配線禁止領域を考慮した配線可能性検証

川口 泰* 磯 直行 平田 富夫
名古屋大学 工学部

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けて行う場合がある．概略配線から詳細配線への変換可能性の検証を配線可能性検証という．我々は端子点のみが存在する配線モデルについて，“容量判定グラフ” と “フロー導出グラフ” を提案し，それらを用いて効率の良い配線可能性検証ができることを示した．本論文では，配線モデルを拡張し，矩形配線禁止領域 (モジュール) が存在する場合を取り扱う．

In VLSI and printed wiring board design, wire routing consists of the two stages: global routing and detailed routing. Routability checking is to decide whether the global wires can be transformed into the detailed ones or not. We have proposed two graphs, the capacity checking graph and the initial flow graph, for efficient routability checking in the case of no obstacle. In this paper, we treat the routing model with rectangular obstacles.

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う手法が提案されている．この手法においては，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判定し，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．概略配線から詳細配線への変換可能性を判定する作業を配線可能性検証という．

配線領域内に端子点のみが存在する配線モデルにおける配線可能性検証のために，筆者らは文献 [1] で容量判定グラフを提案した．容量判定グラフは配線領域内のすべての端子点を節点とするグラフである．グラフ内の各辺には，その辺 (すなわち 2 つの端子点間) を通過することのできる配線本数 (容量) が定められる．そして，すべての辺について，その容量が辺上を通過している概略配線の本数 (フロー) を上回るか，または等しいとき，概略配線から詳細配線への変換可能性が保証される．すなわち配線可能性検証は，容量判定グラフの辺の容量とフローの比較により行うことができる．n 個の端子点が配線領域内にランダムに分布しているとき容量判定グラフの辺の数は $O(n \log n)$ でおさえられる．したがって，このグラフを利用することにより配線可能性検証は $O(n \log n)$ 時間で実行できる．

このグラフを用いた検証手法を実用化するためには，より現実的な配線モデルを考えなくてはならない．本論文では，配線領域内に端子点に加えて矩形の配線禁止領域 (矩形モジュール) が存在する配線モデルを考える．この配線モデルにおいて効率良く配線可能性検証を行うために容量判定グラフの再定義を行う．新しい容量判定グラフにおいては，端子点と矩形モジュールの数の合計を n としたとき節点数 $|\mathbf{V}|$ はたかだか $5n$ である．また，辺の数を $|\mathbf{E}|$ とすると，このグラフは $O(n \log n + |\mathbf{E}|)$ の実行時間で作成できる．なお，$|\mathbf{E}|$ の期待値は，配線領域内に端子点およびモジュールがランダムに分布しているとき，平均で $O(n \log n)$ であることが予想される．

以下に，2. で配線モデルと言葉の定義を行い，3. では拡張した配線モデルにおける容量判定グラフについて述べ，4. では容量判定グラフを効率良く作成するアルゴリズムを与える．5. では配線可能性検証を実際に行うときに必要となるフロー導出グラフについて述べ，6. でまとめを行う．

2 拡張配線モデルと言葉の定義

まず，本論文で扱う配線モデルについて述べる．配線領域は単層で縦横格子が引かれているものとする．配線領域の境界線は四角形で，領域内には端子点および矩形形状のモジュール (以下では単にモジュールという) が互いに重ならないように配置されている．端子点および配線領域の境界線とモジュールの外周線の両端点は格子点上に位置している．端子点には配線経路が 1 本だけ接続されている．配線経路は端子点を始点および終点とし，互いに交わらず，かつ他の端子点上やモジュール内を通過しない．配線経路をトポロジで表現したものが概略配線であり (図 1(a))，配線経路をすべて格子上に乗せたものが詳細配線である (図 1(b))．以下では点はすべて格子点のことである．

配線領域内の 2 点 a, b を考える．線分$\overline{ab}$がモジュールまたは端子点と交わらないとき，a と b は互いに可視であるという．端子点の集合を $\mathbf{V}_t$とする．モジュールの内部と配線領域の外部を配線禁止領域とよび，その境界上にある格子点を境界点とよぶ．境界点の集合を $\mathbf{V}_b$とする．$\mathbf{V}_t$と $\mathbf{V}_b$の和集合を $\mathbf{V}_{t+b}$とする．モジュールの 4 隅の点をモジュール頂点とよび，その集合を $\mathbf{V}_m(\mathbf{V}_m \subset \mathbf{V}_b)$ とする．$\mathbf{V}_t$と $\mathbf{V}_m$の和集合を $\mathbf{V}_{t+m}$とする．モジュール頂点でない境界点の集合を $\mathbf{V}_{b-m}$とする．点 a を端子点もしくはモジュール頂点とする．a を通る水平または垂直な直線と交わる境界点のうち，a から可視なものを a の投影点とよぶ．投影点の集合を $\mathbf{V}_p$とする．ひとつの端

Efficient Routability Checking for Routing Model with Rectangular Obstacles
Yasushi KAWAGUCHI*, Naoyuki ISO and Tomio HIRATA
Faculty of Engineering, Nagoya University
*Presently with Fujitsu Corporation

