# Flip により自己変換するスタイナ木とその VLSI 最適配線への応用

久保 ゆき子[†] 高島 康裕[††]
中武 繁寿[†††] 梶谷 洋司[†]

平面配線領域上に指定された複数の点を相互に結ぶスタイナ木を別のスタイナ木へ変換するアルゴリズム Flip を提案する．これは，現スタイナ木の 1 枝をそれを含む面上のパスで迂回させる操作である．Flip は高速で働き，また任意のスタイナ木から任意のスタイナ木へ多項式回の適用で到達できることが証明できるので，VLSI 配線における複雑な評価に対する有効なアルゴリズムになる．Flip は，2 層 HV 配線領域のスタイナ木にも適用できるよう拡張される．例として採用した総配線長，クロストーク，スキュー，VIA 数などの複合評価関数に対応する実験では，従来の構成的手法では達成できない特徴を有する結果を得ることができた．また，類似の手法である rip-up and reroute 手法との差異を付録で論ずる．

# Self-reforming Steiner Trees by Flip and Applications to VLSI Interconnection

Yukiko Kubo,[†] Yasuhiro Takashima,[††] Shigetoshi Nakatake[†††] and Yoji Kajitani[†]

Given a Steiner tree that connects designated terminals on a plane routing area, the Flip is defind as a procedure that makes a current Steiner tree change its configuration. It is to replace one of the edges of the Steiner tree with a detour on an adjacent face. Since the procedure runs quickly, and the reachability is proved, the idea can be a useful routing algorithm if it is implemeted by simulated annealng. The idea is enhanced to the 2-layer HV routing. The performance of the algorithm was experimented for multiple objectives. Some unique and satisfiable results were observed. The difference from the rip-up and reroute strategy is discussed in Appendix.

## 1. はじめに

配線領域上に指定された端子を相互に接続する配線，すなわちスタイナ木，の実現問題は，単一目的関数による最適化についてはかなり深く研究されてきた．たとえば，総配線長最小化は古典的テーマであり，配線領域の格子化，指定点を中心とする半径制限，各端子への長さ，スキュー，VIA 数など VLSI 配線への応用を志向した最適化問題などがあげられる．そして，これらを解くアルゴリズムの開発努力と計算機の高速化によって，それぞれを実用的な意味で満たす解法はほぼ確立しているといえよう．本文で着目する個別評価関数に対しても，ごく一部をあげれば，文献 1)〜6) などが成功した例である（網羅的な文献としては文献 7) を参照されたい）．

しかし，回路の微細化と高速化にともなって配線モデルは精密化され，3 次元容量，インダクタンスなどの非線形モデルに起因する遅延，電力消費，クロストーク，温度依存特性，製造時のばらつき，その他様々な無視できない要因がいっせいに浮かび上がってきた．それら全部を，必然的に適当なバランスをとってであるが，低く抑えたいという要求に（できないならできない理由をつけて）応えなければならない．こうして，時代は再び「配線は容易ではない」に帰っている．

この事態に対処するために，配線という作業を根本から見直す動きがある．すなわち,「配線を最適に構成する（構成的アルゴリズム）」という発想から,「あらゆる配線から最適なものを選ぶ（探索的アルゴリズム）」

† 東京工業大学工学部電気電子工学科
Department of Electrical and Electronic Engineering, Tokyo Institute of Technology
†† 北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science
††† 北九州大学国際環境工学部設置準備室
Faculty of International Environmental Engineering, Kitakyushu University

への移行である．これは，実用的解を求めるアルゴリズムの本質に触れている．構成的アルゴリズムは，評価に依存して解を構成していくため，評価そのものが厳密に定義されていなければならない．したがって，評価そのものをレイアウトの関数として陽に書き上げる（たとえば，遅延は信号伝播距離に比例する）ことが困難になっている，という背景では，構成的アルゴリズムは適合しないアルゴリズムであるともいえる．実際，評価が計算公式として書き上げることができない場合の対応策として，環境パラメータのテーブルになっている方式すら提案されている[6]．

このような理由に加えて，近年の計算機環境の著しい発展を追い風として，探索的手法が効果的になってきている．構成的解法に対する絶対的な利点は，解を得てから評価することである（そして，基準を満たさなければ捨てる）．したがって，計算方法が確立してさえいればどのような評価でもかまわない．また，評価だけではなく，複雑な制約が課せられている最適化問題の解法にも有効である例があげられる（たとえば，文献 8)）．

この手法では，いかなる評価に対しても最適解を逃さないことが要求されているので，あらゆる候補を尽くすことが保証された生成方法が必要である．また，実用的には評価勾配に沿うサンプル探索になるので，評価の連続性（近い候補には近い評価値を与える）が必要である．そして，全探索に近づくために各候補が超高速で生成されなければならない．この「全列挙」,「連続性」,「高速性」が探索手法の鍵である．

