計算機アーキテクチャ　115－26
設　計　自　動　化　78－26
（１９９５．　１２．　１５）

配線可能性検証のための容量判定グラフの提案

川口　泰　　磯　直行　　平田　富夫

名古屋大学工学部
〒464-01 名古屋市千種区不老町
E-mail: yasushi@hirata.nuee.nagoya-u.ac.jp

ＶＬＳＩやプリント配線板設計において，配線経路決定を概略配線と詳細配線に分割して行なう手法が提案されている．概略配線から詳細配線への変換可能性の検証を行なう問題を配線可能性問題という．この検証は配線領域内のカットと呼ばれる線分について，カット上を通過することのできる配線本数（容量）と概略配線でカット上を通過している配線本数（フロー）の比較により行なうことができる．主な実行時間は概略配線からフローを求める時間と容量とフローの比較を行なう時間である．本論文では容量とフローの比較を効率良く行なうための容量判定グラフを提案する．また概略配線からフローを効率良く求めるためのフロー導出グラフを提案する．

Two Graphs for Efficient Routability Checking

YASUSHI KAWAGUCHI, NAOYUKI ISO and TOMIO HIRATA

Faculty of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya
464-01 Japan

In VLSI and printed wiring board design, a routing method consisting of two stages: global routing and detailed routing has been proposed. In such a method, it is necessary to decide the routability, that is, to decide whether the obtained global routing sketch can be transformed into a detailed routing or not. We can decide it by comparing "capacity" and "flow" for each "cut". In this paper, we propose two graphs for efficient routability checking.

1. はじめに

現在，VLSI やプリント基板の配線経路探索問題においては迷路法や線分探索法などの逐次配線手法が用いられている．これらの手法の問題点は，既配線が障害物として扱われるため経路発見の妨げになることである．この問題を解決するために，経路のトポロジのみを考えて概略配線を行なう手法が提案されている．この手法では，全ての経路について概略配線を行った後に，その概略配線が詳細配線に変換可能かどうかを検証し，可能な場合には詳細配線を出力して終了し，不可能な場合には概略配線を変更するという作業を繰り返す．この配線可能性検証は概略配線を変更する度に行なわれるため高速性が要求される．

配線可能性検証は，配線領域内のカット (配線領域内の互いに可視な 2 点を結ぶ線分) について，カット上を通過することのできる配線本数 (容量) とカット上を通過している概略配線の本数 (フロー) の比較により行なうことができる．その実行時間は比較の回数とフローを求める時間に比例する．配線領域内のオブジェクト (端子点，モジュール等) の数を n とするとき，$O(n^2)$ 本のカットについて容量とフローを比較することにより配線可能性の検証ができることが証明されている[1]．また，$O(n^2)$ 本ではあるが [1] より少ない数のカットについて容量とフローを比較することで，配線可能性の検証ができることが [2] で示されている．[2] ではさらに $O(n^2 \log n)$ の実行時間で比較が必要なカットについて概略配線からフローを求めるアルゴリズムを与えている．

本論文では端子点のみの配線モデルを考え，配線可能性検証を行なうために容量判定グラフとフロー導出グラフという 2 つのグラフを提案する．このグラフを利用して，平均で $O(n \log n)$ 本のカットについて容量とフローを比較することにより配線可能性が検証できることを示す．また，ある特定の $O(n)$ 本のカットについてフローを与えれば，$O(n \log n)$ の平均実行時間で必要なフローをすべて算出できることを示す．

以下に，2 節で配線モデルと言葉の定義を行ない，3 節では容量判定グラフの定義とその性質について述べる．4 節ではフロー導出グラフの定義とフローを求めるアルゴリズムについて述べ，5 節では配線アルゴリズムを示す．6 節でまとめを行なう．

