論文

配線可能性検証のための容量判定グラフとフロー導出グラフ

川口　泰[†]　磯　直行[†]　平田　富夫[†]

Two Graphs for Efficient Routability Checking

Yasushi KAWAGUCHI[†], Naoyuki ISO[†], and Tomio HIRATA[†]

あらまし　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う時間である．本論文では効率の良い配線可能性検証のために容量判定グラフの提案を行い，このグラフに含まれる辺についてのみカットとフローの比較を行うことにより配線可能性の検証ができることを示す．また概略配線からフローを求めるためのフロー導出グラフを提案し，このグラフの辺のフローが与えられたならば，容量判定グラフのすべての辺のフローを計算により求めることができることを示す．

キーワード　レイアウト設計, 概略配線, 配線可能性, 直接支配集合

1. まえがき

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，経路のトポロジーのみを考えて概略配線を行う手法が提案されている [2], [4], [8], [9]．文献 [2], [4], [9] の手法は，すべ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) の比較により行うことができる．その実行時間はフローを求める時間と比較の回数に比例する．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], [7]．文献 [1] では $O(n \log n)$ の実行時間で配線可能性検証を行うアルゴリズムも提案されているが，複雑で実用化するのは困難である．文献 [7] で与えられるアルゴリズムは比較的簡単であるが，$O(n^2 \log n)$ の時間を要する．

本論文では端子点のみの配線モデルを考え，配線可能性検証を効率良く行うために容量判定グラフとフロー導出グラフという二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. で配線モデルと言葉の定義を行い，3. 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. 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. では配線アルゴリズムを示す．6. でまとめを行う．

[†] 名古屋大学工学部，名古屋市
Faculty of Engineering, Nagoya University, Nagoya-shi, 464-01 Japan

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引かれている．領域内には端子点と，端子点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．配線経路は2端子点間を結ぶもののみとし，従って各端子点はちょうど1本の配線経路につながっている．概略配線では配線はトポロジーで表現され，詳細配線ではすべての経路は格子上に乗る(図1)．以下では，点はすべて格子点とする．次にカット，容量およびフローを定義する．

[定義1] 二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とき，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．但し，トポロジー的に交差していない配線は0本，カットと重なる配線は-1本と数える (図2) (-1本と定義するのは4. で示す補題3を成り立たせるためである)．$cap(p,q) \geqq flow(p,q)$ となる条件をカット (p,q) に対する容量制約という．

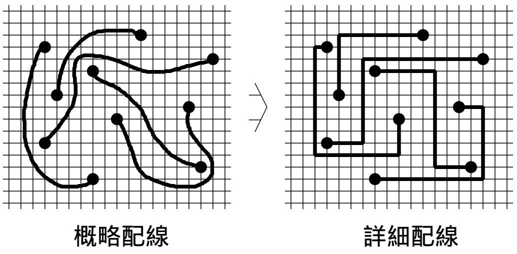

図1 配線モデル
Fig. 1 Routing model.

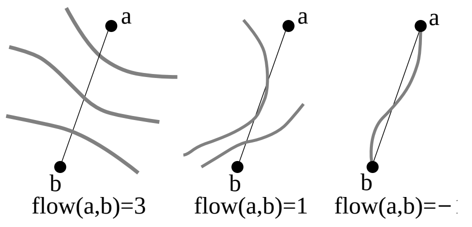

図2 a,b 間のフロー
Fig. 2 flow(a,b).