我々は平面グラフのスタイナ木について，上の条件を満たす探索的解法を提案する．生成は，現在の解から次の解を生成する Flip と呼ばれる手続きを使う．Flip は，現配線の 1 枝をそれを含む面上のパスで迂回させることにより変換する（連続性）．これは平均的に枝数のオーダよりはるかに小さい計算量である（高速性）．また，現在解から任意の解に枝数のオーダで到達できることが証明できる（全列挙）．

さらに，平面グラフのスタイナ木およびそれに対する Flip の自然な拡張として 2 層 HV 配線への適用方法を示す．

解変換操作として Flip を採用するシミュレーテッドアニーリング（SA）を実装し，実験を行った．複合評価として，従来から個別最適化が試みられている総配線長，クロストーク，VIA 数，スキューを採用した．

実験で適用したデータは小規模であるが，たとえばスキューについては，H-tree や geometric matching を用いた構成的アルゴリズムではとうてい得られないような興味ある形状のスタイナ木を得ることが示された．構成的なスキュー最小化アルゴリズムで周辺状況を取り入れるためには，様々な工夫が必要であるが，本探索的手法では Flip が自律的に最適化に向かう．実験では，その様子が観測され興味深い．

## 2. 準　　備

配線領域はより一般的な場合を考慮し，平面描画された平面グラフ $G(V,E)$（$V$：点の集合，$E$：枝の集合）で与える．配線要求は平面グラフ $G$ 上に与えられた端子集合 $R \subset V$ で定義される．ここで説明を分かりやすくするために，与えられる平面グラフは 2 点連結グラフとする．もしそうでない場合は，与えられたグラフを 2 点連結成分に分解し，その成分中に端子が含まれているものだけを個別に解き，全体を統合すれば目的を達成する．そのとき分解した際の関節点には端子を置く．

与えられた端子集合を接続する枝集合を**スタイナ枝集合**と呼ぶ．特に，どの枝を消去してもスタイナ枝集合でなくなるときこれを**スタイナ木**と呼ぶ．

スタイナ枝集合に，端子ではなくかつ次数が 1 であるような点を端点に持つ枝が存在するとき，そのような枝を**冗長な枝**と呼ぶ．

**図 1** のスタイナ枝集合 $S$ において，$j$ は冗長な枝である．これを除くと，新たに $k$ が冗長な枝となる．このような枝をすべて除く操作を定義する．

procedure **冗長除去（スタイナ枝集合 $S$）**
**step 1:** もし，冗長な枝がなければ，終了．
　そうでないとき
**step 2:** $S$ から冗長な枝を取り除く．
**step 3:** **step 1** に戻る．
end procedure

**補題 1** 与えられたスタイナ枝集合は，サイクルや冗長な枝を含まないとき，およびそのときに限り，スタイナ木である．

スタイナ枝集合 $S$ が与えられるとき，サイクルが存在する限りサイクル上の 1 枝を取り除く操作を繰り返し，続いて冗長除去を行う一連の操作を $S$ の**最小化**と呼ぶ．図 1 において $\{j,k,m\}$ または $\{j,k,d\}$ または $\{j,k,f\}$ の枝の除去がスタイナ枝集合 $S$ の最小化に相当する．

サイクルは平面を 2 つの領域に分割する．サイクルを時計回りに探索するとき右側にある領域を**内部領域**，反対を**外部領域**と呼ぶ．サイクルが与えられるとき，そのサイクル上の異なる 2 点を接続し，サイクル上を通過しないパスを**コードパス**と呼ぶ．コードパスが内

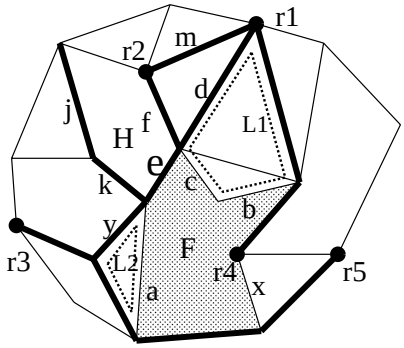



**図1** 面 $F$（影の部分），端子集合 $R=\{r1,r2,r3,r4,r5\}$，スタイナ枝集合 $S$，冗長な枝 $j$，スタイナ木 $T=S-\{j,k,m\}$
Fig. 1 Face $F$ (shaded), teminal set $R=\{r1,r2,r3,r4,r5\}$, Steiner edge set $S$, pendant edge $j$ and Steiner tree $T=S-\{j,k,m\}$.