2. 諸定義

本論文の配線モデルを述べる．配線領域は単層であり縦横格子がひかれている．領域内には端子点と，端子点

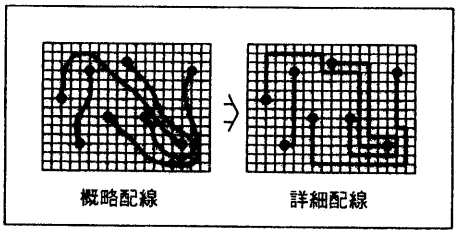

図 1　配線モデル

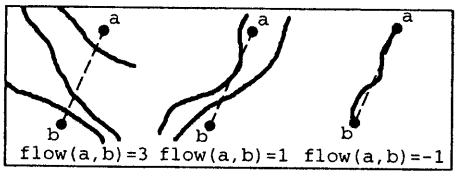

図 2　フロー

間を結ぶ互いに交わらない配線経路がある．端子点は格子点にのみ存在する．概略配線では配線はトポロジで表現され，詳細配線では全ての経路は格子上に乗る (図 1)．以下では，点は全て格子上にあるものとする．

次にカット，容量及びフローを定義する．

定義 1　2 つの端子点間を結ぶ線分をカットと呼ぶ．配線領域内の 2 端子点 $p = (x_p, y_p)$，$q = (x_q, y_q)$ について，カット (p, q) を通過することのできる配線の最大数を容量 ($cap(p, q)$ と表記する) と呼び，$cap(p, q) = max(|x_p - x_q|, |y_p - y_q|) - 1$ と定義する．概略配線が与えられた時，カット (p, q) を通過している概略配線経路の本数をフローと呼び，$flow(p, q)$ と表記する．ただし，トポロジ的に交差していない配線は 0 本，重なる配線は -1 本と数える (図 2)．$cap(p, q) \geq flow(p, q)$ となる条件をカット (p, q) に対する容量制約と呼ぶ．

定理 1[1]　概略配線が与えられた時，すべてのカットに対して容量制約がみたされる場合，そしてその場合のみ概略配線より詳細配線への変換が可能である．

[1] では配線モデルを端子点，矩形モジュール，トポロジで表現された配線経路の集合とし，異なるオブジェクトの外周にあり互いに可視な 2 つの格子間を結ぶ線分をカットと定義している．そしてすべてのカットについて容量制約がみたされるとき，全ての配線経路を交差することなく格子に乗せることが可能であることを示している．本論文では概略配線のモデルにおいて矩形モジュールを考えていないため，任意の 2 端子対は可視であり，上の定理 1 が成立する．

a と b を配線領域内の 2 点とする．a または b を通る 45°および $-45°$ の 4 本の直線によって作られる長方形を，a と b により作られる四角形と呼び，$rect(a, b)$ と表

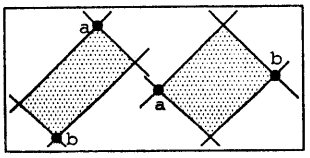

図3　rect (a,b)

記する (図3).

また，以下の言葉の定義を行なう．

点 $a=(x_a,y_a)$ と，a を通る傾きが 45°および−45°の直線を考える．2つの直線に対して共に上にある領域を a の上の領域と呼ぶ．同様に a の右の領域，下の領域，左の領域を定める．また，座標軸を−45°回転した時の a の座標を $(\mu_a,\nu_a)=(\frac{x_a-y_a}{\sqrt{2}},\frac{x_a+y_a}{\sqrt{2}})$ とする．点 $a=(\mu_a,\nu_a)$, $b=(\mu_b,\nu_b)$ について，$\mu_a>\mu_b$ ならば，−45°軸に関して a は b より大きいと言う．また，$\nu_a>\nu_b$ ならば，45°軸に関して a は b より大きいと言う．

