

## 線分展開法の2層配線への 拡張とその評価

小島 直仁<sup>+</sup> 山田 正昭<sup>+</sup> 佐藤 政生<sup>++</sup> 大附 辰夫<sup>+++</sup>

<sup>+</sup> 株式会社 東芝 ULSI 研究所

<sup>++</sup> 拓殖大学 工学部

<sup>+++</sup> 早稲田大学 理工学部

VLSI などのパタン設計における詳細配線手法として、格子に基づかずレイアウト・データを図形の形で扱う配線手法“グリッドレス・ルータ”に関する研究が行われており、様々な手法が提案されている。線分展開法は実用的なグリッドレス・ルータのひとつであるが、2層配線への十分な対応はされていなかった。本稿では線分展開法を実用的な2層配線に拡張する際の問題点について考察し、この手法をヒープ探索木を用いてインプリメントし計算機実験を行った結果を示す。

## A Two-Layer Line-Expansion Algorithm and its Evaluation

Naohito Kojima<sup>+</sup> Masaaki Yamada<sup>+</sup> Masao Sato<sup>++</sup> Tatsuo Ohtsuki<sup>+++</sup>

<sup>+</sup> ULSI Research Center, Toshiba Corporation  
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 210, Japan

<sup>++</sup> Faculty of Engineering, Takushoku University  
815-1 Tatemachi, Hachioji, Tokyo 193, Japan

<sup>+++</sup> School of Science and Engineering, Waseda University  
3-4-1 Okubo, Shinjuku-ku, Tokyo 169, Japan

A new two-layer gridless router based on line-expansion algorithm is presented. This algorithm named **Improved Line-Expansion Algorithm** is the gridless version of the line-expansion algorithm and excellent in two-layer routing. It finds a path in  $O(k \log^2 n)$  time by means of priority search trees, where  $n$  is the number of vertices in a routable region and  $k$  is the number of vertices to be searched ( $k \leq n$ ). Experimental results show **Improved Line-Expansion Algorithm's** efficiency.

# 1 はじめに

従来よりレイアウト設計の核となる逐次配線手法としては、迷路法と線分探索法が最も普及している。これらの手法は本質的に配線領域上に格子構造を想定しているため処理が単純である(すなわちプログラムのステップ数が小さい)という利点があり、また十分小さい領域内の配線であれば妥当な処理時間で配線を行うことができる。しかし配線領域が大きくなるにつれて多大な処理時間とメモリを必要とし、格子構造に依存しているためプロセス技術の進歩に伴う設計規則の変化に追従しにくいという問題点を持つ。これらの問題点を解決するための一つのアプローチとして、格子構造に依存しない配線手法(グリッドレス・ルータ)に関する研究がここ十年來行われている。

文献 [1] で提案されている線分展開法(1, 2層配線可)を、1層配線において改良した手法である改良線分展開法 [2] は有効なグリッドレス・ルータのひとつであるが、本稿ではその手法を2層配線に対応させた手法(以降、[2] と併せてこの手法を“改良線分展開法”と記す)を提案し、文献 [1] ⇒ 文献 [2] における改良点とともに述べる。

以降、議論を簡単にするために配線領域は垂直および水平な線分で囲まれた領域(複合長方形領域: *rectilinear region*)とする。また、2層配線領域で2端子 S, T間の配線を行うものとし、このとき求める配線経路は斜め線を含まないものとする。

各配線経路はその中心線で、またビアは中心の点をもって表現するが、配線幅や最小間隔を考慮しているので設計規則に違反することはない(図1参照)。図1(b)の実線で示された正方形はビアの輪郭を表し、その中の点線で示された正方形はビアと接続する経路の中心線の範囲を示す。つまり、配線経路とビアの接続には多少のずれが生じることも可としている。

## 2 線分展開法のアルゴリズム

線分展開法の経路探索手法は基本的にはダイクストラ法に基づくものであり、複数存在する経路のうち最もコストの小さい経路を求めることを目的としている。ここで従来の線分展開法による経路探索の様子を簡単に説明する。

まず両端子 S, T を通る線分(この線分を active line と呼ぶ)を配線領域内に発生させ、この active line を

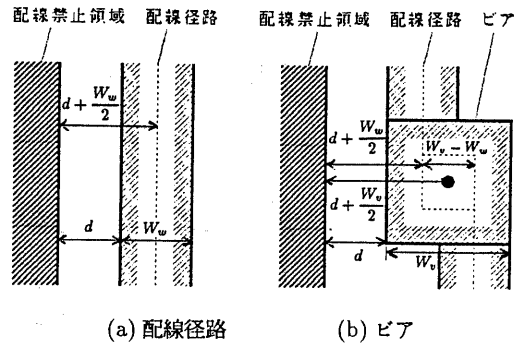


図1: レイアウト・データの表現形式  
Fig.1: Representation of layout data.