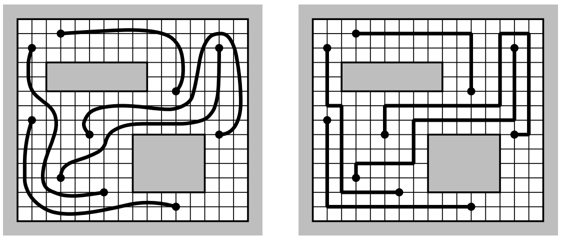

(a) 概略配線　(b) 詳細配線

図 1: 配線モデル

子点はたかだか 4 つの投影点を持ち，ひとつのモジュール頂点はたかだか 2 つの投影点を持つ．

格子の縦と横をそれぞれ一単位長とし，2 点 a,b 間の L_1距離を $||(a,b)||$ と表記する．すなわち $||(a,b)|| = max(|x_a - x_b|, |y_a - y_b|)$ である．互いに可視な 2 点 $p,q(\in \mathbf{V}_{t+b})$ 間を結ぶ線分をカットとよぶ．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，$cap(p,q) = ||(p,q)|| - 1$ 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[1]．$cap(p,q) \geq flow(p,q)$ となる条件をカット (p,q) に対する容量制約という．このとき文献 [2] で次の定理が示されている．

定理 1 概略配線が与えられたとする．全てのカットに対して容量制約がみたされるとき，そしてそのときのみ概略配線から詳細配線への変換が可能である．

可視グラフ $\mathbf{G}_v = (\mathbf{V}_{t+b}, \mathbf{E}_v)$ を次のように定義する．節点集合を $\mathbf{V}_{t+b}$とし，辺集合 $\mathbf{E}_v$は互いに可視な 2 点 $(\in \mathbf{V}_{t+b})$ の対の集合である．このとき定理 1 より $\mathbf{G}_v$の任意の辺に対して容量制約がみたされるとき，そしてそのときのみ概略配線から詳細配線への変換が可能である．

a と b を配線領域内の 2 点とする．$a,b \in \mathbf{V}_{t+b}$のとき，aまたはbを通る傾きが $45°$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，$rect(a,b)$ と表記する．$a \in \mathbf{V}_{t+b}$，$b \in \mathbf{V}_{b-m}$のとき，a を通る傾きが $45°$および$-45°$の 2 本の直線と，b を通り b が属す境界を含む直線により作られる三角形を，a と b により作られる三角形と呼び，$tri(a,b)$ と表記する．(b が配線領域の境界の 4 隅の点のとき $tri(a,b)$ は二つ考えられるが，その内どちらか一方をとるものとする．)

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 の上の領域とよぶ．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 拡張したモデル上での容量判定グラフ

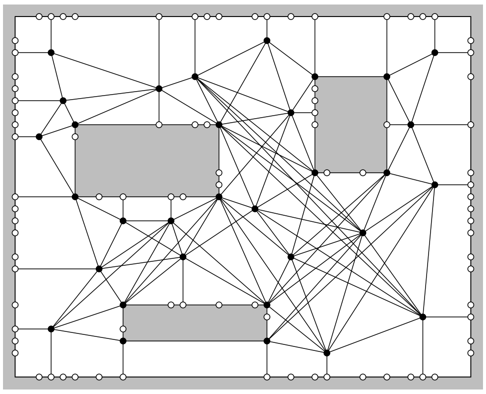


図 2: 容量判定グラフの例

定義 1 容量判定グラフ $\mathbf{G}_c = (\mathbf{V}_c, \mathbf{E}_c)$ は可視グラフ $\mathbf{G}_v = (\mathbf{V}_{t+b}, \mathbf{E}_v)$ の部分グラフであり，節点集合 $\mathbf{V}_c$ は $\mathbf{V}_{t+m}$と $\mathbf{V}_p$の和集合である．辺集合 $\mathbf{E}_c$は次のように定められる．(i) $a,b \in \mathbf{V}_{t+m}$のとき，$rect(a,b)$ 内に a と b から可視な他の点 $c(\in \mathbf{V}_{t+m})$ が存在しないならば $(a,b) \in \mathbf{E}_c$である．(ii) $a \in \mathbf{V}_{t+m}$，$b \in \mathbf{V}_p$のとき，b が a の投影点であり，しかも $\mathrm{tri}(a,b)$ 内に他の点 $c(\in \mathbf{V}_{t+m})$ が存在しないならば $(a,b) \in \mathbf{E}_c$である．(iii) それ以外のとき，(a,b) は $\mathbf{E}_c$に含まれない．

図 2に容量判定グラフの例を示す．容量判定グラフについては以下の補題と定理が成り立つ．

補題 1 $(a,b) \in \mathbf{E}_v$とする．$rect(a,b)$ の内部に a と b から可視な点 $c(\in \mathbf{V}_{t+b})$ が存在し，$\triangle abc$ 内には他の点 $(\in \mathbf{V}_{t+b})$ が存在しないとする．このとき，辺 (a,c) と辺 (b,c) について容量制約がみたされているならば，辺 (a,b) についても容量制約はみたされる．