3. 容量判定グラフ

定理1によれば，配線可能性の検証のためにはすべてのカットに対する容量制約のチェックが必要となるが，実際にはある特定のカットについてのみ行なえばよい．例として内部に他の点が存在しない3角形 abc を考える．カット (a,b) の容量がカット (a,c) と (b,c) の容量の和より大きいとき，(a,c) と (b,c) から3角形内に入る配線が全て (a,b) から出ていく場合にも容量違反は起こらない．すなわち (a,c) と (b,c) について容量制約がみたされるなら，(a,b) についての容量制約のチェックは冗長である．この節では，冗長なカットを取り除いたカット集合を辺とする容量判定グラフを定義する．

3.1 容量判定グラフの定義と諸定理

容量判定グラフ $\mathbf{G}_c=(\mathbf{V},\mathbf{E}_c)$ は，配線領域内の端子点集合を $\mathbf{V}$とし，以下の条件を満たす $\mathbf{E}_c$を辺集合とするグラフである．

$(a,b)\in\mathbf{E}_c$ iff $cap(a,b)=cap(a,c)+cap(b,c)+1$ をみたす点 $c(\in\mathbf{V})$ が存在しない．

$a=(x_a,y_a)$, $b=(x_b,y_b)$ を配線領域内の2点とする．このとき以下の補題が成り立つ．

補題1　a,b と異なる点 $c=(x_c,y_c)$ について，$cap(a,b)=cap(a,c)+cap(b,c)+1$ が満たされる時，そしてその時のみ，点 c は $rect(a,b)$ の内部に存在する．

証明　まず，逆方向を示す．$cap(a,b)=|x_a-x_b|-1$ の場合について考える．点 c の座標 (x_c,y_c) とし，これが $rect(a,b)$ の内部に入る時，$cap(a,c)=|x_a-x_c|-1$，かつ $cap(b,c)=|x_b-x_c|-1$ である．しかも $|x_a-x_c|+|x_b-x_c|=|x_a-x_b|$ である．よって $cap(a,b)=cap(a,c)+cap(b,c)+1$ が成り立つ．$cap(a,b)=|y_a-y_b|-1$ の場合も同様である．次に順方向を示す．$cap(a,b)=|x_a-x_b|-1$ の場合について考える．このとき $cap(a,b)=|x_a-x_b|-1\le|x_a-x_c|+|x_b-x_c|-1=|x_a-x_c|-1+|x_b-x_c|-1+1\le cap(a,c)+cap(b,c)+1$ である．$cap(a,b)=cap(a,c)+cap(b,c)+1$ なので $|x_a-x_b|=|x_a-x_c|+|x_b-x_c|$ かつ $|x_a-x_c|\ge|y_a-y_c|$ かつ $|x_b-x_c|\ge|y_a-y_c|$ である．この範囲は $rect(a,b)$ の範囲と等しい．$cap(a,b)=|y_a-y_b|$ の時についても同様である．■

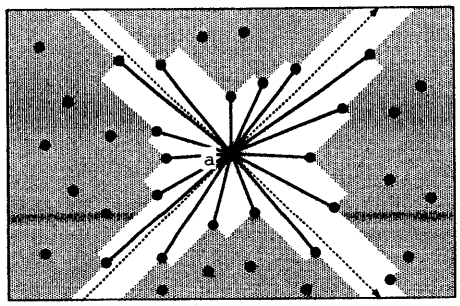

図4　点 a を一方の端点とする容量判定グラフの辺

図4に点 a を一方の端点とする容量判定グラフの辺の例を示す．空白の部分は点が存在しない領域である．図中に示した辺はすべて，両端点により作られる四角形内に他の点が存在しない辺であることがわかる．

補題2　$rect(a,b)$ の内部に a と b 以外の点が存在するとき，$rect(a,b)$ の内部の任意の2点 $p,q((p,q)\neq(a,b))$ について $cap(p,q)\ge flow(p,q)$ ならば，$cap(a,b)\ge flow(a,b)$ である．

