設 計 自 動 化 85－7
（1997. 10. 29）

# 線長の総和と最大に関する均衡平面スタイナー木

三林 秀樹， 高橋 篤司， 梶谷 洋司

東京工業大学 工学部 電気電子工学科
〒152 東京都 目黒区 大岡山 2-12-1
TEL：03-5734-2665， FAX：03-5734-2902
E-mail：{mitsuba, atushi, kajitani}@ss.titech.ac.jp

あらまし　配線幅の減少とともに，信号伝達遅延の中で配線遅延の占める割合が大きくなっている．また，配線遅延は配線総長の関数で近似されていたが，現在では信号源から伝達先までの線長の影響が無視できなくなり線長の総和と最大の関数で近似する必要がでてきた．我々は以前に総和と最大の割合を様々に変化させることができるCRBSTと呼ぶ手法を提案した．これは従来手法に比べて線長の総和，最大共に抑えられた矩形スタイナー木を生成する．しかし枝交差が観察されるなど，線長について明らかに改善の余地がある．そこで任意の矩形スタイナー木に対して線長の最大を増やすことなく総和を減少させる修正手法を提案する．結果的に枝交差はすべて除去され平面スタイナー木が得られる．実験によるとCRBSTで得られる矩形スタイナー木に適用すれば10%程度の改善が見られる．

キーワード　VLSIレイアウト，配線，遅延，スタイナー木

# Cost-Radius Balanced Plane Steiner Trees

Hideki Mitsubayashi, Atsushi Takahashi and Yoji Kajitani

Dept. of Electrical and Electronic Engrg., Tokyo Inst. of Tech.
Ookayama, Meguro, Tokyo, 152 Japan
TEL：+81-3-5734-2665， FAX：+81-3-5734-2902
E-mail：{mitsuba, atushi, kajitani}@ss.titech.ac.jp

Abstract　Though the interconnection delay has been estimated by the function of total wire length, we cannot now ignore the effect of the path length from the source terminal to the sink terminals. Therefore we should estimate the delay by the function of both the total wire length(*cost*) and the source-to-sink path length(*radius*). In the previous paper, we proposed an algorithm which constructs a rectilinear Steiner tree which is optimized according to various balance of the *cost* and *radius*. In this paper, we propose an algorithm which is used to modify a given rectilinear Steiner tree to reduce the *cost* without increasing the *radius*. An important feature of the algorithm is to eliminate all the existing crossings of edges. This is used as a post process for any Steiner tree algorithm.

key words　VLSI layout, routing, delay, Steiner tree

# 1 はじめに

近年のVLSI設計において，回路中の信号伝達遅延は回路の性能に大きな影響を与える要素の一つとなっている．特に配線が微細化するにつれて，配線遅延が信号伝達遅延に占める割合が大きくなって来ている．信号源から1つ以上の伝達先への伝達遅延を最小化するために様々な配線アルゴリズムが提案されている [1, 3, 4, 5, 7, 9, 10, 11].

Elmore遅延モデル [12] において配線遅延を最小化するためには，配線総長および配線木における信号源から各伝達先へのパス長を共に最小化する必要がある．配線総長については最小全域木 (*Minimum Spanning Tree, MST*) または最小スタイナー木により配線総長最小化が実現できるが，信号源から伝達先への非常に長いパスを含む可能性がある．一方，信号源からのパス長については最短パス木 (*Shortest Path Tree, SPT*) により信号源から各伝達先への最短パスを実現できるが，配線総長が過度に大きくなる可能性がある．配線木構成の困難さは配線総長 (*cost*) と信号源から各伝達先へのパス長 (*radius*) を共に小さく抑えることにある．さらに，一般的には配線レイアウトは縦横ルールで行われるため，矩形スタイナー木を構成する必要がある．

最小矩形スタイナー木問題および *radius* に制限を付けた最小矩形スタイナー木問題は共にNP-hardであることが知られている [8, 6]. このため，様々な発見的手法に基づくアルゴリズムが提案されている．