(証明) c は $rect(a,b)$ の内部に存在するため $||(a,b)|| = ||(a,c)|| + ||(b,c)||$ である．よって $cap(a,b) = cap(a,c) + cap(b,c) + 1$ が成り立つ．辺 (a,b) を通過するフローは，辺 (a,c) および辺 (b,c) を通過する配線と，点 c から発生するたかだか一本の配線がすべて (a,b) を通過するとき最大となり，$flow(a,b) \leq flow(a,c) + flow(b,c) + 1$ である．よって $cap(a,b) = cap(a,c) + cap(b,c) + 1 \geq flow(a,c) + flow(b,c) + 1 \geq flow(a,b)$ となり，辺 (a,b) についても容量制約がみたされる． ■

定理 2 概略配線が与えられたとき，$\mathbf{G}_c$の全ての辺 (a,b) について $cap(a,b) \geq flow(a,b)$ ならば，$\mathbf{G}_v$の任意の辺 (p,q) についても $cap(p,q) \geq flow(p,q)$ である．

(証明) $\mathbf{G}_v$の辺の長さ (L_1距離) に関する帰納法により証明する．まず，$||(p,q)|| = 1$ である任意の辺 $(a,b)(\in \mathbf{E}_v)$ について，$cap(a,b) \geq flow(a,b)$ が成り立つことを次の 3 つの場合にわけて証明する．

(i) $a,b \in \mathbf{V}_{t+m}$の場合．$rect(a,b)$ の内部には他の点 $(\in \mathbf{V}_{t+m})$ は存在しないため $(a,b) \in \mathbf{E}_c$であり，仮定より $cap(a,b) \geq flow(a,b)$ (このとき $flow(a,b) = 0$) である．(ii) $a \in \mathbf{V}_{t+m}, b \in \mathbf{V}_{b-m}$の場合．$b$ が a の投影点であるときは，$(a,b) \in \mathbf{E}_c$であり，仮定より $cap(a,b) \geq$

---

[1] 詳しくは [1] を参照されたい．

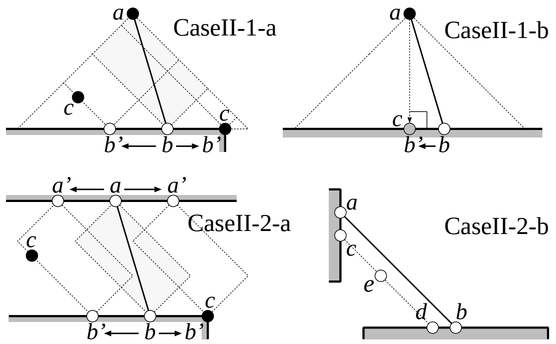


図 3: 定理 2 の証明における場合分け

$flow(a,b)$ である．b が a の投影点でないときは，b は a の投影点 c に隣接する境界点であり，$cap(a,c) = cap(a,b)$ かつ $flow(a,b) = flow(a,c)$ となる．辺 (a,c) については (i) の議論により容量制約がみたされるため，$cap(a,b) \geq flow(a,b)$ となる．*(iii)* $a,b \in \mathbf{V}_{b-m}$ の場合．a と b は配線領域の境界の 4 隅にある格子点の二つの隣接点であるか，または格子一つ分隙間の空けて並んでいるふたつの配線禁止領域 (モジュール) に属する境界点である．前者の場合は，辺 (a,b) にトポロジ的に交差する配線は存在しないため $flow(a,b) = 0$ である．よって $cap(a,b) \geq flow(a,b)$ である．後者の場合はその二つの配線禁止領域の隙間の出口に当たる 2 点 p,q のうち必ず一方は $\mathbf{V}_{t+m}$ の点であるため，*(i)* または *(ii)* の議論により $cap(p,q) \geq flow(p,q)$ である．よって $flow(a,b) = flow(p,q) = 0$ であり，$cap(a,b) \geq flow(a,b)$ が成り立つ．

次に，長さが k 未満の辺 $(p,q)(\in \mathbf{E}_v)$ について $cap(p,q) \geq flow(p,q)$ であるとき，長さが k の辺 $(a,b)(\in \mathbf{E}_v)$ についても $cap(a,b) \geq flow(a,b)$ が成り立つことを証明する．(a,b) が $\mathbf{G}_c$ に含まれる場合は，定理の仮定より $cap(a,b) \geq flow(a,b)$ であるので，(a,b) が $\mathbf{G}_c$ に含まれない場合のみを考えればよい．まず，$rect(a,b)$ の内部に他の点 $(\in \mathbf{V}_{t+b})$ が存在するかどうかで場合分けする．存在する場合を Case I，存在しない場合を Case II とする．

Case I について考える．$rect(a,b)$ 内に存在する点 $(\in \mathbf{V}_{t+b})$ の中で，$\triangle abc$ 内に他の点 $(\in \mathbf{V}_{t+b})$ が存在しないものを c とする．このような c は，a と b から可視であり，$||(a,c)|| < ||(a,b)||$ かつ $||(b,c)|| < ||(a,b)||$ である．帰納法の仮定により $cap(a,c) \geq flow(a,c)$ かつ $cap(b,c) \geq flow(b,c)$ であり，補題 1 より $cap(a,b) \geq flow(a,b)$ が成り立つ．