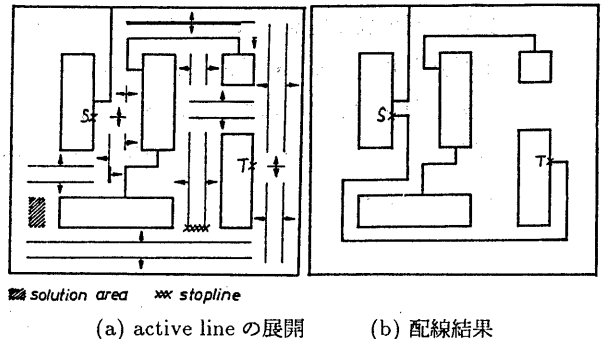


図2: 線分展開法による経路探索の様子(1層配線領域)  
Fig.2: An overview of routing by Line-Expansion Algorithm(1-layer).

垂直方向に掃引(展開)することによって到達可能な全ての配線領域を探索する(図3参照)。探索の際に、対となる端子から探索されてきた領域に遭遇しなければ、探索の結果得られた複合長方形領域の外周上に必要に応じて新たな active line を発生させこれらの active line に対して同様の領域探索を行う。同じ端子から探索されてきた領域と遭遇した場合は、その境界に stop line と呼ばれる障壁を設け、配線領域の多重探索を避ける。探索を繰返していき両端点からの探索領域が交差した場合(その交差領域を solution area と呼ぶ)そこから逆追跡を行うことにより経路を求める(図2(b)参照)。

文献 [1] では、この手法を2層配線領域に対応させた手法も提案されている。しかし、この手法は経路探索が完全に縦横原則に縛られており、またビアと配線

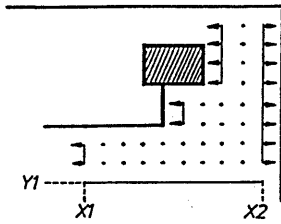


図 3: 線分展開法における active line の展開  
Fig.3: Expansion of active line on  
Line-Expansion Algorithm.

径路の幅の不一致に関する考察も不十分であるなどの問題があるため、このまま実用とするには適当ではない。

### 3 改良線分展開法のアルゴリズム

本章では、線分展開法の 2 層配線対応のための改良点を示し、次に改良線分展開法の径路探索のアルゴリズムについて報告する。

#### 3.1 従来の線分展開法に対する改良点

従来の線分展開法を実用にするために行なった改良点を以下に示す。

##### 1. 図形データを構造化する

従来の線分展開法における図形データの取扱いは、線分を座標でソートした上でリスト構造を作るといふ、格子構造を前提にする度合いがかなり大きいものであった [1]。そこで改良線分展開法では図形データを格子に依存せずしかも高速にデータ探索を行うことのできるデータ構造を用いて構造化する。本稿におけるインプリメンテーションでは図形データ構造にヒープ探索木 [3] を用いている。

- 一度の線分展開による探索領域を長方形に限定する 従来の手法では一度の線分展開で active line が到達可能な全ての配線領域を探索することにより、複合長方形の探索領域が得られていた (図 3 参照)。改良線分展開法では、線分展開の際 active line の一部が障害物に遭遇した時点で線分展開を一度打ち切り、新たな active line を発生する (図 5 参照)。

この手法と、次節以降に述べるコストを active line に与えることにより、必ずしも曲がりの数や縦横原則にこだわらない柔軟な径路探索を行うことが可能となる。