このとき文献[1]より次の定理が成り立つ．

[定理1] 概略配線が与えられたとする．すべてのカットに対して容量制約が満たされるとき，そしてそのときのみ概略配線より詳細配線への変換が可能である．

文献[1]では配線モデルを端子点，方形モジュール，トポロジーで表現された配線経路の集合としている．そしてカット(注1)を，異なるオブジェクトの外周にあり，かつ互いに可視な二つの格子点間を結ぶ線分と定義している．このときすべてのカットについて容量違反が起きないならばすべ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方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，上の定理が成立する．

配線領域内の点 a を通る傾きが45°および−45°の直線をそれぞれ l_a^+，l_a^- と記す．b が l_a^+ または l_a^- 上に存在しないとき，l_a^+，l_a^-，l_b^+，l_b^- の4本の直線によって作られる四角形を，a と b により作られる四角形と呼び，$rect(a,b)$ と表記する．

点 $a=(x_a,y_a)$ と，2本の直線 l_a^+，l_a^- を考える．二つの直線に対してともに上にある領域を 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°回転したときの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より大きいという．また，$\nu_a>\nu_b$ ならば，45°軸に関してaはbより大きいという．

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えばよい．例として内部に他の点が存在しない三角形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) から三角形内に入る配線がすべて (a,b) から出ていく場合で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満たされるなら，(a,b) についての容量制約のチェックは冗長である．この章では，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．

(注1)：カットというのは文献[7]で初めて用いられた言葉であり，文献[1]中ではtest-segmentという言葉を用いている．

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 $cap(a, b) = cap(a, c) + cap(b, c) + 1$ を満たす点 $c(\in \mathbf{V})$ が存在しない．

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

［補題 1］ a,b と異なる端子点 $c=(x_c, y_c)$ について，$cap(a, b) = cap(a, c) + cap(b, c) + 1$ が満たされるとき，そしてそのときのみ，c は $rect(a, b)$ 内に存在する．

（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．c の座標を (x_c, y_c) とし，これが $rect(a, b)$ の内部に入るとき，$cap(a, c) = |x_a - x_c| - 1$，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 $cap(a, b) = |x_a - x_b| - 1 \leqq |x_a - x_c| + |x_b - x_c| - 1 = |x_a - x_c| - 1 + |x_b - x_c| - 1 + 1 \leqq cap(a, c) + cap(b, c) + 1$ 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 $|x_a - x_c| \geqq |y_a - y_c|$ かつ $|x_b - x_c| \geqq |y_a - y_c|$ である．この範囲は $rect(a, b)$ の範囲と等しい．$cap(a, b) = |y_a - y_b| - 1$ の場合についても同様である． □

図 3に容量判定グラフの例を示す．図 4はこのグラフの辺の中で，点 a を一方の端点とするものを示している．a の上の領域にある網がけ部分は，a の上の領域内に存在するすべての点について，各点の上の領域の和集合である．a の右，左および下の領域の網がけについても同様である．網がけ以外の空白の領域には端子点が存在しない．a を一方の端点とする容量判定グラフの辺は，網がけ領域の境界の凸の位置に存在する点である．網がけの領域内部の任意の点 x について，辺 (a, x) は容量判定グラフには含まれない．なぜなら $rect(a, x)$ の内部に必ず他の点が存在するからである．

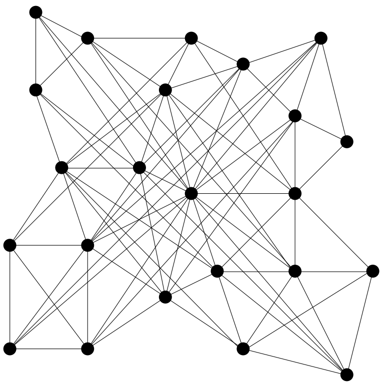

図 3 容量判定グラフの例
Fig. 3 An example of the Capacity-Checking-Graph.

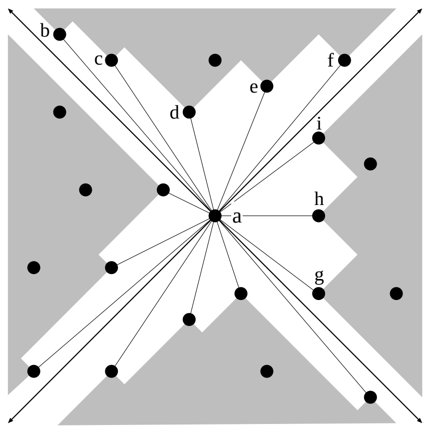

図 4 点 a を一方の端点とする容量判定グラフの辺
Fig. 4 Edges with "a" as an endpoint in the Capacity-Checking-Graph.