部領域に存在しないサイクルを**面サイクル**，その内部領域を**面**と呼ぶ．また，与えられたグラフの外領域も面である．

枝 $e$ はちょうど2つの面サイクルに含まれる．図1において，枝 $e$ は2つの面サイクル $\overline{F}$ と $\overline{H}$ に含まれる．

**補題 2** スタイナ木 $T$，枝 $e$，面サイクル $\overline{F}$ が $e\in T\cap\overline{F}$ を満たすとき，

(1) $T'=T\cup\overline{F}-\{e\}$ はスタイナ枝集合である．

(2) $T'$ の最小化を以下のように行うとき，一意のスタイナ木が得られる．
(a) $T'$ のサイクルに含まれている $\overline{F}$ 上の枝すべてを除去する．
(b) 冗長除去を適用する．

**証明**

(1) $T\cup\overline{F}$ がスタイナ枝集合であることは明らかである．このとき，枝 $e$ の両端点を結ぶパスは $\{e\}$ と $\overline{F}-\{e\}$ の2つ存在する．よって，このスタイナ枝集合から $e$ を取り除いても $\overline{F}-\{e\}$ が存在するので非連結にならない．

(2) 最小化の操作自体が一意なので，得られるスタイナ木も一意である． □

図1において，スタイナ木を $T=S-\{j,k,m\}$ とすると，$T'=T\cup\overline{F}-\{e\}$ は2つのサイクル $\overline{L1}$，$\overline{L2}$（図の点線）を含む．補題に従い，指定されたように最小化を行うとスタイナ木は $T'-\{a,b,c,y\}$ と一意に定まる．

## 3. Flip

Flipはスタイナ木 $T$，その1つの枝 $e$，および面サイクル $\overline{F}$ に対して定義される．

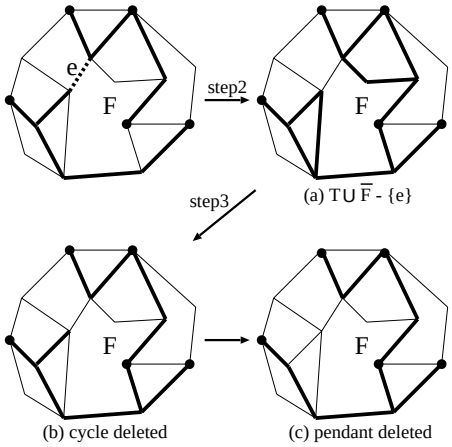



**図2** Flip $(T,e,\overline{F})$ の実行例
Fig. 2 Example of Flip $(T,e,\overline{F})$.

procedure **Flip** $(T,e,\overline{F})$
**step 1:** もし，$e\notin T\cap\overline{F}$ ならば，$T$ を出力する．そうでないとき，
**step 2:** $T=T\cup\overline{F}-\{e\}$
**step 3:** $T$ の最小化を補題2のように行う．
end procedure

Flipの適用例を**図2**に示す．なお，枝 $e$ を指定するとそれを含む面サイクルは枝の両側に存在する．Flipでは $e$ に代わる迂回路としてどちらかの面サイクル上のパスを与える．そのどちらかを指定する引数が $\overline{F}$ である．

**定理 1** Flip $(T,e,\overline{F})$ はスタイナ木を一意に出力する．

**証明** Flipの手続きの中で，**step 1** と **step 2** についてはその操作が一意であることは明らかである．**step 3** についても補題2より操作が一意に定まる． □

任意のスタイナ木から任意のスタイナ木へ多項式回数のFlipの繰返しで変換することが可能であるとき，前者は後者に**P-到達可能**である，という．本文で提案する最適配線探索は，評価にどのような仮定をも置かないので，どの解も最適解でありうる．したがって，多項式計算量で最適解に行き当たる可能性を保つためには，P-到達可能性が保証されていることが重要である．

**定理 2** 任意のスタイナ木はたかだか $\mu=|E|-|V|+1$ 回のFlipを適用することにより，任意のスタイナ木へ到達可能である．

**証明** 任意のスタイナ木の対 $(T_a,T_b)$ を考える．そして，$T_a$ にFlipをたかだか $\mu$ 回繰り返せば $T_b$ に至

る方法があることを示す．

まず，以下の用語を定義し，定理の証明にともなう補題とその証明を与える．

$T_a \cup T_b$ に含まれるサイクルの内部領域の和を $B$，それを囲むサイクルの集合を $\overline{B}$ とする．$d(T_a, T_b)$ を $B$ に含まれる異なる面の数とする．この関数は 2 つのスタイナ木の差を示す 1 つの指標である．

明らかに，$T_a = T_b$ のとき，およびそのときに限り $d(T_a, T_b) = 0$ である．また，$T_a \neq T_b$ のとき，$\overline{B} \cap (T_a - T_b) \neq \emptyset$ であることに注意されたい．

**補題 3** 枝 $e \in \overline{B} \cap (T_a - T_b)$ に対し，$e \in \overline{F}$ かつ $F \subset B$ である面サイクル $\overline{F}$ を選ぶ．Flip $(T_a, e, \overline{F})$ を適用し得られたスタイナ木を $T'_a$ とすると，$d(T'_a, T_b) < d(T_a, T_b)$ である．