証明　点 $c(\neq a,b)$ は $rect(a,b)$ 内に存在し，$\triangle abc$ 内に他の点が存在しないとする．辺 (a,c), (b,c) からそれぞれ $flow(a,c)$ 本，$flow(b,c)$ 本の配線が三角形内に入り，点 c からは高々1本の配線が発生している．これらの配線がすべて辺 (a,b) から出ていくときに $flow(a,b)$ は最大であり，$flow(a,b)\le flow(a,c)+flow(b,c)+1$ である．仮定より $cap(a,c)\ge flow(a,c)$ かつ $cap(b,c)\ge flow(b,c)$ なので，$cap(a,b)=cap(a,c)+cap(b,c)+1\ge flow(a,c)+flow(b,c)+1\ge flow(a,b)$ となり，$cap(a,b)\ge flow(a,b)$ が成り立つ．■

定理2　$\mathbf{G}_c=(\mathbf{V},\mathbf{E}_c)$ を容量判定グラフ，$\mathbf{G}^*=(\mathbf{V},\mathbf{E}^*)$ を完全グラフとする．このとき，$\mathbf{G}_c$の任意の辺 (a,b) について $cap(a,b)\ge flow(a,b)$ ならば，$\mathbf{G}^*$の任意の辺 (p,q) についても $cap(p,q)\ge flow(p,q)$ である．

証明　$\mathbf{G}^*$の辺の，長さに関する帰納法により証明を行

なう．$\mathbf{G}^*$に含まれる辺を長さでソートし，短いものから$e_1, e_2, ...$とする．e_1の端点をc，dとしたとき$rect(c,d)$内には他の端子点は存在しない．もし点eが$rect(c,d)$内に存在するなら(辺(c,d)の長さ)$\geq$(辺(c,e)の長さ)となりe_1が一番短い辺であることと矛盾するからである．補題1よりe_1は$\mathbf{G}_c$に含まれるので仮定より$cap(c,d) \geq flow(c,d)$である．$i \leq k$である任意のiについて，$cap(e_i) \geq flow(e_i)$が成り立っているとする．e_{k+1}の端点をc，dとすると，$rect(c,d)$内に他の点が存在しない場合は，補題1よりe_{k+1}は$\mathbf{G}_c$に含まれるので，仮定より$cap(c,d) \geq flow(c,d)$である．$rect(c,d)$内に他の点が存在する場合，両端点が$rect(c,d)$の内部に存在する任意の辺(e,f)は，e_{k+1}より長さが短いため$cap(e,f) \geq flow(e,f)$であり，補題2より$cap(c,d) \geq flow(c,d)$である．よって，任意の辺について$cap \geq flow$である．■

定理1と定理2より，容量判定グラフに含まれる辺についてのみ容量制約のチェックを行えば，概略配線から詳細配線への変換可能性の検証が出来ることがわかる．

なお，容量判定グラフの辺の両端点は以下に示すように直接支配(direct dominance)の関係にある．

定義2 異なる2点$p=(x_p,y_p)$，$q=(x_q,y_q)$に対し，$x_p \geq x_q$かつ$y_p \geq y_q$であるならpはqを支配するといい，$p \succ q$と表記する．また，点集合$\mathbf{V}$が与えらたとき，2点$p,q(\in \mathbf{V})$について，$p \succ q$であり，かつ$p \succ r \succ q$である点$r(\in \mathbf{V})$が存在しない時pは$\mathbf{V}$の中でqを直接支配するという．また，このような(p,q)の集合を直接支配集合と呼ぶ．

n個の点がランダムに分布している時，直接支配集合の要素数は$O(n \log n)$であることが証明されている[5]．また，直接支配集合を$O(n \log n + e)$(eは直接支配集合の要素数)で求めるアルゴリズムが与えられている[6]．容量判定グラフの辺は，座標軸を$-45°$，$45°$回転させた場合の直接支配関係にある点の対である．よって次の定理が成立する．