Case II を a と b の種類により場合分けし，$a \in \mathbf{V}_{t+m}$ かつ $b \in \mathbf{V}_{b-m}$ の場合を Case II-1 とし，$a,b \in \mathbf{V}_{b-m}$ の場合を Case II-2 とする．$a,b \in \mathbf{V}_{t+m}$ の場合はあり得ない．なぜなら，もし $a,b \in \mathbf{V}_{t+m}$ ならば $rect(a,b)$ の内部に他の点 $(\in \mathbf{V}_{t+m})$ が存在しないことにより (a,b) は $\mathbf{G}_c$ の辺となるため，これは最初の仮定に反するからである．

Case II-1 を考える．このとき b は $tri(a,b)$ 内に存在する．もしそうでなければ b に隣接する 2 つの境界点のうちいずれかが $rect(a,b)$ の内部に存在し，Case II の仮定に反するからである．まず，$tri(a,b)$ の内部に b 以外の点 $(\in \mathbf{V}_{t+m})$ が存在する場合を考える (図 3 Case II-1-a)．a を固定したまま，b を b が属す外周線にそって移動させ (方向はどちらでもよい)，$rect(a,b)$ の境界上に初めて現れた点 $(\in \mathbf{V}_{t+m})$ を c とする．また，このときの b を b' とする．このとき $cap(a,b) = cap(a,b')$ である．また辺 (a,b) 間を通過する配線と辺 (a,b') 間を通過する配線は等しいため $flow(a,b) = flow(a,b')$ である．c が b' と異なる点であるとき，c は $rect(a,b')$ の境界上に存在するため，Case I の議論により $cap(a,b') \geq flow(a,b')$ である．c が b' と同一の点であるとき，$rect(a,b')$ 内に他の点 $(\in \mathbf{V}_{t+m})$ が存在しないため，(a,b') は $\mathbf{G}_c$ に含まれ，よって $cap(a,b') \geq flow(a,b')$ である．よって $cap(a,b) = cap(a,b') \geq flow(a,b') = flow(a,b)$ が成り立つ．

次に $tri(a,b)$ の内部に b 以外の点 $(\in \mathbf{V}_{t+m})$ が存在しない場合を考える (図 3 Case II-1-b)．c を a の投影点とする．先の議論と同様に $cap(a,b) = cap(a,c)$ かつ $flow(a,b) = flow(a,c)$ である．また $tri(a,c)$ 内に他の点 $\mathbf{V}_{t+m}$ が存在しないため，(a,c) は $\mathbf{G}_c$ に含まれ，よって $cap(a,c) \geq flow(a,c)$ である．よって $cap(a,b) = cap(a,c) \geq flow(a,c) = flow(a,b)$ が成り立つ．

Case II-2 について，まず a と b が属する境界線が互いに並行である場合を考える (図 3 Case II-2-a)．b は $tri(a,b)$ の内部に存在し a は $tri(b,a)$ の内部に存在している．もしそうでなければ，a か b に隣接する 4 つの境界点のうちいずれかが $rect(a,b)$ の内部に存在し，Case II の仮定に反するからである．a を a が属す外周線に，b を b が属す外周線にそって同じ方向に移動させたとき，$rect(a,b)$ の境界上に初めて現れた点 $(\in \mathbf{V}_{t+m})$ を c とする．このときの a と b をそれぞれ a' と b' とする．このとき $cap(a,b) = cap(a',b')$ である．辺 (a,b) 間を通過する配線と辺 (a',b') 間を通過する配線は等しいため $flow(a,b) = flow(a',b')$ である．c が a' または b' と異なる点のときは，c は $rect(a',b')$ 内に存在するため，Case I の議論により $cap(a',b') \geq flow(a',b')$ である．c が a' または b' と同じ点のとき，Case II-1 の議論により $cap(a',b') \geq flow(a',b')$ である．従って $cap(a,b) = cap(a',b') \geq flow(a',b') = flow(a,b)$ が成り立つ．

次に a と b が属する境界線が互いに垂直である場合を考える (図 3 Case II-2-b)．線分 $\overline{ab}$ と a または b が属している境界面とのなす角は $45°$ である．さもなければ a または b に隣接する境界点が $rect(ab)$ 内に存在し Case II の仮定に反するからである．a に隣接する境界点のうち b に近いものを c とし，b に隣接する境界点のうち a に近いものを d とする．線分 $\overline{cd}$ 上に他の点 $(\in \mathbf{V}_{t+m})$ が存在しない場合は，$cap(a,b) = cap(c,d) + 1$ であり，かつ $flow(a,b) = flow(c,d)$ である．よって $cap(a,b) \geq flow(a,b)$ が成り立つ．線分 $\overline{cd}$ 上に他の点 $e(\in \mathbf{V}_{t+m})$ が存在する場合は，$cap(a,b) = cap(c,e) + cap(d,e) + 2$ であり，かつ $flow(a,b) \leq flow(c,e) + flow(d,e) + 1$ である．よって帰納法の仮定により，$cap(a,b) \geq flow(a,b)$ が成り立つ． ■