##### 3. 展開する active line にコストを与える

従来の線分展開法では未展開の active line の格納にスタックを用い、その展開は始点からの曲がりの数のみを考慮して行っている。改良線分展開法では、active line に曲がりの数だけでなく他の様々な要素を取り込んで計算した コスト を与えるものとし、このコストの算出のためにコスト関数を設定する。これにより様々な設計規則に対応しうる、様々な傾向の径路を得ることが可能となる。コスト関数の詳細については後述する。

##### 4. 2 層配線に伴うビアの取扱いを考慮する

2 層配線を行う時、ビアの発生は不可欠である。しかしビアが配線領域に占める幅は、径路のそれと比較して大きいことがほとんどである。格子に依存して配線を行う場合は、ビアが配置可能な間隔で格子を作成することによりこの問題に対応するが、グリッドレス・ルータでは径路やビアの間隔をできうる限り最小にして配線するため、層間の径路探索には十分な考察が必要である。本手法では異なる層へ active line を発生させる前にビアが配置可能であるか否かを判断する。詳細については後述する。

#### 3.2 改良線分展開法の径路探索手法

この節では改良線分展開法の径路探索手法の詳細について述べる。提案する改良線分展開法の径路探索手法は基本的には従来の線分展開法と同じである。

ここでまず以下の説明での用語の定義を行う。

**ヒープ** 未展開の active line を保持する。active line の持つコストによる優先度付き待ち行列の形で実現されている。ヒープの内容は径路探索前は空である。新たな active line はその発生とともにヒープに挿入される。最もコストの小さい active line がヒープから削除され、配線領域内で展開される。

**図形データ構造** 配線領域を構成する全てのレイアウト・データ (配線禁止領域の境界線分、既配線、端

子、active line 等)を構造化して管理し、これらに対する効率の良い幾何学的探索を提供する。経路探索前には active line を除く全てのレイアウト・データが格納されている。

前節における従来手法の説明では経路探索は両端点から行っていたが、ここでは説明を簡単にするためソース端子 S、ターゲット端子 T のうち端子 S からのみ経路探索を行うものとする。同時に複数の点から探索を行う場合でも、基本的な手法は同じである。

**Step 0** ヒープの初期化を行う。ヒープは最初、空である。

**Step 1** 配線領域内に S を通る active line を発生させる(図 4 (a1)、(a2) 参照)。active line 上に T が存在すれば、経路を得て Step 5 へ。active line を図形データ構造に挿入する。また、この active line にコスト 0 を与えた上で、ヒープへ挿入する。

**Step 2** ヒープから最もコストの小さい active line L をひとつ取り出す。ヒープが空なら経路は存在しない。この場合は Step 6 へ。

**Step 3** active line L の展開を行う(図 4 (b2)、(c1)、(d2)、図 5 参照)。展開によって得られた長方形領域を R とする。このとき、既に存在する active line は展開の障害物となるので、配線領域を active line が 2 重に展開することはない。R 内に T が存在する場合は Step 4 へ。

R の外周上に図 5 のように新しい active line を発生させる。この時、隣接する層へ active line を発生するが(図 4 (b1)、(c2)、(d1)、図 6 参照)、ビアの配置を十分に考慮して行なわなければならない。この処理については次節以降で詳しく述べる。ここで発生された active line にはすべて L (いわば自分の親である) へのポインタを持たせ、コストを与える。ポインタは Step 4 の逆追跡の際に必要となる。active line に与えるコストの要素については種々のものが考えられるが、これについては次節以降で詳しく報告する。

発生した active line をヒープと図形データ構造へ挿入する。ただし、active line が直線上で隣接する場合には 1 本に併合した上で挿入する。Step 2 へ。

