—— 論　文 ——

UDC 621.3.049.75/.77:658.512.24

一層一行配線問題の片側トラック数の最小性

准員　真鍋　義文[†]　　正員　萩原　兼一[†]　　正員　都倉　信樹[†]

The Minimum Track Number of the Narrower Channel in the Single-Row Single-Layer Routing

Yoshifumi MANABE[†], *Associate Member*, Ken'ichi HAGIHARA[†] *and* Nobuki TOKURA[†], *Regular Members*

あらまし　一層一行配線問題は，一直線上に並んだ頂点集合間のある接続関係（ネットの集合）$\mathcal{N}$が与えられたときに，交差なしで接続関係を満たす配線（実現R）のうちで，配線に必要な上下のチャネルのトラック数$U(R)$, $L(R)$の小さい（(1)$U(R)+L(R)$, (2)max（$U(R)$, $L(R)$），(3)min（$U(R)$, $L(R)$）などをそれぞれ最小にする）実現を求める問題であり，プリント基板上やＬＳＩの配線設計における重要な問題である．(1)の時間複雑度はまだ未解決，(2)はＮＰ完全である．本論文では(3)，すなわちトラック数の大きくない方のチャネルのトラック数tを最大にする実現R_mを求める問題を議論する．$\mathcal{N}$の交錯度$\alpha(\mathcal{N})$という尺度を新たに導入し，この値がtの最小値になることを示す．さらに，$\alpha(\mathcal{N})$およびR_mを求めるアルゴリズムを示し，それぞれの計算時間が最悪時でもネット数と頂点数の積に比例することを示す．この問題は与えられたネット集合を上下トラック数が制限されたときに多層で実現するときに必要な層数を求めるためにも基本的な問題である．

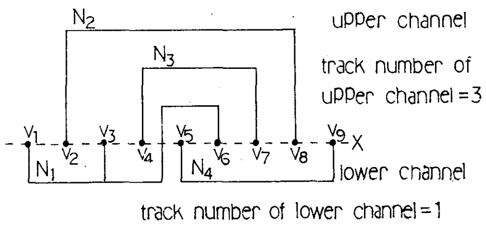
1. まえがき

大規模集積回路の設計における問題の一つに配線設計がある．プリント基板の配線設計やマスタスライス方式によるＬＳＩ設計において一層一行配線問題(single-row single-layer routing)は重要である．

一層一行配線問題は，図１に示すように，与えられたネット集合（net set）に対して，配線に必要な面積の小さな実現（realization）を求める問題である．配線に必要な面積を与える尺度として，上下のチャネル（upper channel, lower channel）の配線の数（トラック数，track number）を考える．本論文では，上チャネルと下チャネルの対称性により，一般性を失うことなく上チャネルのトラック数が下チャネルのトラック数以上と仮定する．

ネット集合が与えられたときに，含まれるネットの数をmとすると，**2.2**で述べるように，m個のネットの順列はそのネット集合のある実現に対応している．順列は$m!$通り存在し，上チャネルと下チャネルの対称性を考慮しても，相異なる実現が$m!/2$個存在するネット集合があり得る．従って，$m!/2$個の実現のうちで，配線に必要な面積すなわち上下のチャネルのトラック数の和を最小にする実現を求める問題（総和トラック数最小問題）が工学的に興味深い．しかし，この問題を解く効率の良いアルゴリズムは現在のところ発見されていない．

次に重要な問題は上チャネルのトラック数を最小にする実現を求める問題（最大トラック数最小問題）である．この実現に関する必要十分条件が求められている[3]が，この問題はＮＰ完全である[1]．



$V=\{v_1, v_2, \cdots, v_9\}$

$\mathcal{N}=\{N_1, N_2, N_3, N_4\}$

$N_1=\{v_1, v_3, v_6\}$　$N_2=\{v_2, v_8\}$

$N_3=\{v_4, v_7\}$　$N_4=\{v_5, v_9\}$

図１　一層一行配線問題とその解の例
Fig. 1–An example of single-row single-layer routing problem and its solotion.