定理 1 と定理 2 より，容量判定グラフの辺についてのみ容量制約をチェックすれば，配線可能性検証が行えることがわかる．

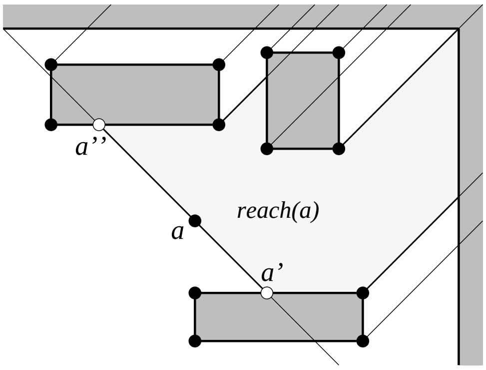


図 4: a の到達領域

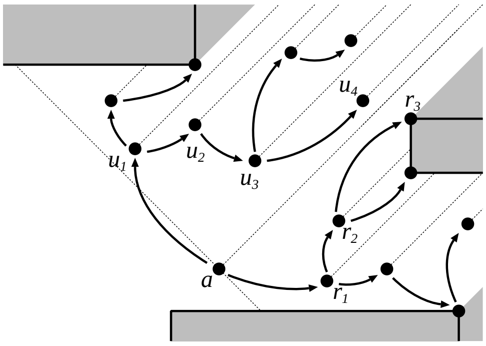


図 5: a の到達木

4　容量判定グラフの作成

文献 [1] においては，容量判定グラフの辺の両端点は直接支配 (direct dominance) の関係にあった．直接支配集合の要素を列挙する効率の良いアルゴリズムが文献 [4] で既に与えられており，そのアルゴリズムを用いて容量判定グラフを作成することができた．しかし本論文の拡張したモデル上の容量判定グラフの辺は，厳密には直接支配関係をみたす点の対ではない．そのため [4] のアルゴリズムをそのまま適用して容量判定グラフを作成することはできない．本章ではスイープ法を用いた容量判定グラフ $\mathbf{G}_c = (\mathbf{V}_c, \mathbf{E}_c)$ を作成する効率の良いアルゴリズムを与える．入力は端子点およびモジュール頂点の集合 $\mathbf{V}_{t+m}$ であり，出力は辺集合 $\mathbf{E}_c$ である．本アルゴリズムは $O(|\mathbf{V}_{t+m}|\log|\mathbf{V}_{t+m}| + |\mathbf{E}_c|)$ 時間で実行でき，漸近的に最適である．また，配線禁止領域を考えない場合の (文献 [1] の) モデルについても実行でき，そのときの実行時間は文献 [4] のアルゴリズムを用いた場合と等しい．

本アルゴリズムは次の 3 つの段階から成る．(1)$\mathbf{V}_{t+m}$ の点の投影点の集合 $\mathbf{V}_p$ を求める．(2) 両端点が共に $\mathbf{V}_{t+m}$ に含まれる容量判定グラフの辺を求める．(3) 一方の端点が $\mathbf{V}_{t+m}$ に含まれ，もう一方の端点が $\mathbf{V}_p$ に含まれる容量判定グラフの辺を求める．

段階 (1) は，スイープ法による平面走査を上下左右方向に 4 回行うことにより，$O(|\mathbf{V}_{t+m}|\log|\mathbf{V}_{t+m}|)$ 時間で行えることは明らかである．よって本論文では特に言及を避ける．

次に段階 (2) について説明する．配線領域内の任意の点 $a(\in \mathbf{V}_{t+m})$ について考える．a を通る傾き -45° の直線を引き，その直線上にある境界点の中で a から可視な 2 点をそれぞれ a'，a'' とする．この直線より右上方向にある全てのモジュール頂点について，その点を始点として右上 45° 方向伸びる半直線を引く．このとき線分 $\overline{a'a''}$，モジュールの外周線，配線領域の境界線およびモジュール頂点より引いた半直線で囲まれる領域 (図 4) を a の到達領域とよび，$reach(a)$ と表記する．

補題 2 配線領域の互いに可視な 2 点 $a, b(\in \mathbf{V}_{t+m})$ について $\nu_a < \nu_b$ とする．$(a,b) \in \mathbf{E}_c$ ならば b は $reach(a)$ の内部に存在する．

(証明) 対遇を示す．b が $reach(a)$ の内部に存在しないとする．このとき線分 $\overline{ab}$ は，モジュール内を通過するか，またはモジュール頂点から引かれた半直線と交差している．$\overline{ab}$ がモジュール内を通過するとき，a と b は互いに可視でないため $(a,b) \notin \mathbf{E}_c$ である．$\overline{ab}$ がモジュール頂点から引かれた半直線と交差するとき，$rect(a,b)$ 内にそのモジュール頂点が存在する．もしそのモジュール頂点が a と b から可視であれば容量判定グラフの定義により $(a,b) \notin \mathbf{E}_c$ である．もしそのモジュール頂点が a と b から可視でなければ、可視であることを妨げるモジュールまたは端子点が存在する。結局 $rect(a,b)$ には a と b から可視な $\mathbf{V}_{t+m}$ の点が一つは存在するはずである。よってこの場合も $(a,b) \notin \mathbf{E}_c$ である． ■