Alpertら [1] は *cost* と *radius* はトレードオフの関係にあることを示した．Pyoら [11] はKruskalのMSTアルゴリズムを応用して *radius* に制限を付けた矩形スタイナー木を構成するアルゴリズム (BKST) を提案した．我々はこれらの手法よりも *cost* と *radius* を共に小さく抑えた矩形スタイナー木を構成するアルゴリズム (CRBST) を文献 [2] で提案した．しかし，これらのアルゴリズムにおいて *radius* を強く抑えた矩形スタイナー木を構成すると，*cost* に大きな無駄が生じたり，枝に交差が生じることがある．CRBSTについては，矩形スタイナー木を構成する際に信号源に近い端子ほど先にレイアウトが確定する傾向があり，信号源から遠い端子のレイアウトを定めるときに，先に定まっているレイアウトがふさわしくないことが原因である．他のアルゴリズムについても構成方法は異なるが，やはり先に定まったレイアウトがふさわしくないことにより，*cost* の無駄や枝の交差が生じることがある．

本論文では矩形スタイナー木を入力として，それを修正する後処理のアルゴリズムを提案する．このアルゴリズムでは枝のレイアウトを修正することによって，*radius* を増やすことなく *cost* を減少させる．このアルゴリズムの効果はCRBSTで *radius* を重視して構成された矩形スタイナー木に対して顕著に現れる．さらに枝の交差はすべて除去されるため，木の平面化が実現できる．

# 2 諸定義

ネットの端子集合 $V = \{v_s, v_1, \ldots, v_n\}$ は $v_s$ を信号源，$v_1, \ldots, v_n$ を伝達先とするマンハッタン ($L_1$ metric) 平面上の任意の点集合とする．配線木 $T = (V_T, E_T)$ は $v_s$ を根として，$V$ を接続する有向矩形スタイナー木である．$V_T$ はすべての端子とスタイナー点 (端子でない分岐点) の集合で，$E_T$ は $V_T$ の要素を接続する枝の集合である．点 $v \in V_T$ を根とする部分木を $T_{sub}(v)$ とする．

$D(p_i, p_j)$ は平面上の2点 $p_i$ と $p_j$ (端子またはスタイナー点) の距離を表す．また $E_T$ 中の枝の長さはその枝の2端点間の距離である．木 $T$ 上の2点 $p_i, p_j$ に対して，$p_i$ から $p_j$ までの $T$ 上でのパス長 (経路上の枝長さの和) を $D_T(p_i, p_j)$ と表す．木 $T$ の枝長さの和 $W(T)$ を $T$ の *cost* とし，$R(T) = \max_{v \in V}(D_T(v_s, v))$ を *radius* とする．

また3点 $p_0, p_1, p_2$ について，$x$ 座標および $y$ 座標の中央値をそれぞれ $x, y$ 座標とする点 $p_m$ を，3点 $p_0, p_1, p_2$ の *middle-point* とする．このとき，異なる $i, j$ $(0 \leq i, j \leq 2)$ に対して，

$$D(p_i, p_j) = D(p_i, p_m) + D(p_m, p_j)$$

が成り立つ．

# 3 アルゴリズム CRBST

提案するアルゴリズムは任意のスタイナー木を入力とする．しかしその入力を与える現在知られている最良の方法CRBST(cost-radius balanced Steiner trees) を紹介する．従って本章は提案手法とは直接関係ないので，読み飛ばしてもかまわない．

アルゴリズム CRBST は $P$を定数とすると，$R(T) \leq P$を満たす矩形スタイナー木 $T$ を構成する．詳細については文献 [2] を参照されたい．以下 CRBST の概要について示す．

構成中の矩形スタイナー木に含まれる点および枝の集合をそれぞれ $V_T, E_T$とする．$e_{ij} \in E_T$は$v_i$から$v_j$への有向枝である．また$v_t$は$v_s$に最も近い点とする．CRBST は，$V_T = \{v_s, v_t\}, E_T = \{e_{st}\}$ を初期木として，以下の評価関数 $I(e_{ij}, v_k)$ をあらゆる $e_{ij} \in E_T, v_k \notin V_T$ の中で最小にするような点$v_k$を順次木に加えていく．ここで$v_m$は 3 点 $v_i, v_j, v_k$の *middle-point* である．

$$I(e_{ij}, v_k) = C(v_s, v_k, P) \cdot (D_T(v_s, v_i) + D(v_i, v_m)) + D(v_m, v_k),$$

ただし，$C(v_s, v_k, P) = D(v_s, v_k)/P$, $e_{ij} \in E_T$, $v_k \notin V_T$ .