† 大阪大学基礎工学部情報工学科，豊中市
Faculty of Engineering Science, Osaka University, Toyonaka-shi, 560 Japan

現在の技術では，配線の密度にはあまり大きくない限界がある．従って，上下のチャネルにおけるトラック数が制限されたときにそのトラック数で実現可能かどうかを判定するのは重要な問題（トラック数制限問題）である．上下のトラックのトラック数がいずれもk以下に制限されたときの配線可能性を決定する$O((2k)!\cdot n\cdot k\cdot \log k)$時間（ただし$n$はネット集合中の頂点数）のアルゴリズムが与えられている[2]．もしも制限されたトラック数で配線が不可能であれば，そのネット集合は多層に分割して配線を行なわなければならず，必要な層数に対する見積りを効率良く行うことが重要である．

本論文では，一層一行配線問題における下チャネルのトラック数を最小にする実現を求める問題（片側トラック数最小問題）を議論する．下チャネルのトラック数の最小値および下チャネルのトラック数が最小の実現を求める効率の良いアルゴリズムを示す．この問題は多層で実現するときの必要な層数を求めるためにも基本的な問題である．

また，今までに知られている，指数的時間を用いずに求められる，上下のチャネルのトラック数の合計の上限は本稿の結果を用いて改善される．

2 一層一行配線問題

2.1 一層一行配線問題の定義

直線X上に並んだn個の頂点の集合をVとする．Vの頂点間の順序関係$<$を，頂点vがv'の左にあるときに$v<v'$と定義する．Vのn個の頂点をこの関係で順序づけし，$V=\{v_1, v_2, \cdots, v_n\}$（$v_i<v_{i+1}$, $i=1, 2, \cdots, n-1$）とおく．ネットNはVの部分集合であり，互いに接続されるべき頂点の集合を表す．ネットNの頂点の個数を$|N|$で表す．$|N|\leqq 1$のネットは配線上無意味なので$|N|\geqq 2$とする．ネット集合$\mathcal{N}=\{N_1, N_2, \cdots, N_m\}$は$n$個の頂点間の接続関係を表す．ただし，$i\neq j$のとき$N_i\cap N_j\neq\phi$であれば$N_i$と$N_j$を一つのネットに併合できるので$N_i\cap N_j=\phi$とする．

ネットN_iを実現するとは，N_iの各頂点間を，直線Xに平行か垂直な線分を用いてつなぐことである．ただし，文献(6)と同様の配線条件をつけ加える．条件は，直線Xに平行な方向のジグザグな道のとり方は許さないことである．すなわち，図2(a)に示すように，直線Xに垂直な任意の直線と交わる線分が2本以上あってはならない．しかし，直線Xに垂直な方向のジグザグな道のとり方は許される（図2(b)）．

ネット集合$\mathcal{N}$を実現するとは，相異なるネットを実現する線分は交わらないという条件のもとで，$\mathcal{N}$に属するm個のネットを実現することである．

直線Xの上側を上チャネル，下側を下チャネルと呼ぶ．ネット集合$\mathcal{N}$が与えられたとき，その一つの実現Rに対して，各頂点において上チャネルに直線Xと平行にひかれている線分の数の最大値を上トラック数と呼び，$U(R)$と書く．同様に下チャネルに対しては下トラック数を定義し，$L(R)$と書く（前述のように，$U(R)\geqq L(R)$と仮定する）．図1の実現Rに対しては$U(R)=3$，$L(R)=1$である．

一層一行配線問題は．$U(R)$，$L(R)$に関するある条件が与えられたとき，任意のネット集合$\mathcal{N}$に対して，その条件を満足する$\mathcal{N}$の実現Rを効率良く求める問題である．本論文では片側トラック数最小問題，すなわち$L(R)$を最小にする実現Rを求める問題について考察する[4]．

2.2 諸定義

ネットNが頂点v（$\notin N$）をカットするとは，$v_1<v<v_2$となるような2頂点v_1，v_2がNに含まれていることをいう．ネットNが頂点vをカットしているとき，$\mathcal{N}$の実現においてNを実現する，直線Xに平行な線分が頂点vにおいて上チャネルか下チャネルに必ず存在する．ネット集合$\mathcal{N}$における頂点vのカット数を，vをカットする$\mathcal{N}$のネットの数として定義し，$\mathrm{cut}_1(v, \mathcal{N})$で表す．$\mathcal{N}$を実現したとき，$v$を含むネット以外のネットを実現する直線Xに平行な線分で，頂点vにおいて上下のチャネルに存在するものの個数が合計$\mathrm{cut}_1(v, \mathcal{N})$である．ネット集合$\mathcal{N}$におけるネット$N$のカット数$\mathrm{cut}_2(N, \mathcal{N})$を

$$\mathrm{cut}_2(N, \mathcal{N})=\max_{v\in N}\mathrm{cut}_1(v, \mathcal{N})$$

で定義する．ネット集合$\mathcal{N}$のカット数$\mathrm{cut}_3(\mathcal{N})$を，

$$\mathrm{cut}_3(\mathcal{N})=\min_{N\in\mathcal{N}}\mathrm{cut}_2(N, \mathcal{N})$$

で定義する．ただし，$\mathcal{N}=\phi$のときは$\mathrm{cut}_3(\mathcal{N})=0$とす

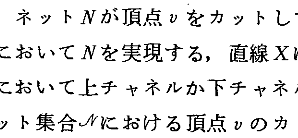


図2 許されない配線(a)と許される配線(b)
Fig. 2–Routing constraint : (a) is not allowed, but (b) is allowed.