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

（証明） $rect(a, b)$ 内に存在する端子点 c で，$\triangle abc$ の内部に他の端子点が存在しないものを選ぶ．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 からはたかだか 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，$flow(a, b) \leqq flow(a, c) + flow(b, c) + 1$ である．仮定より $cap(a, c) \geqq flow(a, c)$ かつ $cap(b, c) \geqq flow(b, c)$ なので，$cap(a, b) = cap(a, c) + cap(b, c) + 1 \geqq flow(a, c) + flow(b, c) + 1 \geqq flow(a, b)$ となり，$cap(a, b) \geqq flow(a, b)$ 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 $cap(a, b) \geqq flow(a, b)$ ならば，$\mathbf{G}^*$ の任意の辺 (p, q) についても $cap(p, q) \geqq flow(p, q)$

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，$e_2, \ldots$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．もし端子点 e が $rect(c, d)$ 内に存在するなら 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．次に任意の i($\leq k$) 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，$rect(c, d)$ 内に他の端子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．$rect(c, d)$ 内に他の端子点が存在する場合を考える．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，帰納法の仮定により $cap(e, f) \geq flow(e, f)$ である．よって補題 2 より $cap(c, d) \geq flow(c, d)$ である．従って，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できることがわかる．

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を支配すると言い，$p \succ q$と表記する．また，点集合 $\mathbf{V}$が与えられたとき，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 $r(\in \mathbf{V})$ が存在しないとき p は $\mathbf{V}$の中で qを直接支配すると言う．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とき，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 [6]．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 [3]．容量判定グラフの辺は，座標軸を$-45°$および $45°$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．

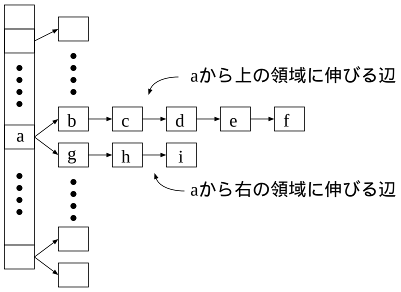

図 5　容量判定グラフのデータ構造
Fig. 5　Data structure for the Capacity-Checking-Graph.

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以下のように構成する．これは 4 章で述べるフロー導出グラフの作成およびフローの算出を効率良く行うためのものである．

1 次元配列に $\mathbf{V}$の点が入る．配列の各点には二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中で上方向およ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．図 4のグラフに対応するデータ構造を図 5に示す．

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，左およ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．

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．容量については辺の両端点の座標より求めることができる．ここ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れば，“フリップ操作” を繰り返し行うことにより，容量判定グラフのすべての辺について 1 本当たり $O(1)$ 時間でフローを求めることができる．

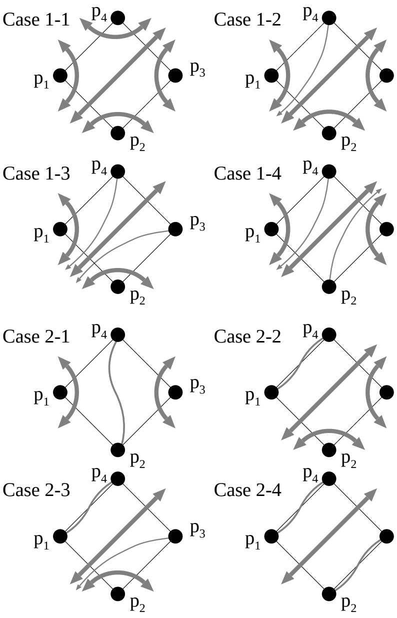

図 6　四角形内の配線の場合分け
Fig. 6　Wire toporogy in a rectangle.