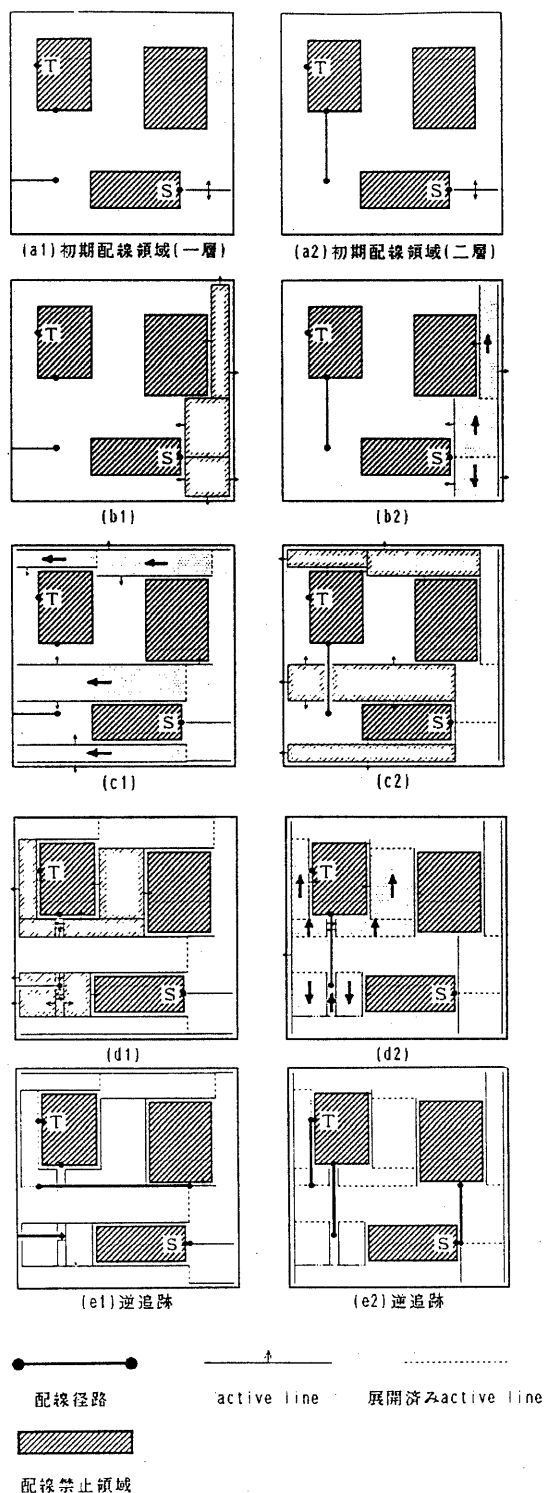


図 4: 改良線分展開法の経路探索手順(2層配線領域)

Fig.4: An overview of routing by Improved Line-Expansion Algorithm(2-layer).

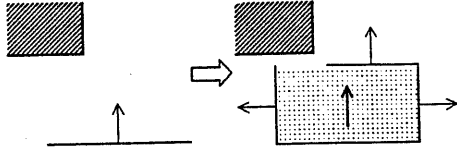
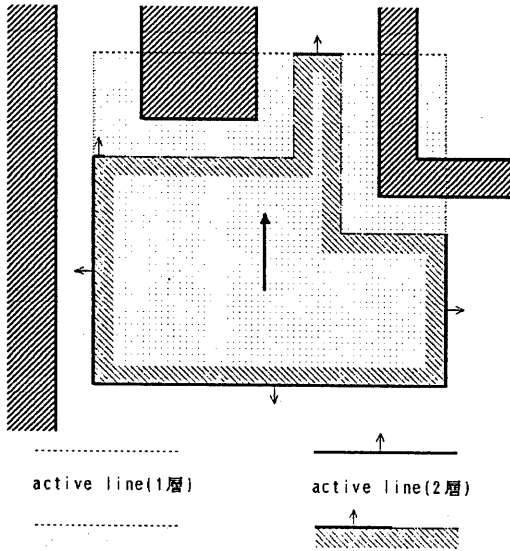


図 5: active line の展開 (1 層配線領域)

Fig.5 : Expansion of active line(1-layer).



径路配置可能領域(1層) 径路、ビア配置可能領域(2層)  
1層における線分展開の結果から2層へ active line を発生する

図 6: active line の展開 (2 層配線領域)

Fig.6 : Expansion of active line(2-layer).