**証明** まず与えられた $e, \overline{F}$ の条件より $T_a \neq T'_a$ である．$T'_a \cup T_b$ に含まれるサイクルの内部領域の和を $B'$ とすると，$e \in \overline{B}$ であることから，$B'$ は，領域 $B$ の外側から 1 面 $F$ を取り除いたものとなる．よって，$B' \subseteq B - F$，すなわち $d(T_a, T_b) - d(T'_a, T_b) \geq 1$ となる． □

この補題より，たかだか $d(T_a, T_b)$ 回の Flip で $T_b$ に至る．その回数はたかだか平面グラフ $G$ の面の個数で，これはよく知られているように $|E| - |V| + 1$ である． □

**図 3** に変換の例を示す．図において影の部分が $B$ に相当する．図の (a) では，$d(T_a, T_b) = 5$ である．(a) において $T_a$ は Flip $(T_a, e, \overline{F})$ により (b) の $T_a$ のように変換され，$d(T_a, T_b) = 4$ となる．さらに，(b) において $T_a$ は Flip $(T_a, e, \overline{F})$ により (c) の $T_a$ のように変換され，$d(T_a, T_b) = 2$ となる．さらに (d) を経て，(e) で $d(T_a, T_b) = 0$，すなわち $T_a = T_b$ となる．

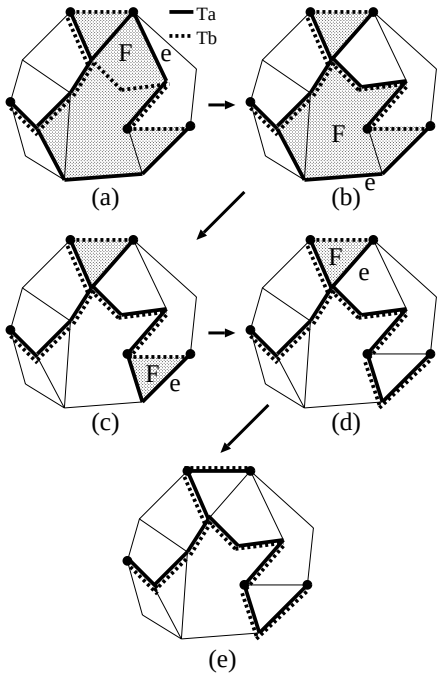



**図 3** Flip の繰返しによるスタイナ木の変換例
Fig. 3 Example of transformation of a Steiner tree by repetition of Flip.

## 4. HV 配線における Flip

Flip は平面グラフ上で定義した．そのため，スタイナ木の 1 枝を迂回させるパスを指定するのにその枝を含む面サイクルを利用した．これは平面レイアウトにおけるスタイナ木としては最小の変更であり，評価の連続性を最大限に保つ．Flip 導入の本来の動機はここにあった．しかし，Flip そのものは与えられたスタイナ木の任意の枝を含むサイクルを指定する方法さえ与えられれば同様に定義できる．連続性を考慮しあまり大幅な解変更でない拡張として，2 層 HV 配線における Flip を導入する．

配線規則は周知であるが改めて定義しておく：第 1 層は水平，第 2 層は垂直に限ってグリッドに沿って配線できる．両層の配線は同じグリッド点を通過しても交差しない．しかし必要ならば，任意のグリッド点で両層の配線を結ぶことができる（ビアを設定するという）．ここでは，端子は両層貫通型（どちらの層でも接続できる）とする．

2 層 HV 配線は，しばしば**図 4** 左上のように平面グリッド上で表現され，曖昧さなく読み取れる．これは規則：垂直に走る配線と水平に走る配線は，ビアが設定されているグリッド点において，そこだけで交わる，が仮定されているからである．我々もこの表現を使って，スタイナ木 $T$，枝 $e$ に対し，面に見える面サイクル $\overline{F}$ を採用する．任意にビアが設定できるので，この面サイクルの水平，垂直いかなる部分も連結な配線になりうる．したがって，枝 $e$ を除いて分離した $T - \{e\}$ を与えられた面サイクル $\overline{F}$ 上のパスで迂回させることができるので，ここでも Flip をそのように定義する．このとき，平面配線時の Flip と同様，「連続性」,「高速性」が保たれる．到達可能性（全列挙）についても同様に保たれる．

簡単な例を図 4 に示す．左上図の実線が着目するスタイナ木である．障害物として他のスタイナ木が示してある．それに接続している枝は取り除く．もし，障害物を取り除いたときグラフが 2 点連結でなくなるときは 2 点連結成分に分解して考える．平面グラフの場