補題 2より，a から右または上方向に伸びる容量判定グラフの辺を発見するためには $reach(a)$ の内部を探索すればよいことがわかる．

配線領域内の任意の点 $a(\in \mathbf{V}_{t+m})$ について，以下の条件を満たす木を a の到達木とよび，$\mathbf{Tree}(a)$ と表記する．$\mathbf{Tree}(a)$ は a を根とし，$reach(a)$ 内の全ての点 $(\in \mathbf{V}_{t+m})$ を木の要素として持つ．親はたかだか 2 つの子を持ち，その一方は上の子供とよばれ，もう一方は右の子供とよばれる．上の子供は親の上の領域にあり，その子孫たちもすべて親の上の領域にある．右の子供は親の右の領域にあり，その子孫たちもすべて親の右の領域にある (図 5)．このとき，次の補題が成り立つ．

補題 3 $\mathbf{Tree}(a)$ を a の到達木とする．a の上の子供を u_1，右の子供を r_1 とする．任意の i, j について r_i の上の子供を r_{i+1} とし，u_j の右の子供を u_{j+1} とする．このときすべての (a, r_i)，(a, u_j) は $\mathbf{G}_c$ に含まれる．また，r_i，u_j 以外の任意の点 p について，(a, p) は $\mathbf{G}_c$ には含まれない．

(証明) 詳細な証明は省略するが，図 5より，$rect(a, r_i)$ および $rect(a, u_j)$ 内には a および r_i または u_j から可視な点 $(\in \mathbf{V}_{t+m})$ が存在しない．よって任意の (a, r_i) および (a, u_j) は容量判定グラフに含まれる． ■

補題 3より a から上および右方向に伸びる容量判定グラフの辺を発見するためには，$\mathbf{Tree}(a)$ を作成し，木の探索を行えばよいことがわかる．探索に必要な時間は容量判定グラフの辺の数に線形である．到達木は全ての点 $(\in \mathbf{V}_{t+m})$ についてはじめから作成するわけではなく，以前に作られた $\mathbf{Tree}$ を修正することにより作成する．$\mathbf{Tree}(a)$ の作成については，次の補題が成り立つ．

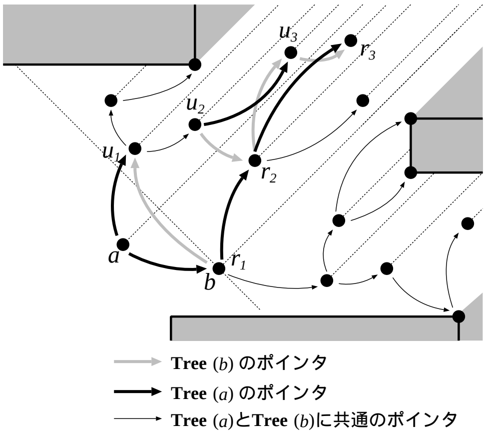


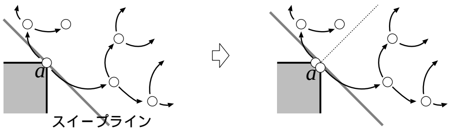
図 6: a の到達木の作成

補題 4 a を配線領域内の端子点とする．$reach(a)$ の内部に存在する点 ($\in \mathbf{V}_{t+m}$) の中で，45°軸に関する大きさが最も小さい点を b とする．このとき $\mathbf{Tree}(b)$ を修正して $\mathbf{Tree}(a)$ を作成することができ，その作成時間は a から上または右方向に伸びる $\mathbf{G}_c$ の辺の数に線形である．

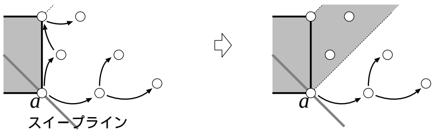
(証明) $reach(b)$ は $reach(a)$ に含まれるため，$\mathbf{Tree}(a)$ の要素は $\mathbf{Tree}(b)$ の要素に b を加えた集合である．b が a の右の領域にあるとする．このとき b を r_1 と名付ける．$\mathbf{Tree}(b)$ のなかで，r_1 から上へ上へと子孫を辿っていき，そのとき発見した点を $r_2, r_3, ...$ とする．はじめて a の上の領域にある点に到達したとき，その点を u_1 とする．これ以降，到達した点が a の右の領域に存在するとき，その点を r_i(i は 1 ずつ増加していく) とし，そこから上の子供を辿る．到達した点が a の上の領域に存在するとき，その点を u_j(jは 1 づつ増加していく) とし，そこから右の子供を辿る．この操作を子供がなくなるまで続ける．このとき $\mathbf{Tree}(b)$ の中に存在し，かつ a の右の領域にある全ての点について考える．任意の i について，r_iおよび r_iの右の子孫は a の右の領域にある．r_iの右の子孫はすべて r_iの右の領域にある．$r_j(j > i)$ は r_iの上の子孫であるため，r_jおよびその子孫は r_iの上の領域にある．よって a の右の子供を r_1，r_iの上の子供を r_{i+1} とし，また r_1の親を a，r_{i+1}の親を r_iとすることにより $\mathbf{Tree}(a)$ の条件をみたす木を作成できる (図 6)．このとき $r_1, r_2, ...$以外の任意の点 qについても qとその子孫の親子関係は $\mathbf{Tree}(a)$ と $\mathbf{Tree}(b)$ とで共通なため，到達木の条件はみたされている．a の上の点についても同様に a の上の子供を u_1，u_jの右の子供を u_{j+1}とし，また u_1 の親を a，u_{j+1}の親を u_jとすればよい．