Step 4 逆追跡を行い径路を得る (図 4 (e1)、(e2) 参照)。各 active line は自分の親の active line へのポイントを持っているので、逆追跡の際はそれを一方的に追跡して行くことで径路が得られる。

ここで、相互連結した、向かい合う active line を認識することにより「長方形領域」を考慮した逆追跡を行うことが容易に可能となり<sup>1</sup>、必ずしも active line と重ならない径路を得ることができ、ビアの発生にも柔軟に対応可能である。また、径路探索および逆追跡の際に配線可能な幅を認識できるので、束配線 (バス配線) などへの応用が考えられる。

<sup>1</sup>相互連結がなくても改めて領域探索を行えば処理時間は要するが同様の逆探索 (“追跡” とは呼び難い) が可能である。

Step 5 径路を構成する線分を図形データ構造に挿入する。

Step 6 図形データ構造から active line を削除し、終了。

### 3.3 コスト関数の要素

径路探索の過程で発生した active line にはその状態に応じたコストが与えられる。コストは線分の方角に傾斜をつけて設定している。これにより、同じ active line の中でもソース端子から離れた点ほどコストを大きくするなどの設定が容易に実現できる。図 7 では、複数の active line 間でその長さに応じてコストの値が連続して設定されている様子を表している。

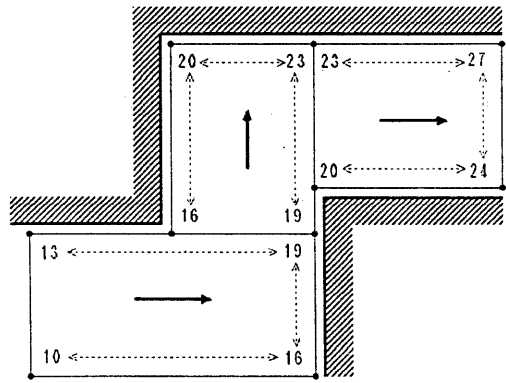


図 7: active line に与えられるコストの例

Fig.7 : Cost of active lines.

active line に与えるコストによって、最終的に得られる径路の傾向が決定するので、ユーザは自分の求める径路を得るために十分に考慮してコスト関数を決定しなければならない。コスト関数にはあらかじめ様々な要素が組み込まれているので、各要素の係数を調整することによって、得られる径路の性質 (あるいは傾向) を変更することができる。また、新たな要素を付加してより柔軟な径路探索を行なうことも可能である。

本節では、コストを決定する上で考慮している要素を示す。

ソース端子からの距離 一般にソース端子からの距離<sup>2</sup>が大きくなるとコストが大きくなる。最短距離の径路を求めようとする傾向を持つ。

<sup>2</sup>マンハッタン距離を用いることが多い。

ターゲット端子からの距離 一般にターゲット端子からの距離<sup>3</sup>が大きくなるとコストが大きくなる。径路探索の課程で、ターゲット端子との距離が小さい active line を優先的に展開させる目的に使用される。得られる径路の形状に極端な影響は与えない。本稿におけるインプリメントでは、文献 [4, 5] と同様のコストを採用している。

縦横原則に従う優先方向 縦横原則に反する径路を発生するような active line に大きなコストを与える。方法としては、1) 縦横原則に従わない active line のコストに一定の値を加算する、2) 縦横原則に従わない方向の active line のコストの傾きを急ににする、などがあるが、本稿のインプリメントでは後者を採用している。具体的には、「ソース端子からの距離」で述べたコスト “ $ax + b$ ” の  $a$  の値を縦横原則に従うか否かによって異なる値に設定する。この項の係数を大きくすると縦横原則をくずしにくくなり、その逆の場合は径路の曲がりと層の移動の相関が小さくなる。

層間の移動 縦横原則とは独立に、層の移動(すなわちビア)に対してコストを与える。本稿のインプリメントでは、この項よりも縦横原則の方を重視している。この項の係数を大きくすると、得られる径路において層の移動が少なくなる。

active line の長さ active line の長さに応じてコストを与える。active line の長さとはすなわち線展開の幅であり、この幅が大きいときはその周囲の配線領域が空いていることが期待できる。将来3層配線を H-V-H あるいは V-H-V で行うとき、1層と3層には同じ向きの径路が発生するが、径路探索の時点でこれらの層のうち、より空いている層を選んで径路を得ることも十分可能である。

### 3.4 2層配線における径路探索

2層配線領域の配線においてはビアの存在を考慮しなければならない。ビアが配線領域に占める幅は径路のそれを上まわるため、グリッドレス配線においては「径路は存在できるがビアの配置はできない」という領域が存在し得る。層を変えて active line を発生する場合はこのような領域に対する十分な考察が不可欠で