4.1　フリップ操作による flow の算出

凸四角形内の概略配線について，次の補題が成り立つ．

［補題 3］ **V** を端子点の集合とし，$\square v_1v_2v_3v_4(v_i \in \mathbf{V})$ を内部に端子点を含まない凸四角形とする．フローを定義 1 に従って定めたとき，次の関係が成り立つ．$flow(v_1,v_3)+flow(v_2,v_4) = max(flow(v_1,v_2)+flow(v_3,v_4), flow(v_2,v_3)+flow(v_4,v_1))$

(証明)　配線経路のトポロジの場合分けを行う (図 6)．まず，外周辺若しくは対角辺と重なる配線がない場合 (Case1) とある場合 (Case2) に分ける．Case1 を更に，四つの頂点のいずれかを始点として四角形内を通過する配線の本数により分ける．その本数が 0 本，1 本のときをそれぞれ Case1-1，Case1-2 とし，2 本のときはそれらが出てゆく辺が一致する場合と異なる場合についてそれぞれ Case1-3，Case1-4 と分ける．Case2 については，対角辺と重なる配線がある場合を Case2-1 とし，外周辺と重なる配線がある場合については，それが 1 本のときに Case2-2 と Case2-3 とし，2 本のときには Case2-4 と分ける (外周辺に重なる配線と対角辺に重なる配線が同時に存在することはあり得ない)．このときいずれの場合でも $flow(p_1,p_2)+flow(p_3,p_4) \geqq flow(p_2,p_3)+flow(p_4,p_1)$ である．よってすべての場合について $flow(p_1,p_2)+flow(p_3,p_4) = flow(p_1,p_3)+flow(p_2,p_4)$ が成り立つことを示せばよい．

Case1 について考える．辺 (p_1,p_2) から四角形内に入り辺 (p_3,p_4) から出ていく 1 本の配線は，$flow(p_1,p_2)$，$flow(p_3,p_4)$，$flow(p_1,p_3)$ および $flow(p_2,p_4)$ に対してそれぞれ 1 加算するので，上の等式の右辺と左辺に対してそれぞれ 2 を加算する．それ以外のすべての配線は $flow(p_1,p_2)$ または $flow(p_3,p_4)$ のどちらか一方に対して 1 加算し，かつ $flow(p_1,p_3)$ または $flow(p_2,p_4)$ のどちらか一方に対して 1 加算するため，上の等式の右辺と左辺に対してそれぞれ 1 を加算する．よって，どのような場合についても右辺と左辺の均衡は崩れない．次に Case2 について考える．対角辺と重なる配線 (Case2-1) は，$flow(p_1,p_3)$ と $flow(p_2,p_4)$ に対してそれぞれ 1 と -1 を加算するのみで $flow(p_1,p_2)$ と $flow(p_3,p_4)$ に対しては影響を与えないため，上の等式の右辺と左辺に対して影響を与えない．外周辺と重なる配線 (Case2-2〜4) は $flow(p_1,p_2)$，$flow(p_3,p_4)$，$flow(p_1,p_3)$ および $flow(p_2,p_4)$ のいずれに対しても影響を与えないため，上の等式の右辺と左辺に対して影響を与えない．それ以外のすべての配線は Case1 の場合の議論と同様である．よって，右辺と左辺の均衡は崩れないことがわかる．従って，Case1 と Case2 のいずれについても上の等式が成り立つ．　□

補題 3 の式を用いることにより，凸四角形について，四つの外周辺と一つの対角線のフローが与えられたとき，それらをもとにもう一つの対角辺についてのフローを求めることができる．この操作をフリップ(注2)と呼ぶことにする．図 7の例では，$\square ABDC$についてフ

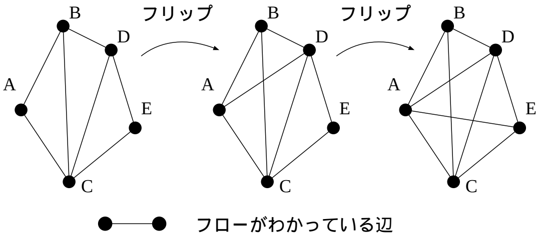

図 7　フリップ操作
Fig. 7　Flippings.