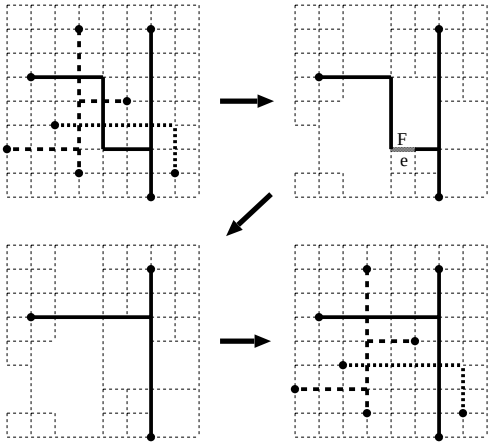



**図4** 2層HV配線に対するFlip
Fig. 4 Flip applied to the 2-layer HV routing.

合と異なるのは，端子ではないグリッド点で配線方向に直交しているグリッド枝は接続してはいない，と見なされることである．そのようなグリッド枝は残さなければならない．こうして右上図を得る．ここで枝 $e$ を決め，この表現においてそれを含む面サイクル $\overline{F}$ を決める．これに対し，Flipを行えば左下図を得る．最後に，待機させていたスタイナ木を復帰させる（右下図）とHV配線になっていることが認められる．

## 5. 実　　験

我々はFlipを以下のシミュレーテッドアニーリング（SA）のアルゴリズムで実装した．

algorithm **Self-reforming Routing**
既存のアルゴリズムにより初期解を生成する
初期解をそれまでで最も良い解として保持する
**for** ( $t = \underline{T_{init}}$ ; $t < \underline{T_{final}}$ ; $t = t * \underline{C}$) {
  **for** ( $i = 0; i < \underline{L}; i++$ ) {
    (1) Flipにより解を更新する
    (2) 解の評価を行い，その解の採用
        不採用を $t$ の関数で表される
        確率で決定する
    (3) それまでで最も良い解を保持
        する
  }
}

Flipの適用において，その引数であるネット，枝，面サイクルはすべて乱数により選ぶ．通常のSAと同じく，出力は繰り返し生成された解のうち最も良い解とする．しかし，収束解（改善が見られなくなったときの解）が最も良い解とされているので，本実験でもそのようにスケジューリング（下線部の変数の値の調整）を行った．

### 5.1 クロック配線の最適化

クロック配線においては配線長とスキューを最適化する．**図5** 左の図を初期クロック配線木とする．これは，ソースから始めて既成の部分木に近い点を順に結ぶPrimのアルゴリズムをもとにした迷路法を用いて生成したもので，スキューなどは考慮されていない．図の影のかかった14個の矩形は配線禁止領域である．

総配線長をLEN，ソースからシンクまで長さの最大値と最小値の差をSKEWとし，評価関数は (LEN) $+\alpha$(SKEW)$^2$ とする．$\alpha$ はパラメータで，値が小さいときは最小スタイナ木を，大きいときはゼロスキューのクロック木を構築しようとする．実験は3通りの $\alpha = 0$，0.25，1について行った．SAの各変数は $T_{init} = 100$，$T_{final} = 0.01$，$C = 0.98$，$L = 500$ とした．結果として得られたクロック木を図5に，その数値データを**表1** に示す．

**考　　察：**

評価関数に忠実なアルゴリズムであることが分かる．$\alpha = 1$ のときほぼゼロスキューが達成できた．

この実験のように端子や他のネットなどの障害物があるために均一でない配線領域には文献1)に見られるような修正を加えないクロック木構成手法では，適用することができない．文献10)の手法は障害物なども考慮できるが，いくつかの配線パターンを探索し，評価（この文献では配線長）の良いものを選ぶ手順を必要とする．我々の手法ではそのような計算が自動的に行われる．遅延の大きさを合せるために，周囲の隙間を探して勝手に迂回している様子が曲がりとして観測される．

### 5.2 2層HV配線の最適化

我々のアルゴリズムの平均改善割合を調べるために入力として様々な配線経路を生成した．ネット数が $n$ のときの入力を *class*-$n$ と呼ぶ．*class*-30，*class*-40，*class*-50 に対し，それぞれ10個ずつのデータを生成した．各ネットの端子数は2〜4の間で，それぞれの端子位置はランダムに決めた．

入力の配線経路はPrimのアルゴリズムをもとにした迷路法により生成し，明らかに不必要なビアは整形して除去した．この生成方法は，ネットの入力順に依存するが，ビアと配線長についてほぼ最適化されているといえる．