Algorithm CRBST

1. $V_T = \{v_s, v_t\}$, $E_T = \{e_{st}\}$.

2. Let $e_{ij} \in E_T$ and $v_k \notin V_T$ be the pair which minimizes the cost function $I$.

3. $V_T = V_T \cup \{v_k, v_m\}$.
   $E_T = (E_T - \{e_{ij}\}) \cup \{e_{im}, e_{mj}, e_{mk}\}$.
   ($v_m$ is the *middle-point* of $v_i, v_j, v_k$).

4. If $V \subseteq V_T$, then goto step 5, else return to step 2.

CRBST によって構成された木には矩形でない枝 (斜め枝) が含まれることがあるが，この枝は矩形レイアウトを定める必要のなかったもので，適当な矩形レイアウトをとるものとする．

同じ点配置 $V$に対しても，評価関数 $I$にふくまれる $P$の値によって，構成される矩形スタイナー木の特徴が変化する．$P$は $\max_{v \in V} D(v_s, v)$ 以上の値に設定することができる．$P$を大きくすると，$I$は主に $v_k$を加えたときの *cost* 増加分を表すため，*cost* を重視した木が構成される．$P$を小さくすると，$I$は主に $v_s$から $v_k$までのパス長を表すため，*radius* を重視した木が構成される．CRBST の計算複雑度は $O(n^2 \log n)$ である．

# 4 後処理による木の修正

## 4.1 在来手法の欠点

他のアルゴリズムも同様であるが，CRBST では *radius* を重視するにつれて，*cost* に大きな無駄が目立つ矩形スタイナー木を構成することがある．例えば，図 1(a) のような入力に対して *radius* 重視で構成すると並行する枝の多い *cost* に無駄のある木ができる．これは $v_s$から近い点を順次加えていったのが裏目に出た例である．また図 1(b) のような入力に対して，*radius* 重視で構成すると枝に交差が生じる．しかし，枝の交差はスタイナー点を用いて $v_s$から他の各端子までのパス長を延ばすことなく除去できる．

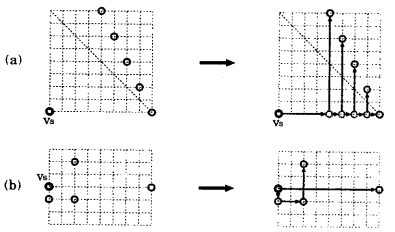



図 1: CRBST が悪い結果を出す例

このような *cost* の無駄および枝の交差を修正するアルゴリズムを以下に提案する．このアルゴリズムを用いることにより，図 1(a),(b) はそれぞれ図 2(a),(b) のように修正される．

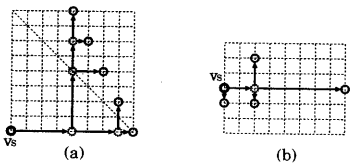



図 2: 後処理による木の修正

## 4.2 パス長を保存した木の修正

以下では，点 $a_1$から点 $a_2$への有向枝は $a_1a_2$と表すものとする．入力された矩形スタイナー木 $T$(斜め枝が存在

してもよい) に含まれる 2 枝 $a_1a_2, b_1b_2$について (図 3(a) 参照), $T$上での $v_s$から $a_2$までのパスを経路 $A$, $v_s$から $b_1$へのパスを通り平面上で $b_1$から $a_2$に至る経路を経路 $B$とする. このとき経路 $A$ の長さ $D_T(v_s, a_2)$ と経路 $B$ の長さ $D_T(v_s, b_1) + D(b_1, a_2)$ を比較して, 等しいまたは経路 $B$のほうが短い場合, 枝 $a_1a_2$を削除して枝 $b_1a_2$を加えるという修正をしても, $a_2$へのパス長が元のパス長 $D_T(v_s, a_2)$ より大きくなることはない. ただし, $v_s$から $b_1$への到達可能性を考慮すると $b_1 \neq a_2$でなければならない. このとき, 部分木 $T_{sub}(a_2)$ に含まれる点について $v_s$からのパス長が大きくなることはなく, それ以外の点についてはパス長は変化しない. つまり *radius* は増加しない.