(注 2)：計算幾何学においてフリップは，凸四角形内の対角辺を交換し三角分割を変更する操作を言うものであるが，本論文中のフリップは単に未知の対角辺のフローを求める操作を指し，もとの対角辺のフローの情報を破棄してしまうものではない．

リップを行い $flow(A,D)$ を求め，次に □$ADEC$ についてフリップを行い $flow(A,E)$ 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．

4.2 フロー導出グラフの定義と諸定理

$\mathbf{V}$を配線領域内の端子点の集合とする．フロー導出グラフ $\mathbf{G}_f=(\mathbf{V},\mathbf{E}_f)$ は容量判定グラフ $\mathbf{G}_c=(\mathbf{V},\mathbf{E}_c)$ の部分グラフで，辺集合 $\mathbf{E}_f$は以下の条件を満たすものとする．

$(a,b)\in\mathbf{E}_f$ iff (i) $\nu_a \leqq \nu_b$ (ii) 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．

図 3の端子点配置に対するフロー導出グラフを図 8 に示す．フロー導出グラフについて，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 (図 9)．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できる．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 $(a,u_i)(1\leqq i\leqq\alpha)$ 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. 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． □

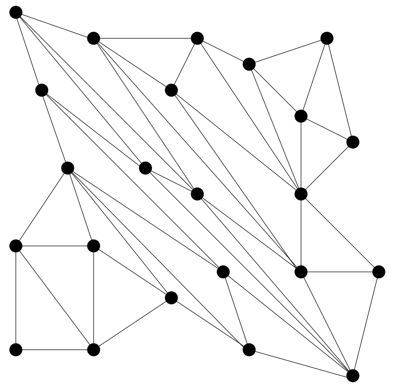

図 8 フロー導出グラフの例
Fig. 8 An example of the Initial-Flow-Graph.

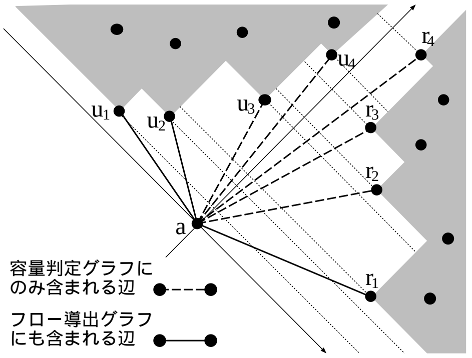

図 9 フロー導出グラフの辺の例
Fig. 9 Edges with “a” as an endpoint in the Initial-Flow-Graph.

［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について考える．傾きが$-45°$に近い方のカットを (a,b) とし，45°に近い方のカットを (c,d) する．$\mu_a\leqq\mu_b$，$\nu_c\leqq\nu_d$とする．もし $rect(a,b)$ 内に c か d が存在するなら，フロー導出グラフの定義より辺 (a,b) は $\mathbf{G}_f$には含まれない．同様に $rect(c,d)$ 内に a か b が存在するなら，辺 (c,d) は $\mathbf{G}_f$には含まれない．$rect(a,b)$ および $rect(c,d)$ 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．よってフロー導出グラフの定義より，辺 (c,d) は $\mathbf{G}_f$に含まれない．互いに交差する辺のうち，必ず一方は $\mathbf{G}_f$に含まれないので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 $\mathbf{G}_c = (\mathbf{V}, \mathbf{E}_c)$ およ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 $flow(p, q)$ がわかっているとき，$O(|\mathbf{E}_c|)$ の実行時間で $\mathbf{G}_c$のすべての辺 (a, b) について $flow(a, b)$ を求めることができる．