定理3 容量判定グラフは$O(|\mathbf{V}| \log |\mathbf{V}| + |\mathbf{E}_c|)$時間で作成できる．また，端子点が配線領域内にランダムに分布しているとき，$|\mathbf{E}_c|$の期待値は$O(|\mathbf{V}| \log |\mathbf{V}|)$である．

3.2 容量判定グラフのデータ構造

容量判定グラフを保持するデータ構造を図5に示す．1次元配列に$\mathbf{V}$の点が入る．配列の各点には2つのリストがリンクされ，それぞれ容量判定グラフに含まれる辺のなかで上方向及び右方向に伸びる辺のもう片方の端点が入る．配列の中で各点は$45°$軸に関して大きい順にソートされている．また，リストの中の点は$45°$軸に関して小さい順にソートされている．

このデータ構造の中では，各点から右上方向に伸びる辺についてのデータは保持しているが，左及び下方向に伸びる辺についてのデータは保持していない．これは，ある点から見て左方向に伸びる辺はもう一方の端点から見ると右方向に伸びる辺となっているからである．

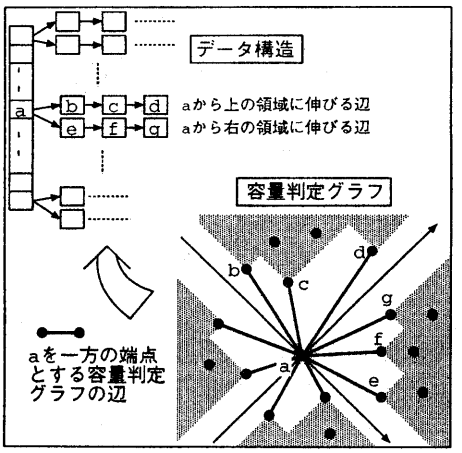

図5 データ構造

4. フロー導出グラフ

配線可能性検証のためには，容量判定グラフの辺について容量とフローが求められている必要がある．この節ではフローを効率的に求めるために，フロー導出グラフを提案する．このグラフの辺についてのみフローがわかっているならば，フリップという操作を行なうことにより，容量判定グラフのすべての辺について1本あたり$O(1)$時間でフローを求めることができる．

4.1 フリップ操作による flow の算出

$\square v_1v_2v_3v_4(v_i \in V)$を凸四角形とする．$flow(v_1,v_2)=\alpha$，$flow(v_2,v_3)=\beta$，$flow(v_3,v_4)=\gamma$，$flow(v_4,v_1)=\delta$，$flow(v_1,v_3)=\epsilon$のとき$flow(v_2,v_4)$は次式で求められる．

$$flow(v_2,v_4)=\frac{|\alpha-\beta+\gamma-\delta|}{2}+\frac{\alpha+\beta+\gamma+\delta-2\epsilon}{2}$$

凸四角形について，4つの外周辺と1つの対角線のフローが与えられた時，それらを基にもう1つの対角線についてのフローを求める操作をフリップと呼ぶことにする．多角形の内部領域が3角分割され，各枝の$flow$が与えられた時，フリップを繰り返しながら任意の対角線についての$flow$を求めることが可能である．図6の例では，$\square ABDC$についてフリップを行ない$flow(A,D)$を求め，$\square ADEC$についてフリップを行ない$flow(A,E)$を求めている．1度のフリップにかかる計算時間は$O(1)$である．