評価関数は $\alpha *$ (現在のVIA/初期解のVIA) $+ \beta *$ (現在のCTLK/初期解のCTLK) とする．ここで，

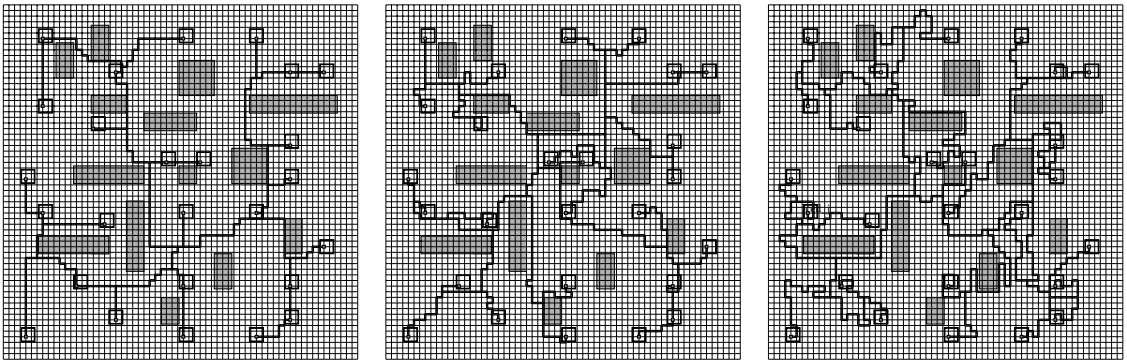


**図 5** 実験 1：入力（左）と出力（$\alpha = 0.25$（中），$\alpha = 1$（右））
Fig. 5 Exp. 1: input (left) and outputs ($\alpha = 0.25$ (center), $\alpha = 1$ (right)).

**表 1** 実験 1：クロック配線の最適化
Table 1 Exp. 1: Clock routing optimization.

| | LEN | SKEW | time (sec) |
|---|---|---|---|
| input tree | 1344 | 352 | − |
| $\alpha = 0$ | 1184 | 272 | 363 |
| $\alpha = 0.25$ | 1848 | 48 | 440 |
| $\alpha = 1$ | 2728 | 4 | 529 |

**表 2** 実験 2：2 層 HV 配線における改善割合の平均
Table 2 Exp. 2: Average improvement of 2-layer HV routing.

| | $\alpha = 0, \beta = 1$ | | $\alpha = 1, \beta = 9$ | | $\alpha = 1, \beta = 3$ | |
|---|---|---|---|---|---|---|
| class | VIA | CTLK | VIA | CTLK | VIA | CTLK |
| class-30 | 4.31 | 0.27 | 3.51 | 0.31 | 2.74 | 0.32 |
| class-40 | 3.89 | 0.34 | 3.30 | 0.33 | 2.61 | 0.35 |
| class-50 | 3.26 | 0.51 | 2.81 | 0.53 | 2.32 | 0.51 |

VIA はビアの総数，CTLK は異なるネットが格子グラフ上で 1 単位長分だけ離れて並列に配線されているとき，その部分の配線の長さの最大値とする．$\alpha$，$\beta$ はそれぞれ重み付けを表すパラメータで，これらの値を変えて最適化を行った．

1 つの工夫として，初期解で達成されている配線長が短いことを活かすために，配線長が初期解の配線長の 1.1 倍を超える配線は非許容解として受け付けないようにした．

実験はビア数とクロストークの重み付けを変えて 3 種類行った．SA の各変数は $T_{init} = 100$，$T_{final} = 0.1$，$C = 0.98$，$L = 20000$ とした．結果のデータを**表 2** に示す．表の数値は，class ごとにデータ 10 個について，(出力の評価値/入力の評価値) の平均をとった値である．その値は小さいほど配線は改善されている．**図 6** 左の入力に対する実験結果を図 6 中と右に示す．本実験で計算時間は 1200〜1800 秒を要した．

**考　察：**

平均的にどのクラスにおいてもクロストークは大幅に減少している．一方ビアは増加しているが，これは初期解の生成の際にビア数を最小化しているためである．前の実験と同様，重み付けの係数に忠実にビア数，クロストークとも改善割合は変化している．

## 6. おわりに

配線経路修正アルゴリズムを Flip に基づいて提案した．この手法の特徴は，解の変更が近傍変換なので高速に解を列挙でき，しかも，評価が連続的であることにある．したがって，たとえば SA による解空間の探索に適している．

配線領域および複合評価に関する簡単なモデルに対する実験では良い結果が得られることが分かった．

なお，本手法の Flip を再配線と見なせば，比較対象として再配線を基本操作とする rip-up and reroute [9]（以降，RR と記す）がある．RR では構成的アルゴリズムを必要とするが，評価関数がそれほど複雑でない場合にはその評価に基づく配線経路を高速に生成可能なので，RR を探索アルゴリズム化した方法も可能である．これは構成的アルゴリズムは困難であるとする本文の立場からは外れるが，興味深いと思われるので 1 つの試みを付録で紹介する．

**謝辞** 本研究を進めるにあたって，適切な助言を下さった東京工業大学高橋篤司助教授に感謝する．なお，本研究は CAD21 プロジェクトの一部である．

## 参考文献

1) Edahiro, M.: An Efficient Zero-Skew Routing Algorithm, *Proc. 31st DAC*, pp.375–380 (1994).
2) Boese, K.D., Kahng, A.B., McCoy, B.A. and