木の修正は, 3 点 $b_1, b_2, a_2$の *middle-point* を $v_m$として 2 枝 $a_1a_2, b_1b_2$を削除して, スタイナー点 $v_m$および 3 枝 $b_1v_m$, $v_mb_2$, $v_ma_2$を加えることによりなされる (図 3(b) 参照). このとき, *middle-point* の性質より $D(b_1, b_2) = D(b_1, v_m) + D(v_m, b_2)$ である. つまり, $b_1b_2$間の *cost* は変化せず, $b_1$から $b_2$へのパス長も変化しない. よって, $D(a_1, a_2)$ が *cost* 減少分, $D(v_m, a_2)$ が *cost* 増加分になるので, $D(a_1, a_2) \geq D(v_m, a_2)$ が成り立っていれば *cost* が $D(a_1, a_2) - D(v_m, a_2)$ 減少する.

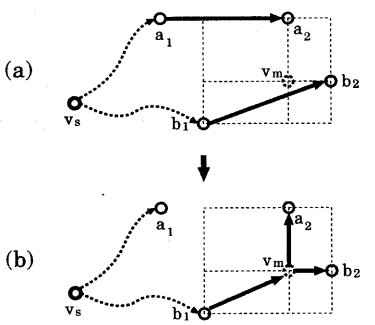



図 3: パス長を保存した木の修正

このように, 3 つの条件

- $D_T(v_s, b_1) + D(b_1, a_2) \leq D_T(v_s, a_2)$
- $D(a_1, a_2) - D(v_m, a_2) > 0$
- $a_2 \neq b_1$

を満たす 2 枝 $a_1a_2$, $b_1b_2$を **tm**(*tree modification*) **候補**, 図 3に表される修正を **tm 修正**と呼ぶことにする. tm 修正により, 各点までのパス長を延ばすことなく *cost* を減少させることができる.

## 4.3 アルゴリズム TM-1

tm 修正を利用した以下のアルゴリズム TM-1 を提案する.

Algorithm TM-1 :

入力となる木 $T$に tm 候補が存在するならば, tm 候補のうち $D(a_1, a_2) - D(v_m, a_2)$ が最大な候補について tm 修正を行い, これを tm 候補がなくなるまでくりかえす. tm 修正の際に長さ 0 の枝が生じた場合, 縮退して除去する.

**補題 1** アルゴリズム *TM-1* は有限回の繰り返しで終了する.

証明 : 入力となる木 $T = (V_T, E_T)$ の点集合 $V_T$から得られる *Hanan grid* (端子点を含む極小なグリッド)[7] について, 格子間隔の差の最小値を $p$ とすると, tm 候補が存在する場合, 木の修正でコストは $p$ 以上減少する. 与えられた点集合について, 最小矩形スタイナー木のコストを $W(MRST)$ とすると, TM-1 によってコストが $W(MRST)$ より小さくなることはないので, TM-1 の繰り返しは有限回である. □

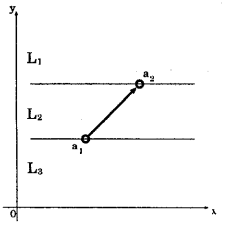
ここで, 2 枝が交差するとは, 2 枝のバウンディングボックス (枝を囲む最小矩形) に共通座標が存在する, と定義する. また, その共通座標が枝端点以外に存在するとき, 2 枝が枝端点以外で交差するという. このとき以下の補題が成り立つ.

**補題 2** *2* 枝 $a_1a_2$, $b_1b_2$が交差する場合,

$$D(a_1, a_2) = D(a_1, m(a)) + D(m(a), m(b)) + D(m(b), a_2)$$

である. ここで, $m(a), m(b)$ はそれぞれ *3* 点 $a_1, a_2, b_2$, $b_1, b_2, a_2$ の *middle-point* である.