る．

ネット集合$\mathcal{N}$に対する頂点集合$T(\mathcal{N})$を

$T(\mathcal{N})=\{ v \mid v\in N,\ N\in\mathcal{N}\}$

で定義する．また，ネット集合$\mathcal{N}$の密集度$mc(\mathcal{N})$を，

$$mc(\mathcal{N})=\max_{v\in T(\mathcal{N})}\mathrm{cut}_1(v,\mathcal{N})+1$$

で定義する．$\mathcal{N}$を実現したとき，各頂点において，上下のチャネルに存在する，（その頂点を含むネットを加えた）各ネットを実現する，直線Xに平行な線分の数の最大値が$mc(\mathcal{N})$である．ネット集合$\mathcal{N}$に対して，$\mathcal{N}$の交錯度（tangle number）$\alpha(\mathcal{N})$を，

$$\alpha(\mathcal{N})=\max_{\mathcal{N}'\subseteq\mathcal{N}}\mathrm{cut}_3(\mathcal{N}')$$

で定義する．任意のネット集合$\mathcal{N}$に対して，$mc(\mathcal{N})$の定義より，任意の頂点$v\in T(\mathcal{N})$に対して，

$$\mathrm{cut}_1(v,\mathcal{N})\leqq mc(\mathcal{N})-1$$

が成立つ．従って，任意のネット$N\in\mathcal{N}$に対して，

$$\begin{aligned}\mathrm{cut}_2(N,\mathcal{N})&=\max_{v\in N}\mathrm{cut}_1(v,\mathcal{N})\\&\leqq mc(\mathcal{N})-1\end{aligned}$$

である．さらに，任意のネット集合$\mathcal{N}'\subseteq\mathcal{N}$に対して，

$$\begin{aligned}\mathrm{cut}_3(\mathcal{N}')&=\min_{N\in\mathcal{N}'}\mathrm{cut}_2(N,\mathcal{N}')\\&\leqq\min_{N\in\mathcal{N}'}\mathrm{cut}_2(N,\mathcal{N})\\&\leqq mc(\mathcal{N})-1\end{aligned}$$

が成立つ．従って，

$\alpha(\mathcal{N})\leqq mc(\mathcal{N})-1$

が成立つ．

図1のネット集合$\mathcal{N}$に対して，$\mathrm{cut}_1(v_5,\mathcal{N})=3$，$\mathrm{cut}_2(N_3,\mathcal{N})=2$，$mc(\mathcal{N})=4$，$\alpha(\mathcal{N})=1$である．

〔ネット集合の区間グラフ的表現〕

文献(3)に示されているように，ネットをその左端の頂点から右端の頂点までの区間グラフ（interval graph）に対応させたとする．ネット集合$\mathcal{N}$に対し，$\mathcal{N}$のすべてのネットに対応する区間グラフを任意の順序で上から下へ並べる．図1のネット集合$\mathcal{N}$に対して区間グラフを対応させて並べた例を図3(a)に示す．図3(a)を頂点の順序および区間グラフの上下関係を保ちながら変形すると図3(b)のようにこのネット集合に対する一つの実現が得られる（図3(a)の破線が図3(b)の直線Xに対応している）．すなわち，ネットに対応する区間グラフの順序づけはネット集合の実現に対応している．この順序づけに対して，ある頂点vにおいて区間グラフ上で垂直な直線をひく．ある区間グラフがこの直線と頂点vより上で（下で）交わるとき，この区間グラフは頂点vの上に（下に）あるという．図3の例ではN_3に対応する区間グラフは頂点v_5の上，v_6の下にある．ある区間グラフが頂点vの上に（下に）あるとき，この順序づけに対応する実現において，対応するネットが頂点vの上に（下に）存在する．ネットNが頂点vをカットしていることは，ネットNに対応する区間グラフが頂点vの上または下にあることに対応している．ネット集合$\mathcal{N}$における頂点vのカット数は頂点vの上または下にある区間グラフの数である．たとえば図3(a)の例では$\mathrm{cut}_1(v_5,\mathcal{N})=3$，$\mathrm{cut}_1(v_4,\mathcal{N})=2$である．この順序づけにおいて，任意のネット$N$のある頂点$v$の上下それぞれに他のネットの区間グラフが$x$個，$y$個だけ存在したとすると，対応する実現においても頂点vで上下のチャネルにN以外ネットがそれぞれx個，y個存在する．ネットNを実現する線分を頂点vにおいて上チャネルにひけば上下のチャネルに直線Xに平行な線分はそれぞれ$(x+1)$本，y本存在し，下チャネルにひけばx本，$(y+1)$本になる．このどちらの実現も可能である．