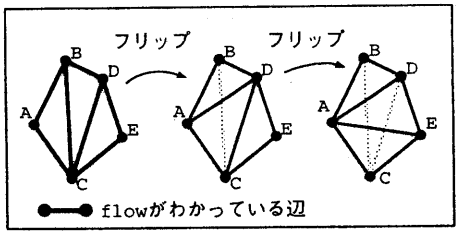

図6　フリップ操作

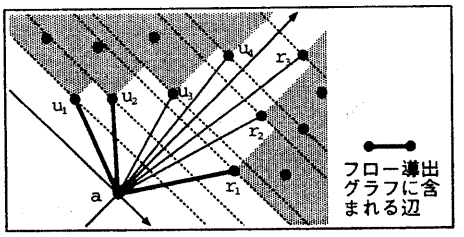

図7　フロー導出グラフの辺の例

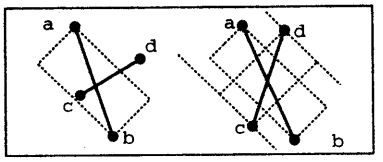

図8　互いに交差する辺の位置関係

4.2 フロー導出グラフの定義と諸定理

フロー導出グラフ $\mathbf{G}_f = (\mathbf{V}, \mathbf{E}_f)$ は，配線領域内の端子点の集合を $\mathbf{V}$とし，以下の条件をみたす $\mathbf{E}_f$を辺集合とするグラフである．

$(a,b) \in \mathbf{E}_f$　iff　(1) $rect(a,b)$ 内に他の点が存在しない　かつ　(2) a の上かつ b の左の領域に他の点が存在しない，または，a の右かつ b の下の領域に他の点が存在しない．

このグラフについて以下の定理が成り立つ．

定理4　容量判定グラフ $\mathbf{G}_c = (\mathbf{V}, \mathbf{E}_c)$ が与えられたとき，フロー導出グラフ $\mathbf{G}_f = (\mathbf{V}, \mathbf{E}_f)$ は $O(|\mathbf{E}_f|)$ 時間で作成できる．

証明　定義により，$\mathbf{G}_f$の任意の辺は $\mathbf{G}_c$に含まれている．$\mathbf{G}_c$に含まれる辺の中で，点 a から上の領域に伸びる辺の (a 以外の) 端点を 45°軸に関して小さい順に並べたものを $u_1, u_2, ..., u_\alpha$とする．同様に，右の領域に伸びる辺の (a 以外の) 端点を 45°軸に関して小さい順に並べたものを $r_1, r_2, .., r_\beta$とする．このとき a の上かつ u_1の左の領域には他の点は存在しないので辺 (a, u_1) は必ず $\mathbf{G}_f$に含まれる．また，他の辺 $(a, u_i)(i \neq 1)$ については，r_1との 45°軸に関する大きさの比較により $\mathbf{G}_f$に含まれるか否かの判定が出来る．つまり，もし$\nu_{u_i} < \nu_{r_1}$ならば，a の左かつ u_iの下の領域に他の点が存在しないので，辺 (a, u_i) は $\mathbf{G}_f$に含まれる．もし$\nu_{u_i} > \nu_{r_1}$ならば，a の右かつ u_iの下の領域には必ず r_1が存在し，a の上かつ u_iの左の領域には必ず u_{i-1}が存在するので，辺 (a, u_i) は $\mathbf{G}_f$に含まれない．また，r_1が存在しない場合は，a の右かつ u_iの下の領域に他の点が存在しないので，辺 $(a, u_i)(1 \leq i \leq \alpha)$ は必ず $\mathbf{G}_f$に含まれる．任意の辺 (a, r_i) についても同様に $\mathbf{G}_f$に含まれるかの判別ができる．よって，a から上方向に伸びる辺については，3 節で示した容量判定グラフのデータ構造上で，$u_1, u_2, ..., u_\alpha$の順にたどることにより見つけることができる．もし$\nu_{u_i} > \nu_{r_1}$であるようなν_{u_i}が見つかったなら，そこでリストの探索を打ち切る．a から右方向に伸びる辺についても同様である．すべての点に対してこの操作を行なうことで，フロー導出グラフに含まれる辺を見つけることができる．チェックに必要な時間は $O(|\mathbf{E}_f|)$ である．■

定理5　フロー導出グラフ $\mathbf{G}_f = (\mathbf{V}, \mathbf{E}_f)$ は平面グラフであり，辺の数は $O(|\mathbf{V}|)$ である．