証明 : 点 $a_1$の $x$ 座標, $y$座標をそれぞれ $a_1.x$, $a_1.y$のように表すものとする. 一般性を失わず, $a_1.x \leq a_2.x$, $a_1.y \leq a_2.y$とする. このとき $b_1$, $b_2$が以下のどの領域に存在するかで場合分けして考える. (図 4参照)




$L_1$ : $y > a_2.y$

$L_2$ : $a_1.y \leq y \leq a_2.y$

$L_3$ : $y < a_1.y$

図 4: 領域の分割

- $b_2$ *in* $L_1$ の場合 $m(a).y = a_2.y$ さらに,
  $b_1$ *in* $L_1$ $\Rightarrow$ 2 枝は交差しない
  $b_1$ *in* $L_2$ $\Rightarrow$ $m(b).y = a_2.y$
  $b_1$ *in* $L_3$ $\Rightarrow$ $m(b).y = a_2.y$
- $b_2$ *in* $L_2$ の場合 $m(a).y = b_2.y$ さらに,
  $b_1$ *in* $L_1$ $\Rightarrow$ $m(b).y = a_2.y$
  $b_1$ *in* $L_2$ $\Rightarrow$ $m(b).y = \max(b_1.y, b_2.y)$
  $b_1$ *in* $L_3$ $\Rightarrow$ $m(b).y = b_2.y$
- $b_2$ *in* $L_3$ の場合 $m(a).y = a_1.y$ さらに,
  $b_1$ *in* $L_1$ $\Rightarrow$ $m(b).y = a_2.y$
  $b_1$ *in* $L_2$ $\Rightarrow$ $m(b).y = b_1.y$
  $b_1$ *in* $L_3$ $\Rightarrow$ 2 枝は交差しない

以上より, $a_1.y \leq m(a).y \leq m(b).y \leq a_2.y$ である. また, 同様にして $a_1.x \leq m(a).x \leq m(b).x \leq a_2.x$ が得られる. よって, $D(a_1, a_2) = D(a_1, m(a)) + D(m(a), m(b)) + D(m(b), a_2)$ となる. □

**補題 3** *2* 枝 $a_1a_2$, $b_1b_2$ が交差する場合,

$$D_T(v_s, a_1) + D(a_1, b_2) \leq D_T(v_s, b_2)$$

または,

$$D_T(v_s, b_1) + D(b_1, a_2) \leq D_T(v_s, a_2)$$

が成り立つ

証明 : 背理法で証明する. $D_T(v_s, a_1) + D(a_1, b_2) > D_T(v_s, b_2)$ かつ $D_T(v_s, b_1) + D(b_1, a_2) > D_T(v_s, a_2)$ と仮定する. $D_T(v_s, a_2) = D_T(v_s, a_1) + D(a_1, a_2)$, $D_T(v_s, b_2) = D_T(v_s, b_1) + D(b_1, b_2)$ より両辺それぞれ加えてまとめると,

$$D(a_1, b_2) + D(b_1, a_2) > D(a_1, a_2) + D(b_1, b_2)$$

となる. ここで, *middle-point* の定義より

$$D(a_1, b_2) = D(a_1, m(a)) + D(m(a), b_2)$$
$$D(b_1, a_2) = D(b_1, m(b)) + D(m(b), a_2)$$

である. さらに補題 1 より,

$$D(a_1, a_2) = D(a_1, m(a)) + D(m(a), m(b)) + D(m(b), a_2)$$
$$D(b_1, b_2) = D(b_1, m(b)) + D(m(b), m(a)) + D(m(a), b_2)$$

なので, それぞれ不等式に代入してまとめると,

$$D(m(a), m(b)) < 0$$

が得られる. これは $D(m(a), m(b)) \geq 0$ に矛盾する. □

**補題 4** アルゴリズム *TM-1* の入力となる木 $T$ に *tm* 候補が存在しないならば, $T$ に含まれる任意の *2* 枝は枝端点以外で交差しない.

証明 : 対偶を証明する. 枝端点以外で交差している 2 枝を $a_1a_2, b_1b_2$ とする. 一般性を失わず, 補題 3 より $D_T(v_s, b_1) + D(b_1, a_2) \leq D_T(v_s, a_2)$ とする. このとき $a_2 = b_1$ ならば $D_T(v_s, a_1) + D(a_1, b_2) \leq D_T(v_s, b_2)$ が成立するため, 一般性を失わず $a_2 \neq b_1$ とする.