<sup>3</sup>マンハッタン距離を用いることが多い。

ある。本節では以上の問題に対する対策について述べる。

#### 3.4.1 active line 発生時の対策

図1に示すとおり、配線径路はその中心線を持って表される。その中心線と配線禁止領域の間の距離  $D_w$  は、配線径路と配線禁止領域間の最小間隔を  $d$ 、配線径路の幅を  $W_w$  とするとき  $D_w = d + W_w/2$  である。active line は配線禁止領域との間に常に上記  $D_w$  の距離を保っているため、逆追跡の際 active line と同じ位置に配線径路を発生可能である。しかしビアは配線禁止領域からさらに一定距離だけ離して配置しなければならない。ここでビアの幅を  $W_v$  とすると、active line が展開した領域の幅が少なくとも  $W_v - W_w$  だけあれば、その領域中にビアが存在可能である(図1参照)。

ある active line が配線領域内を展開し長方形領域が得られた時、同じ層の active line は長方形領域の輪郭に沿って発生する(図6参照)。ここで長方形領域の幅が  $W_v - W_w$  を下回る場合にはその領域内にビアは存在し得ないので隣接層には active line は発生しない。長方形領域の幅が  $W_v - W_w$  以上ある場合は隣接する層の配線領域内に長方形領域を投影し、その中に active line を発生する(図6参照)。このときビアの配置を保証するため長方形領域のどの部分をとっても幅が  $W_v - W_w$  以上なければならない。具体的には、この処理は“長方形領域内の図形データの列挙”を行うことにより実行される。

#### 3.4.2 逆追跡時の対策

前節の方針で active line を発生する限り、層を変えた逆追跡が起こる場合には必ずビアが配置可能である。しかしこの時単純に active line の上にビアを配置できる場合は極めて少なく、多少内側にずれた位置にビアを配置することになる。このため、逆追跡の際には active line の追跡のある程度先行した上で配線径路およびビアの位置を決定していく(図8参照)。

本手法の逆追跡によって得られた配線結果の中にはビアと配線径路の中心線が一致して見えない部分が発生することがあるが、レイアウト記述言語を用いる際はビア近辺で単層で曲がりを設ければ対応できる(図9参照)。一般に、ビアと配線径路の中心線が一致しない場合の方が、それを許さない場合よりもグリッドレ

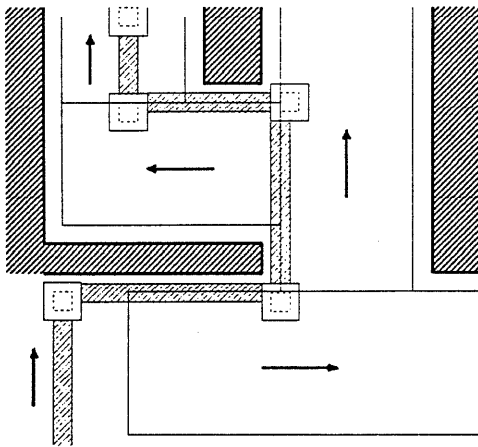


図 8: 2層配線領域における逆追跡

Fig.8 : Backtracking on 2-layer routing area.

ス・ルータの特長を生かした効率のよい配線結果を得ることができる。

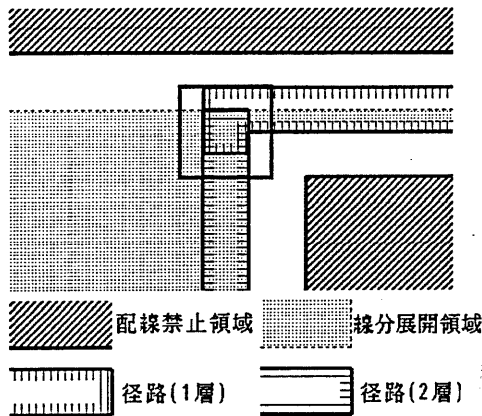


図 9: ビアと径路の接続の様子

Fig.9 : Connection between via and wire.