(証明) スイープ法により右上の辺からフローを計算していく．配線領域において$-45°$軸に平行なスイープラインを考え右上方向から左下方向に並行移動 (走査) する．イベントポイントは各端子点とする．$\mathbf{G}_c$に含まれる辺のうち，両端点がスイープ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．今，スイープラインが端子点 a で止まったとする．$\mathbf{G}_c$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．また，$u_1, u_2, ..., u_\alpha$，$r_1, r_2, ..., r_\beta$を $45°$軸に関して小さい順にマージしたものを $w_1, w_2, ..., w_{\alpha+\beta}$とする．そして $flow(a, w_1), flow(a, w_2), flow(a, w_3)...$の順に $flow$を求めていく．今 $flow(a, w_i)$ に注目したとき，辺 (a, w_i) が $\mathbf{G}_f$に含まれないならば次のことが言える．w_iが u_kであるとする．u_1は必ず $\mathbf{G}_f$に含まれるため $u_k \neq u_1$であり，u_{k-1}が存在する．また，$r_1, ..., r_\beta$のうち，$45°$軸に関する大きさが w_iよりも小さくかつ最も近いものを r_mとする．このような r_mは必ず存在する．なぜなら r_mが存在しないとき，フロー導出グラフの定義により辺 (a, w_i) が $\mathbf{G}_f$に含まれるからである．このとき，図 10に示すように，$rect(u_{k-1}, u_k)$，$rect(u_{k-1}, r_m)$ および $rect(u_k, r_m)$ 内には他の点は存在しない．よって，辺 (u_{k-1}, u_k)，(u_{k-1}, r_m)，(u_k, r_m) は $\mathbf{G}_c$に含まれ，スイープラインより右上にあるのでフローは既に求められている．また辺 (a, u_{k-1}) と (a, r_m) については $\mathbf{G}_c$に含まれ，これまでのフリップ操作によりフローは既に求められている．四角形 $au_{k-1}u_kr_m$の外周の四つの辺と一つの対角線についてフローがわかっているので，この四角形内でフリップを行うことにより $flow(a, u_k)$ すなわち $flow(a, w_i)$ を求めることができる．w_iが r_kである場合も同様であり，任意の $flow(a, w_i)$ はそれぞれ一度のフリップで求めることができる．$w_1, w_2, ..., w_{\alpha+\beta}$は図 5のデータ構造の順に $O(1)$ 時間で見つけることができ，フリップに必要な実行時間は 1 回当たり $O(1)$ なので，辺 1 本当たり $O(1)$ の時間でフローを求めることができる．このアルゴリズムはすべての点を走査し終えたときに終了し，このとき容量判定グラフのすべての辺についてフローが求められている．実行時間は $O(|\mathbf{E}_c|)$ である． □

5. 概略配線処理手順

図 11に本論文で提案した容量判定グラフとフロー導出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．これらは最初に 1 回行うのみでよい．またフロー算出にかかる時間は

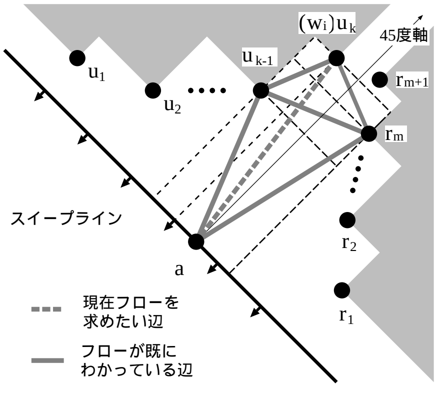

図 10 フローの求め方
Fig. 10 Finding a flow in the Capacity-Checking-Graph.