ここで $v_{m_1}$ を 3 点 $b_1, b_2, a_2$ の *middle-point* とする.

(i) $a_2$ が $b_1b_2$ のバウンディングボックス内にある場合.
このとき $v_{m_1} = a_2$ である. よって, $D(v_{m_1}, a_2) = 0$ となり $D(a_1, a_2) - D(v_{m_1}, a_2) > 0$ を満たすので tm 候補が存在する.

(ii) $a_2$ が $b_1b_2$ のバウンディングボックスの外にある場合.
$v_{m_1}$ は $b_1b_2$ のバウンディングボックス外周上にあり, また $D(v_{m_1}, a_2)$ は $b_1b_2$ のバウンディングボックスと $a_2$ の最短距離なので, $a_1a_2$ と $b_1b_2$ が交差することより, $D(a_1, a_2) \geq D(v_{m_1}, a_2)$ である. よって, $a_1 \neq v_{m_1}$ の場合 $D(a_1, a_2) - D(v_{m_1}, a_2) > 0$ となり tm 候補が存在する.
$a_1 = v_{m_1}$ の場合 (図 5参照), さらに 2 つの場合に分けて考える.

(a) $a_1 \neq v_s$ の場合.
$a_1 = b_2$ とすると枝端点以外で $a_1a_2$ と $b_1b_2$ が交差しないので $a_1 \neq b_2$ である. よって図 5のように $a_1$ を終点とする枝を $c_1a_1$ とする. このとき 3 点 $b_1, b_2, a_1$ の *middle-point* を $v_{m_2}$ とす

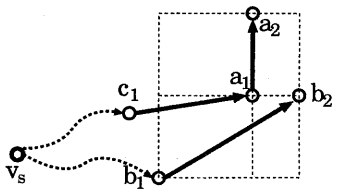



図 5: $a_1 = v_{m_1}$の場合

ると，$D_T(v_s,b_1) + D(b_1,a_2) \leq D_T(v_s,a_2)$ および $D(b_1,a_2) = D(b_1,a_1) + D(a_1,a_2)$ より，$D_T(v_s,b_1) + D(b_1,a_1) \leq D_T(v_s,a_1)$ が得られる．さらに，$a_1$が$b_1b_2$のバウンディングボックス外周上にあることより，前述の (i) の場合の $a_1a_2$を $c_1a_1$ と置き換えた場合と同様になる．よって tm 候補が存在する．

(b) $a_1 = v_s$の場合．
$D_T(v_s,a_1) + D(a_1,b_2) = D(v_s,b_2) \leq D_T(v_s,b_2)$ が成り立つ．さらに $a_1$は $b_1b_2$のバウンディングボックス外周上にあるので，3 点 $a_1,a_2,b_2$の *middle-point* を $v_{m_3}$とすると $v_{m_3}$も $b_1b_2$のバウンディングボックス外周上にある．$v_{m_3}$が $b_1$に一致するとき，2 枝 $a_1a_2, b_1b_2$は交差しない．よって $v_{m_3} \neq b_1$である．これより，$D(b_1,b_2) - D(v_{m_3},b_2) > 0$ となり，tm 候補が存在する．

以上より，すべての場合において tm 候補が存在することが証明された．□

**定理 1** アルゴリズム *TM-1* を適用した木については，任意の *2* 枝は枝端点以外で交差しない．(木の平面化)

証明：補題 1 および補題 4 より明らかである．□

### 4.4 アルゴリズム TM-2

TM-1 では，各端子までの $v_s$からのパス長を延ばさないという条件で木の修正を行った．TM-2 は各端子までの $v_s$からのパス長が，入力された木 $T$の *radius* $R(T)$ までは延びてもよいという条件で修正を行う．この制約の緩和により tm 候補の増大，したがって *cost* のさらなる減少が期待される．

具体的には，TM-1 での tm 候補の条件を以下のように変更する．端子 $v_i \in V_T$について $slack(v_i) = \min_{v \in T_{sub}(v_i)}(R(T) - D_T(v_s,v))$ とする．$v_s$から $v_i$までのパス長が最大 $slack(v_i)$ 延びても $R(T)$ は変化しない．このとき TM-2 における tm 候補の条件は，