#### 4 計算機実験結果

改良線分展開法を計算機にインプリメントし、他の配線手法との処理時間の比較実験を行った結果を報告する。インプリメントにあたってはSun/SPARCstation 370上でC言語を使用した。レイアウト・データの取り扱いにはヒープ探索木[3]を用いた。実験は2層配線で行い、配線禁止領域、配線要求の端子は乱数によって発生させた。本手法との比較を行う手法としては従来の線分展開法[1](2層配線)を用いた。

処理時間の点で改良線分展開法が従来の線分展開法よりも優れている点は、主に次の2点である。

1. レイアウト・データを構造化して保持する。

ヒープ探索木を用いることにより、データの探索を $O(\log^2 n)$  (但し $n$ はデータの数)で行なうことができる。

2. active lineのコストにターゲット端子からの距離の要素を付加する。

ターゲット端子に近いactive lineが優先的に展開されることにより、径路探索における処理時間が小さくなる。

処理時間は、2層配線領域内の配線禁止領域の頂点数を変え、乱数で発生させた10種類の異なる2端子間を配線した結果の平均をとった。処理時間の測定結果を表1と図10に示す。

表 1: 線分展開法と改良線分展開法の径路探索における処理時間の比較

配線禁止領域の 頂点数	線分展開法 (Sec.)	改良線分展開法 (Sec.)
32	0.08	0.05
64	0.13	0.07
128	0.24	0.13
256	1.39	0.30
512	4.56	0.54
1024	10.17	0.83

図10によると、改良線分展開法の処理時間は配線禁止領域の頂点数にほぼ比例しているが、線分展開法は配線禁止領域の頂点数が大きくなるにつれて傾きを増している。これは線分展開法がレイアウト・データの取り扱いにリスト構造を用いていることにより、処理時間は配線禁止領域の頂点数の2乗に比例する。改良線分展開法ではターゲット端子からの距離の要素をactive lineのコストに加えることで、ターゲット端子に到着するまでに発生するactive lineの数を小さくし、これによっても処理時間が小さくなるが、やはりレイアウト・データを構造化して扱うことによる高速化が大きいと言える。以上の結果に加えて、径路探索における高い柔軟性を考慮にいれば、改良線分展開法は、有効な配線手法であると言える。

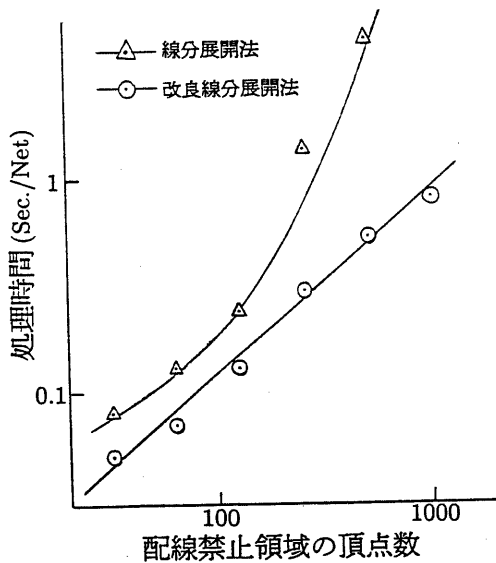


図 10: 線分展開法と改良線分展開法の径路探索における処理時間の比較

## 5 おわりに

以上、本稿では格子に基づかない詳細配線手法である改良線分展開法を提案し、従来の線分展開法との比較を行ないその有効性を示した。改良線分展開法は、その処理時間の点だけでなく、高い柔軟性を持った径路探索を行なうことにより様々な設計規則に対応することが可能である。

今後の課題としては、3層以上の多層配線への対応等が考えられる。

## 参考文献

- [1] W.Heyns, W.Sansen, and H.Beke: "A Line-Expansion Algorithm for the General Routing Problem with a Guaranteed Solution," Proc. 17th DA Conf., pp.243-249 (1980).
- [2] 小島, 佐藤, 大附: "線分展開法の改良とその評価," 設計自動化, Vol.48, No.6, pp.1-8(1989).
- [3] E.M.McCreight: "Priority Search Trees," SIAM J.Comput., Vol.14, No.2, pp.257-276(1985).
- [4] R.K.Korn: "An Efficient Variable-Cost Maze Router," Proc. 19th DA Conf., pp.425-431 (1982).

- [5] 高野, 山田: "チャンネルフリーゲートアレイ用の高速概略配線手法," 信学春季全国大会, No.A-267, p.1.270 (1989).