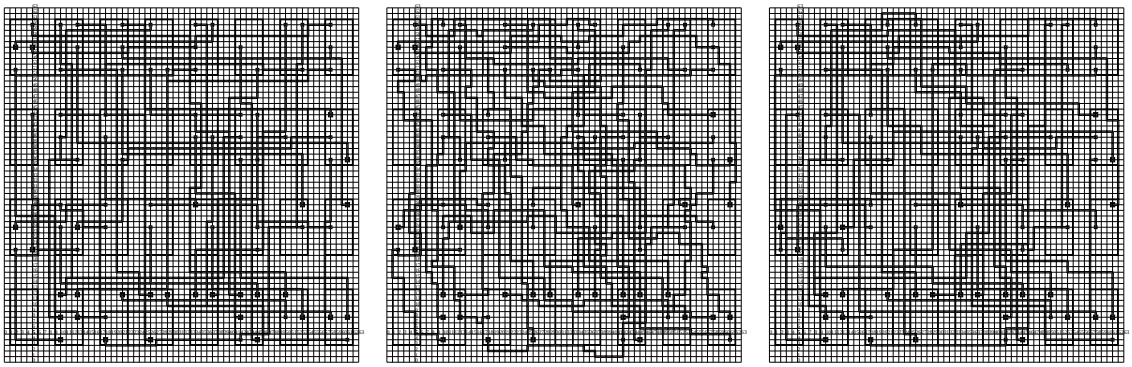


**図6** 実験2：入力（左）と出力（$\alpha = 1$, $\beta = 9$（中），$\alpha = 1$, $\beta = 3$（右））
Fig. 6 Exp. 2: input (left) and outputs ($\alpha = 1$, $\beta = 9$ (center), $\alpha = 1$, $\beta = 3$ (right)).

Robins, G.: Rectilinear Steiner Trees with Minimum Elmore Delay *Proc. 31st DAC*, pp.381–386 (1994).
3) Gao, T. and Liu, C.L.: Minimum Crosstalk Channel Routing, *IEEE Trans. CAD*, Vol.15, No.5, pp.465–474 (1996).
4) Vittal, A. and Sadowska, M.M.: Crosstalk Reduction for VLSI, *IEEE Trans. CAD*, Vol.16, No.3, pp.290–298 (1997).
5) Krauter, B. and Mehrotra, S.: Layout Based Frequency Dependent Inductance and Resistance Extraction On-Chip Interconnect Timing Analysis, *Proc. 35th DAC*, pp.303–308 (1998).
6) Lilis, J. and Buch, P.: Table-Lookup Methods for Improved Performance-Driven Routing, *Proc. 35th DAC*, pp.368–373 (1998).
7) Cong, J., Hei, L., Koh, C.-K. and Madden, P.H.: Performance optimization of VLSI interconnect layout, *INTEGRATION, the VLSI Journal*, Vol.21, pp.1–94 (1996).
8) Esbensen, H. and Kuh, E.S.: An MCM/IC Timing-Driven Placement Algorithm Featuring Explicit Design Space Exploration, *Proc. IEEE Multi-Chip Module Conference*, pp.170–175 (1996).
9) Shin, H. and Sangiovanni-V., A.: Mighty: A Rip-Up and Reroute Detailed Router, *Proc. ICCAD-86*, pp.2–5 (1986).
10) Tsay, R.-S.: Exact Zero Skew, *Proc. ICCAD-91*, pp.336–339 (1991).

## 付録 再配線法との比較

改善を図って再配線するという点でFlipと似ているともいえるrip-up and reroute（RR）について触れておく．

RRは複数ネットの100%配線を目指す技法である．これが起動するのは，ネットを順に完成する方式において，あるネットの完成を妨げる直接の原因となった別の既配線ネットが特定できるときである．そして，既配線を取り去り自分を配線する．

このようにRRはFlipと異なる概念であるが，次のような方式に発展させることは今後の研究テーマになるかもしれない：多数の評価の複合評価下での最適配線問題を解くのに，はじめのいくつかを考慮して配線を実現し，続いて残りの評価で整形する．

一例として，2層HV配線において，総配線長最小化で実現し，クロストークとVIA数の最小化を整形関数とする配線実験の試みを紹介する．

RRのアルゴリズムを現在解の評価改善手法に発展させるには，再配線するきっかけが必要である．我々の提案と比較するために再配線するスタイナ木を乱数で選ぶこととした．再配線できない場合に対する工夫はいろいろあるが，本実験ではFlipとの性能を比較するためそのような解は受け付けないものとした．

このRRの探索手法化に使う解の更新方法を定義する．

procedure **確率再配線**
**step 1:** 乱数により1ネットを選ぶ
**step 2:** 選ばれたネットを引き剥がす
**step 3:** Primのアルゴリズムを基にした迷路法を用いて再配線する．ただし，枝の重みは0.5とする．端点がビアになるときは0.1を，並走する枝に他のネットが存在するときは0.9をそれぞれ枝重みに加える．
end procedure