証明　互いに交差するカット (a,b), (c,d) について考える．図8の左の例のように $rect(a,b)$ 内に点 c が存在する場合は，定義より辺 (a,b) は $\mathbf{G}_f$には含まれない．図8の右の例のように $rect(a,b)$, $rect(c,d)$ 内に互いの端点が存在しない場合は，a は c の上かつ d の左の領域に存在する．また b は c の右かつ d の下の領域に存在する．よって辺 (c,d) は $\mathbf{G}_f$に含まれない．a, b, c, d の位置関係がかわっても，互いに交差する辺のうち，必ず一方は $\mathbf{G}_f$に含まれない．よってフロー導出グラフは平面グラフであり，辺の数は $O(|\mathbf{V}|)$ である．■

定理6　容量判定グラフ $\mathbf{G}_c = (\mathbf{V}, \mathbf{E}_c)$ 及びフロー導出グラフ $\mathbf{G}_f = (\mathbf{V}, \mathbf{E}_f)$ が与えられたとする．$\mathbf{G}_f$の任意の辺 (p,q) について $flow(p,q)$ がわかっている時，$O(|\mathbf{E}_c|)$ の実行時間で $\mathbf{G}_c$の全ての辺 (a,b) について $flow(a,b)$ を求めることができる．

証明　アルゴリズムの概略を示す．

配線領域に−45°軸に平行にスキャンラインを右上方向から左下方向に走査する．今，スキャンラインが点 a で止まったとする．$\mathbf{G}_c$に含まれる辺のうち，両端点がスキャンラインより右または上の領域にあるものについては $flow$ が求められているとする．$\mathbf{G}_c$に含まれる辺の中で，点 a から上の領域に伸びる辺の (a 以外の) 端点を 45°軸に関して小さい順に並べたものを $u_1, u_2, ..., u_\alpha$とする．同様に，右の領域に伸びる辺の a 以外の端点を 45°軸に関して小さい順に並べたものを $r_1, r_2, ..., r_\beta$とする．このとき容量判定グラフの性質より次のことが言える．任意

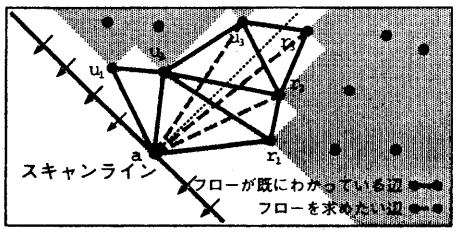

図9　フローを求めたい辺と，既に求められている辺

のi,jについて辺(u_i,u_{i+1})，辺(r_i,r_{i+1})は$\mathbf{G}_c$に含まれる．$\nu_{r_i} \leq \nu_{u_j} \leq \nu_{r_{i+1}}$または$\nu_{u_j} \leq \nu_{r_i} \leq \nu_{u_{j+1}}$のとき辺$(r_i,u_j)$は$\mathbf{G}_c$に含まれる．$\nu_{u_\alpha} \leq \nu_{r_i}$のとき，辺$(r_j,u_\alpha)$は$\mathbf{G}_c$に含まれる．また，$\nu_{r_\beta} \leq \nu_{u_i}$のとき，辺$(u_i,r_\beta)$は$\mathbf{G}_c$に含まれる．点$a$から右及び上の領域を見たときの例を図9に示す．辺$(a,u_1)$，$(a,u_2)$，$(a,r_1)$は$\mathbf{G}_f$に含まれる辺であり$flow$はすでに求められている．辺$(u_2,r_1)$，$(u_1,u_2)$，$(u_2,r_2)$，$(r_1,r_2)$，$(r_2,u_3)$，$(u_2,u_3)$，$(u_3,r_3)$，$(r_2,r_3)$は$\mathbf{G}_c$に含まれる辺であり，両端点がスキャンラインより右または上の領域のあるので$flow$はすでに求められている．したがって，多角形$ar_1r_2....r_\beta u_\alpha....u_2u_1$の内部領域は，$flow$がわかっている辺により3角分割されている．よってフリップを繰り返すことにより点aから右及び上方向に伸びる全ての辺について$flow$を求めることができる．一度のフリップに必要な実行時間は$O(1)$であるので，辺1本あたり$O(1)$の時間でフローを求めることが可能である．