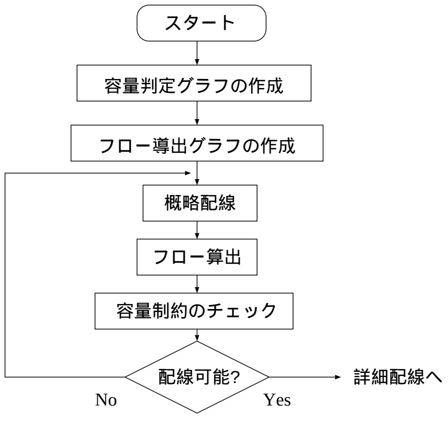

図 11 提案する配線処理手順
Fig. 11 Proposed flowchart for the routing problem.

$O(|\mathbf{E}_c|)$，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，配線可能性検証に関するすべての操作は $O(n \log n)$ 時間で実行できる．もし配線不能であるときには概略配線の更新を行うが，その後の検証では変更した経路と交わる容量判定グラフの辺についてのみ容量制約のチェックを行えばよい．

この配線処理手順では，フロー導出グラフの辺に対するフロー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三角形分割し，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 Delaunay 三角形分割して経路探索を行うものがあるが [4], [9]，Delaunay 三角形分割された領域に対してフリップを行うことにより，フロー導出グラフを分割辺として含む三角分割に変形させるという手法も考えられる．

6. むすび

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．

本論文で扱うモデルは 1 層配線であるが，多層配線への応用も考えられる．層分割を行った後に各層ごとに概略配線を行う．配線の層間接続はスルーホールにより行うため，このスルーホールを端子点として扱うことにより本アルゴリズムを適用することができる．

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している．今後の課題として，モジュールなどの配線禁止領域や，配線の枝分かれを考慮に入れる等，配線モデルの拡張を行うことが挙げられる．

謝辞　本研究に関し，貴重な御助言を賜りました大阪電気通信大学の浅野哲夫教授，中京大学の伊藤誠教授ならびに東海大学の譚学厚講師に感謝致します．

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] W.W.Dai, R.Kong, and M.Sato, "Routability of a rubber-band sketch," 28th ACM/IEEE Design Automation Conference, pp.45–48, 1991.

[3] R. Guting, O. Nurmi, and T. Ottmann, "Fast algorithms for direct enclosures and direct dominances," Journal of Algorithms 10, pp.170–186, 1989.

[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," 信学技報 VLD 94-70, 1994.

[5] 川口 泰, 磯 直行, 平田富夫, "配線可能性検証のための容量判定グラフの提案," 信学技報 VLD 95-126, 1995.

[6] R. Klein, "Direct dominance of points," Int.J.Comput.Math., Vol.19, pp.225–244, 1986.

[7] 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.

[8] Y.Shiraishi, J.Sakemi, and K.Fukuda, "A global routing algorithm based on the multi-commodity network flow method," IEICE Trans. Fundamentals, vol.E76-A, No.10 Oct. 1993.

[9] 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," 情処学研報 DA 70-9, pp.63–70, 1994.

（平成 8 年 1 月 4 日受付，6 月 4 日再受付）

川口　泰　（正員）

平 6 名大・工・電気卒．平 8 同大大学院博士課程前期課程了．現在，富士通株式会社勤務．在学中は遺伝的アルゴリズムの応用に関する研究，配線問題に関する研究に従事.

磯　直行

平 2 山梨大・工・電子工卒，平 4 同大大学院修士課程了，平 7 名大大学院博士課程後期課程了，現在，名大・電子工学科助手．VLSI，プリント配線板のレイアウト設計に関する研究に従事．情報処理学会会員．

平田　富夫　（正員）

昭 51 東北大・工・通信卒．昭 56 同大大学院博士課程了．昭 56 豊橋技科大助手，昭和 61 名大・情報工学科講師を経て，現在，名大・電子工学科教授．グラフアルゴリズムとデータ構造の研究に従事．情報処理学会会員．工博．