99	0/100	1/596
TG1:1:0	11/ 35	58/ 7	99/ 0	0/ 0	61/ 7	66/ 4	32/ 22	327/ 75
M1	1/ 77	16/ 29	98/ 1	7/ 61	0/ 0	27/ 20	2/ 58	151/246
M2	0/ 84	12/ 33	99/ 0	4/ 66	20/ 27	0/ 0	3/ 61	138/271
Conv	7/ 43	48/ 6	100/ 0	22/ 32	58/ 2	61/ 3	0/ 0	296/ 86

表 1: 実験結果 (配線結果)

種類のコストのどれが欠けても良好な結果が得られないことがわかる.

6.2 計算時間

端点成長法の計算時間を評価するために, 次のような実験を行った.

10×10 から 100×100 の 10 通りの配線領域に対して, 1000 ネットの配線を行い, 従来手法と計算時間を比較した. 使用マシンは DEC ALPHA STATION 600 5/333 で, gcc version 2.7.2 で実装した. 縦軸は計算時間 (秒), 横軸は配線領域の大きさである. その結果が図 13である.

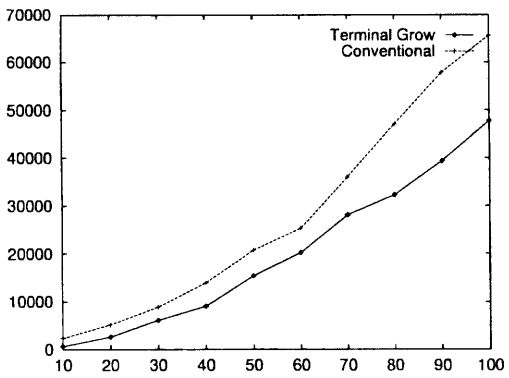

図 13: 計算時間 : 配線面積の影響

7 結論

本研究では, 従来手法のネット毎に経路を確定して行くことに対する弊害に着目し, すべてのネットを同時に配線していく新しいルーティングアルゴリズムを開発した. それを可能にしたアイデアは端点成長法である. また, 端点成長法を実装するにあたって, 時間的なボトルネックとなる仮配線密度の更新計算に関して, これを高速に実行できるデータ構造を提案した.

そして, 従来手法との配線結果および計算速度の点で比較実験を行い, その有効性を確認した.

謝辞

本研究を進めるにあたり有益な御検討を頂いた中武繁寿助手を始めとする梶谷研究室の諸氏に感謝する. また, 本研究の一部は, CAD21 研究体の援助による.

参考文献

[1] S.Nakatake,Y.Kajitani,"Channel-Driven Global Routing with Consistent Placement", Proc. International Conf. on CAD pp.350-355, 1994.

[2] Jingsheng Cong, Bryan Preas, "A New Algorithm for Standard Cell Global Routing", Proc. of ICCAD(1988) pp.176-179.

[3] Jonathan Rose, "Parallel Global Routing for Standard Cells", IEEE Trans. on CAD, pp.1085-1095, Vol.9, No.10, October 1990.

[4] J.T.Mowchenko, C.S.R.Ma, "A New Global Routing Algorithm for Standard Cell ICs", Proc. ISCAS. 1987, pp.27-30.

[5] Carl Sechen,Alberto Sangiovanni-Vincentelli, "TimberWolf3.2: A New Standard Cell Placement and Global Routing Package", Proc of 23rd Design Automation Conference pp.432-439 1986.

[6] Jan-Ming Ho, Gopalakrishnan Vijayan, C.K.Wong, "New Algorithms for the Rectlinear Steiner Tree Problem", IEEE Trans. on CAD, pp.185-193, Vol.9, No.2, February 1990.