1. $D_T(v_s,b_1) + D(b_1,a_2) \leq D_T(v_s,a_2) + slack(a_2)$
2. $D(a_1,a_2) - D(v_m,a_2) > 0$
3. 枝 $a_1a_2$は $v_s$から $b_1$のパス上の枝ではない

の 3 つである (図 3参照)．最後の条件判定は TM-1 になかったもので計算量の増大につながっている．

TM-2 も TM-1 と同様に，有限回の tm 修正で終了し，入力された木を平面化する．

## 5 実験結果

### 5.1 *cost* と *radius* のバランス

図 6はCRBST に後処理 TM-1, TM-2 を行った場合と行わない場合，および Alpert's[1], BKST[11] のそれぞれにおいて，構成された矩形スタイナー木の *cost* と *radius* のバランスを比較したものである．平面上の 50x50 のグリッドに 20 個の点をランダムに配置して，様々な $P$の値に対して矩形スタイナー木を構成した．各 $P$について 400 回の試行の平均値をとっている．縦軸は最短パス木 SPT の *radius* $R(SPT)$ に対する，構成された矩形スタイナー木 $T$の *radius* $R(T)$ の比を表す．同様に横軸は最小全域木 MST の *cost* $W(MST)$ に対する，$T$の *cost* $W(T)$ の比を表す．TM-1, TM-2 を行わなかった場合と比べて，*cost* 重視の部分ではそれほど差は見られないが，*radius* 重視になるにつれて，後処理を行わなかった場合より大幅に *cost* が改善されている．最も *radius* 重視の部分では CRBST のみと比べて，TM-2 を行った場合の *cost* は 9% 程減少している．

### 5.2 tm 修正の回数および時間

提案したアルゴリズムでは tm 修正実行のたびに tm 候補の探索をすべての枝対についておこなわねばならな

いので，理論的最悪計算量は非常に大きい．しかしtm修正実行回数は実問題では計算量に影響を与えるほど多くはないと期待されるので，実験によってtm修正回数と時間を確かめた．

TM-1，TM-2においてtm修正の行われる回数について実験を行った結果が表1である．ランダムに配置されたsize個の点に対してCRBSTで矩形スタイナー木を構成し，それをTM-1, TM-2の入力とした．CRBSTで用いられる評価関数内の変数$P$は$P = \frac{\max_{v \in V} D(v_s, v)}{c}$に設定した．$c$は0以上1以下の値をとり，$c = 0$のときに最も *cost* 重視，$c = 1$のときに最も *radius* 重視となる．それぞれの$c$に対して400回試行したときのtm修正の回数の平均値と最大値および1回の試行にかかった平均時間を示してある．tm修正は *radius* を重視した矩形スタイナー木が入力されたときに多く行われていることがわかる．またその回数は最も *radius* 重視のときでも点数に対して平均で1割ほど，最大でも点数の半分程度となっている．

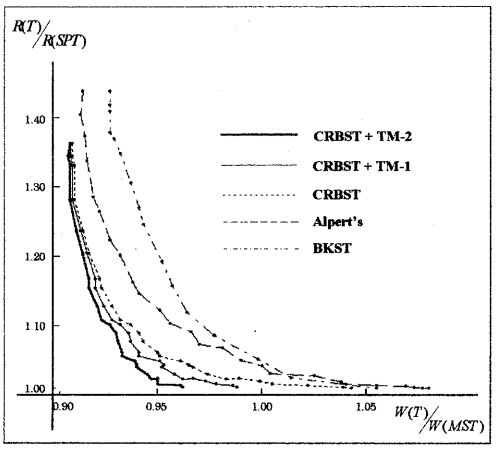



図 6: *cost* と *radius* のバランス

## 6 まとめ

伝達遅延を軽減するための配線手法として，以前提案したアルゴリズムCRBSTによって構成された矩形スタイナー木に対する後処理TM-1, TM-2を提案した．これによって木の平面化を実現し，さらに *cost* と *radius* を抑えた矩形スタイナー木を構成することができた．