このアルゴリズムはすべての点を走査し終えた時に終了し，このとき容量判定グラフのすべての辺について$flow$が求められている．実行時間は$O(|\mathbf{E}_c|)$である．■

5. 概略配線処理手順

図10に本論文で提案したグラフを用いた概略配線処理手順を紹介する．端子点の個数をnとすると，容量判定グラフの作成に必要な時間は$O(n\log n+|\mathbf{E}_c|)$，フロー導出グラフの作成に必要な時間は$O(n)$である．またフロー算出にかかる時間は$O(|\mathbf{E}_c|)$，容量制約のチェックのために必要な時間もまた$O(|\mathbf{E}_c|)$である．端子点がランダムに分布している場合には$|\mathbf{E}_c|$は$O(n\log n)$であり，すべての操作は$O(n\log n)$時間で実行できる．

この配線処理手順では，フロー導出グラフの辺にたいする$flow$を出力とする概略配線手法を考えなくてはならない．これについては，配線領域をフロー導出グラフの辺が分割辺となるように3角形分割し，各辺の配線交差数(フロー)を出力する手法が考えられる．また，既存の概略配線手法には，配線領域をDelaunay3角形分割して

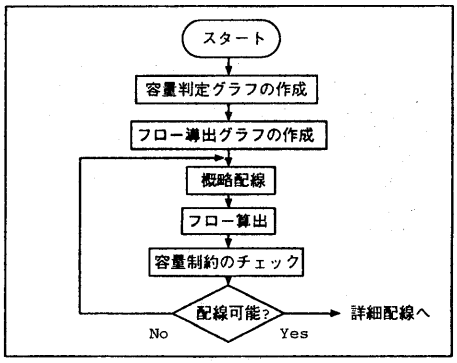

図10　配線処理手順

経路探索を行なうものがあるが[3,4]，3角形分割された領域に対してフリップを行なうことにより，フロー導出グラフを分割辺として含む3角分割に変形させるという手法も考えられる．

6. ま と め

配線可能性検証に有用な容量判定グラフとフロー導出グラフを提案し，その性質について述べた．

今回扱った配線モデルは，オブジェクトを端子点のみとし，配線経路は縦横格子に乗るという非常に限定されたものである．今後の課題としては，配線モデルをより現実のものに近付けることがある．

謝辞　本研究に関し，貴重な御助言を賜わりました中京大学の伊藤誠教授ならびに東海大学の譚学厚博士に感謝致します．

参 考 文 献

[1] R.Cole and A.Siegel: "River routing every which way, but loose," Proceeding of 25th annual Symposium on Foundations of Computer Science, pp.65–73 (1984).

[2] C.Leiserson and F.Maley: "Algorithms for Routing and Testing Routability of Planar VLSI Layouts," Proceeding of the 17th Annual ACM Symposium on Theory of Computing, pp.69–78 (1985).

[3] 田中博, 金沢正博, 田中秀彦, 佐藤政生, 大附辰夫: "スケッチ表現に基づく多層配線システム," 情報処理学会研究報告 DA 70-9, pp.63–70 (1994).

[4] 磯直行, 平田富夫, 伊藤誠: "詳細配置問題を組み込んだ配線モデル," 情報処理学会研究報告 DA 72-8, pp.43–48 (1995).

[5] Rolf Klein: "Direct Dominance of Points," Intern. J. Computer Math., Vol.19, pp.225–244 (1986).

[6] Ralf-Hartmut Guting, Otto Nurmi and Thomas Ottmann: "Fast Algorithms for Direct Enclosures and Direct Dominances," Journal of Algorithms 10, pp.170–186 (1989).