枝重みは配線長が大きくなり過ぎないように，ビア数，およびクロストークを増やさないための工夫で

**表 3** 実験 3：既存の手法と Flip の改善割合の比較
Table 3 Exp. 3: Comparison of performance of Flip and RR.

| | 評価関数 | | VIA | | CTLK | |
|---|---|---|---|---|---|---|
| class | Flip | RR | Flip | RR | Flip | RR |
| class-30 | 0.62 | 0.54 | 3.35 | 1.29 | 0.32 | 0.46 |
| class-40 | 0.64 | 0.81 | 2.98 | 1.30 | 0.38 | 0.75 |
| class-50 | 0.73 | 0.79 | 2.50 | 1.32 | 0.53 | 0.73 |

**表 4** 実験 3：既存の手法と Flip の改善割合の最大値と最小値
Table 4 Exp. 3: Comparison of performance.

| | Flip | | RR | |
|---|---|---|---|---|
| class | MIN | MAX | MIN | MAX |
| class-30 | 0.51 | 0.78 | 0.20 | 1.00 |
| class-40 | 0.56 | 0.71 | 0.46 | 1.00 |
| class-50 | 0.48 | 0.87 | 0.41 | 1.00 |

ある．

アルゴリズムは Self-reforming Routing を用い，RR の際には，(1) の操作を確率再配線による解の更新とする．入力となる配線経路の生成方法，使用するデータ，制約，SA の各変数は 5.2 節の実験と同じである．ただし，この実験では RR と Flip が同時間（1800 秒）でどの程度改善が図れるか調べるため，その時間で終了するように SA の変数 $L$ を調整した．また，評価関数は $1*$(現在の VIA/初期解の VIA)$+9*$(現在の CTLK/初期解の CTLK) とする．

結果のデータを**表 3** に示す．表の数値は評価関数，VIA，CTLK それぞれについて，class ごとにデータ 10 個について，(出力の評価値/入力の評価値) の平均をとった値である．**表 4** は評価関数のみについて，class ごとにデータ 10 個の (出力の評価値/入力の評価値) の最小値（MIN）と最大値（MAX）を示したものである．この表の値が 1 のときまったく解の改善が見られないことを意味し，また，値が小さいほど解が改善されていることを示している．

**考　　察：**

表 3 より，評価関数に関する改善の割合は Flip の方が若干良いが同等であるといえよう．

ところが，表 4 から明らかなように，RR の方が非常に値がばらついている．これは RR の 1 回の操作のコストの変化が Flip と比べ，非常に大きいためである．このため，解が改善されるときは 1 回の操作でコストが大きく減少して，そのあと変化しない．よって，短時間である程度良い解を出力するという点では RR の方が優れているといえるが，SA で解を探索するのには向いていない．

また，表 4 の RR の MAX が 1 であることから，RR にはまったく改善の見られないデータも含まれていることが分かる．一方 Flip はどのデータに関しても多少の改善は見られている．このことから，実験で採用したデータに関しては，我々の手法は RR では改善しにくいデータに対しても微修正ができるといえる．

(平成 11 年 9 月 20 日受付)
(平成 12 年 2 月 4 日採録)

**久保ゆき子**

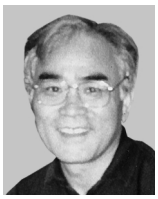
昭和 49 年生．平成 10 年東京工業大学工学部情報工学科卒業．現在，同大学大学院修士課程在学中．VLSI 設計に関する研究に従事．

**高島　康裕**

昭和 44 年生．工学博士．平成 5 年東京工業大学工学部電気電子工学科卒業．平成 7 年同大学大学院修士課程修了．平成 10 年同大学院博士課程修了．現在，北陸先端科学技術大学院大学助手．VLSI 設計に関する研究に従事．

**中武　繁寿**

昭和 44 年生．工学博士．平成 5 年東京工業大学工学部電気電子工学科卒業．平成 7 年北陸先端科学技術大学院大学博士前期課程修了．平成 9〜11 年東京工業大学工学部電気電子工学科助手．平成 11 年東京工業大学工学部電気電子工学科博士取得．現在，北九州大学国際環境工学部設置準備室講師．VLSI 設計に関する研究に従事．



**梶谷　洋司**

昭和 16 年生．工学博士．昭和 44 年東京工業大学大学院博士課程修了．東京工業大学助手．同助教授を経て現在，東京工業大学教授．平成 3〜8 年北陸先端科学技術大学院大学教授（東工大と併任）．グラフ理論，組合せ理論の応用，特に VLSI 設計のための計算機援用の研究に従事．平成 4 年 IEEE Fellow.