$\mathbf{Tree}(a)$ の作成に必要な時間は，$\mathbf{Tree}(b)$ の探索中に発見した点 $(r_1, r_2, ..., u_1, u_2, ...)$ の数に線形である．これらの点について次のことがいえる．r_1は a の右の子供であり，$r_i(i \geq 2)$ は r_1から上へ上へと辿った子孫である．u_1は vの上の子供であり，$u_j(j \geq 2)$ は u_1から右へ右へと辿った子孫である．よって補題 3より，任意の i, j について (a, r_i)，(a, u_j) は容量判定グラフに含まれる．従って，補題 4が成り立つ． ■

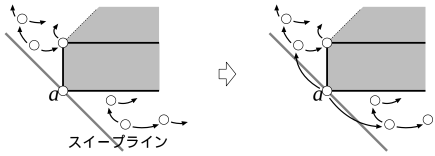
段階 (2) のアルゴリズムの実現にスイープ法を用いる．配線領域に$-45°$軸に並行にスイープラインを引き，右上方向から左下方向に走査する．イベントポイントは $\mathbf{V}_{t+m}$ の点とする．イベントポイントの点 a が端子点のとき，それまでに既に作成されている $\mathbf{Tree}$ を基に $\mathbf{Tree}(a)$ を作成する．そして $\mathbf{Tree}(a)$ の内部で探索を行い，a から上または右の領域に伸びる容量判定グラフの辺を出力する．もし a が端子点でなくモジュール頂点であったなら，次に示す操作を行う必要がある．



(a) イベントポイントがモジュールの右上頂点の場合



(b) イベントポイントがモジュールの 右下(左上)頂点の場合



(c) イベントポイントがモジュールの左下頂点の場合

図 7: イベントポイントがモジュール頂点である場合

a がモジュールの右上の頂点であった場合は (図 7 (a))，到達木を作成しその中を探索した後，根から右の部分木と上の部分木に分割する．このとき根は二つになる．このようにする理由は，a 以降のイベントポイントの点からは，a から右上 45°方向に伸びる半直線により到達領域が二つに分割されるからである．a がモジュールの右下の頂点であった場合は (図 7 (b))，到達木を作成し，その木を探索した後，a の上の子孫を到達木から消去する．なぜなら a 以降のイベントポイントの点では，到達領域に a の上の領域は含まれないからである．a がモジュールの左上の頂点であった場合は，同様に根から右方向の子孫を到達木から消去する．a がモジュールの左下の頂点であった場合は (図 7 (c))，a の上側の到達領域内の $\mathbf{Tree}$ の根を a の上の子供とし，右側の到達領域内の $\mathbf{Tree}$ の根を a の右の子供とする．これは，a 以降のイベントポイントでは a の上側の到達領域と a の右側の到達領域が併合されるからである．この操作を行うのみで，作成された木は a の到達木の条件を満たしている．この後，木の探索を行う．

スイープ中にはそれまでに作成された到達木の集合を森として保持しておく．そして任意のイベントポイント a において，$reach(a)$ 内に存在する到達木を森の中から選択する．選択された木を $\mathbf{Tree}(a)$ に修正し，それを森の中に残すという操作を繰り返す．

このとき各イベントポイントにおいては森の中から目的の到達木を選択しなくてはならない．このため到達木の根へのポインタを 2 色木のような平衡 2 分木の中に保

持しておく．2 色木の要素のオーダーは各到達木の根の -45°軸に関する大きさとする．森の中の到達木の数はたかだか $|\mathbf{V}_{t+m}|$ 個である．n 個の要素をもつ 2 色木では，挿入，消去および探索の操作は $O(\log n)$ 時間で実行できるので，目的の到達木の選択は $O(\log|\mathbf{V}_{t+m}|)$ 時間で行うことができる．

全てのイベントポイントを通過した時アルゴリズムは終了する．片方の端点から上または右方向に伸びる辺は，もう片方の端点からは下または左方向に伸びる辺になっている．よってアルゴリズムの終了時には，求めたい全ての辺が出力されている．

段階 (2) に必要な実行時間について述べる．はじめに点集合 $\mathbf{V}_{t+m}$の点をソートするために $O(|\mathbf{V}_{t+m}|\log|\mathbf{V}_{t+m}|)$ 時間が必要である．スイープ中の各イベントポイントの点について，森の中の到達木を選択し，新しい到達木を作成し，到達木の中を探索するという操作を行う．全てのイベントポイントを通過した時，それぞれの累積実行時間は $O(|\mathbf{V}_{t+m}|\log|\mathbf{V}_{t+m}|)$，$O(|\mathbf{E}_c|)$ および $O(|\mathbf{E}_c|)$ である．従って実行時間の合計は $O(|\mathbf{V}_{t+m}|\log|\mathbf{V}_{t+m}|+|\mathbf{E}_c|)$ となる．