任意の木を入力として，それにtm修正を施すことにより，良い矩形スタイナー木生成アルゴリズムとすることができる．そのときに入力の *radius* を増やさないことが保証されている．したがって，例えば最短パス木を入力とすれば *radius* を重視した結果が得られるであろう．それがCRBSTで *radius* を重視した場合と比べて，計算量および *cost* についてどちらが優れているかは興味ある話題である．

表 1: tm修正の回数および時間

| size | $c$ | TM-1 | | | TM-2 | | |
|---|---|---|---|---|---|---|---|
| | | avg | max | time (ms) | avg | max | time (ms) |
| 10 | 0.00 | 0.010 | 1 | 0.618 | 0.025 | 2 | 0.653 |
| | 0.50 | 0.317 | 2 | 0.877 | 0.427 | 3 | 0.983 |
| | 0.75 | 0.595 | 4 | 1.097 | 0.767 | 5 | 1.103 |
| | 1.00 | 1.205 | 9 | 1.545 | 1.385 | 8 | 1.683 |
| 20 | 0.00 | 0.047 | 2 | 2.932 | 0.123 | 2 | 3.055 |
| | 0.50 | 0.650 | 3 | 5.790 | 1.023 | 5 | 5.730 |
| | 0.75 | 1.077 | 5 | 7.330 | 1.720 | 7 | 8.438 |
| | 1.00 | 2.307 | 13 | 10.89 | 2.830 | 13 | 11.23 |
| 30 | 0.00 | 0.085 | 2 | 7.115 | 0.238 | 3 | 7.253 |
| | 0.50 | 0.863 | 5 | 14.99 | 1.543 | 6 | 15.63 |
| | 0.75 | 1.447 | 8 | 22.84 | 2.697 | 8 | 24.05 |
| | 1.00 | 3.187 | 16 | 36.40 | 4.430 | 16 | 37.79 |

## Acknowledgement

本研究はCAD21(東京工業大学)の研究の一環として行われているものである．

## 参考文献

[1] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng and D. Karger, "Prim-Dijkstra tradeoffs for improved performance driven routing tree design," IEEE Trans. Computer-Aided Design, vol.14, no.7, pp.890–896, July 1995.

[2] H. Mitsubayashi, A. Takahashi, Y. Kajitani, "Cost-Radius balanced spanning/Steiner trees" IEICE

Trans. Fundamentals, vol.E80-A, no.4, pp.689–694, April 1997.

[3] K. D. Boese, J. Cong, A. B. Kahng, K. S. Leung, and D. Zhou, "On highspeed VLSI interconnects: analysis and design," Proc. Asia-Pacific Conf. Circuits Syst., pp.35–40, Dec. 1992.

[4] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C .K. Wong, "Provably good performance-driven global routing," IEEE Trans. CAD, vol.11, no.6, pp.739–752, June 1992.

[5] J. Cong, K. Leung, and D. Zhou, "Performance-driven interconnect design based on distributed RC delay model," Proc. ACM/IEEE Design Automat. Conf., pp.606–611, June 1993.

[6] M. Garey and D. S. Johnson, "The rectilinear Steiner problem is NP-complete," SIAM Journal of Applied Mathematics, vol.32, no.4, pp.826–834, April 1977.

[7] M. Hanan, "On Steiner's problem with rectilinear distance," SIAM Journal of Applied Mathematics, pp.255–265, March 1966.

[8] J. Ho, D. T. Lee, C. H. Chang, and C. K. Wong, "Bounded diameter spanning trees and related problems," Proc. ACM Symp. Computational Geometry, pp.276–282, 1989.

[9] J-M. Ho, G. Vijayan and C. K. Wong, "New algorithms for rectilinear Steiner trees," IEEE Trans. Computer-Aided Design, vol.9, no.2, pp.185–193, Feb. 1990.

[10] S. Khuller, B. Raghavachari, and N. Young, "Balancing minimum spanning and shortest path trees," Proc. ACM/SIAM Symp. Discrete Algorithms, pp.243–250, Jan. 1993.

[11] I. Pyo, J. Oh, and M. Pedram, "Constructing minimal spanning/Steiner trees with bounded path length" Proc. European Design & Test Conf., pp.244–249, March 1996.

[12] J. Rubinstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," IEEE Trans. Computer-Aided Design, vol.2, no.3, pp.202–211, July 1983.