次に，段階 (3) について考える．a を $\mathbf{V}_{t+m}$に含まれる点とし，a'を a の上方向の投影点とする．$tri(a,a')$ 内に他の点 ($\in \mathbf{V}_{t+m}$) が存在するとき，そしてそのときのみ a から上方向に伸びる $\mathbf{G}_c$の辺が存在する．その他の方向についても同様のことがいえる．よって段階 (3) を次のように行うことができる．$\mathbf{V}_{t+m}$の中の任意の点 a について，a から上に伸びる容量判定グラフの辺が段階 (2) で発見されているかどうかを調べる．もし発見されていなければ，(a,a') (a'は段階 1 で求めた a の上方向の投影点) を $\mathbf{G}_c$に含まれる辺として出力する．その他の方向についても同様である．段階 (3) は $O(|\mathbf{E}_c|)$ 時間で実行できる．

よって，次の定理が成り立つ．

定理 3 配線領域の境界と，端子点，モジュール頂点の集合 $\mathbf{V}_{t+m}$が与えられた時，容量判定グラフ $\mathbf{G}_c=(\mathbf{V}_c,\mathbf{E}_c)$ を $O(|\mathbf{V}_{t+m}|\log|\mathbf{V}_{t+m}|+|\mathbf{E}_c|)$ 時間で作成するアルゴリズムが存在する．

5 拡張したモデル上のフロー導出グラフ

定義 2 フロー導出グラフ $\mathbf{G}_f=(\mathbf{V}_c,\mathbf{E}_f)$ は容量判定グラフ $\mathbf{G}_c=(\mathbf{V}_c,\mathbf{E}_c)$ の部分グラフであり，辺集合 $\mathbf{E}_f$は次のように定める．点 a と点 b については$\nu_a \le \nu_b$であるとする．(i) $a,b\in\mathbf{V}_{t+m}$のときは，a の上かつ b の左の領域に a または b から可視な点 ($\in\mathbf{V}_{t+m}$) が存在しない，または，a の右かつ b の下の領域に a または b から可視な点 ($\in\mathbf{V}_{t+m}$) が存在しないならば，(a,b) は $\mathbf{G}_f$に含まれる．(ii) $a\in\mathbf{V}_{t+m}$，$b\in\mathbf{V}_p$のときは，b が a の投影点であるならば (a,b) は $\mathbf{G}_f$に含まれる．

図 8にフロー導出グラフの例を示す．詳細な証明は省略するが，フロー導出グラフについては，文献 [1] と同様の定理が成立する．すなわち，容量判定グラフ $\mathbf{G}_c=(\mathbf{V}_c,\mathbf{E}_c)$ が与えられたとき，フロー導出グラフ $\mathbf{G}_f=(\mathbf{V}_c,\mathbf{E}_f)$ は $O(|\mathbf{E}_f|)$ 時間で作成できる．また，$\mathbf{G}_f$は平面グラフであり，辺の数 $|\mathbf{E}_f|$ は $O(|\mathbf{V}_c|)$ である．

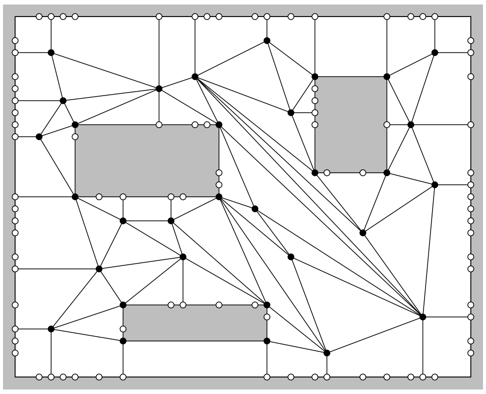


図 8: フロー導出グラフの例

文献 [1] と同様に，$\mathbf{G}_f$の全ての辺についてフローを与えさえすれば $\mathbf{G}_c$の全ての辺についてフローを求めることができる．(a,b) を $\mathbf{E}_c$に含まれる辺とする．$a\in\mathbf{V}_{t+m}$，$b\in\mathbf{V}_p$である場合は $(a,b)\in\mathbf{E}_f$なのでフローはすでに与えられていることになる．$a,b\in\mathbf{V}_{t+m}$の場合は，文献 [1] と同様にフリップを行うことによりそれぞれの辺のフローを $O(1)$ の実行時間で求めることができる．このとき実行時間の合計は $O(|\mathbf{E}_c|)$ である．これにより $\mathbf{G}_c$の辺について容量とフローの比較を行うことにより配線可能性検証が行うことができる．

6 まとめ

矩形配線禁止領域を考慮に入れた配線モデルについて，容量判定グラフとフロー導出グラフを定義した．また，容量判定グラフを効率良く求めるアルゴリズムを与えた．これにより，今回の配線モデルについても効率的な配線可能性検証が可能となった．

今後の課題としては，概略配線発見手法と配線可能性検証手法を組み合わせた配線システムを開発することなどがあげられる．

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博士に感謝致します．

参考文献

[1] 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” 情処研報 95-DA-78, pp.157–162 (1995).

[2] R.Cole and A.Siegel : “River routing every which way, but loose,” Proc. of 25th IEEE Symp. on Foundations of Computer Science, pp.65–73 (1984).

[3] C.Leiserson and F.Maley : “Algorithms for routing and testing routability of planar VLSI layouts,” Proc. of 17th ACM Symp. on Theory of Computing, pp.69–78 (1985).

[4] R.Guting, O.Nurmi and T.Ottmann : “Fast algorithms for direct enclosures and direct dominances,” Journal of Algorithms 10, pp.170–186 (1989).