〔既知の結果〕

本論文で用いられる，既知の結果を以下に記す．

〔命題1〕[6] 任意のネット集合$\mathcal{N}$に対して，$U(R)\leqq mc(\mathcal{N})-1$，$L(R)\leqq mc(\mathcal{N})-1$，である実現$R$が存在する．

〔命題2〕[3] 任意のネット集合$\mathcal{N}$の任意の実現Rに

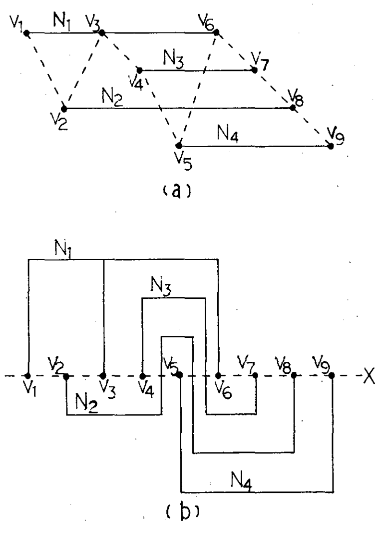


図3　区間グラフの順序づけと，それに対応する実現
Fig. 3–An ordering of interval graphs and its corresponding realization.

対して，$U(R)+L(R)\geqq mc(\mathcal{N})$が成立つ．

3.3 片側トラック数の最小値

ネット集合$\mathcal{N}$に対して，下チャネルのトラック数が$\alpha(\mathcal{N})-1$以下の実現は存在しない．また，下チャネルのトラック数が$\alpha(\mathcal{N})$で実現可能である．以下にその証明を記す．

〔定理１〕 任意のネット集合$\mathcal{N}$に対して，$L(R)<\alpha(\mathcal{N})$であるような実現Rは存在しない．

（証明） 背理法で証明する．$L(R)<\alpha(\mathcal{N})$であるような実現Rが存在したと仮定する．ところで，$\alpha(\mathcal{N})$の定義より，$\mathrm{cut}_3(\mathcal{N}')=\alpha(\mathcal{N})$, $\mathcal{N}'\subseteq\mathcal{N}$であるようなネット集合$\mathcal{N}'$が存在する．$\mathcal{N}'$は$\mathcal{N}$の部分集合であるので$\mathcal{N}$の実現に要するトラック数以下のトラック数で実現可能である．その実現の一つをR'とする．このとき，$L(R')\leqq L(R)<\alpha(\mathcal{N})$である．

R'に対応する区間グラフの順序づけで一番上のネットN_1を考える．$\mathcal{N}'$におけるN_1のカット数$\mathrm{cut}_2(N_1, \mathcal{N}')$は定義より$\alpha(\mathcal{N})$以上なので$N_1$の頂点のうちの少なくとも一つ$v$では$\mathcal{N}'$におけるカット数$\mathrm{cut}_1(v, \mathcal{N}')$が$\alpha(\mathcal{N})$以上である．その頂点$v$では，下に$\alpha(\mathcal{N})$以上の区間グラフが存在する（図4(a)）．従って，R'では頂点vにおいて下チャネルに$\alpha(\mathcal{N})$以上のネットが存在することになるので，すなわち$L(R')\geqq\alpha(\mathcal{N})$となり（図4(b)），仮定に反する．従って，$L(R)<\alpha(\mathcal{N})$である実現$R$は存在しない．　（証明終）

〔補題１〕 任意のネット集合$\mathcal{N}$に対し，ある非負整数aが存在して$L(R)=\mathrm{a}$である実現Rが存在するとき，

$$L(R')=\mathrm{a},\ U(R')\leqq mc(\mathcal{N})-\min(\mathrm{a},1)$$

である$\mathcal{N}$の実現R'が存在する．

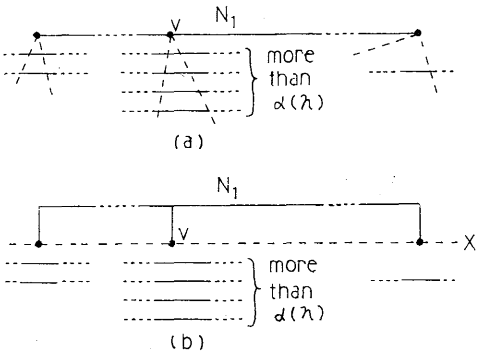


図4　区間グラフの順序づけにおいて一番上の区間グラフN_1とそれに対応する実現

Fig. 4–The top interval graph N_1 in an ordering of interval graphs and its corresponding realization.

（証明） 直線Xに平行なジグザグな道のとり方を許さないので，任意の頂点において直線Xに平行な線分の数は$mc(\mathcal{N})$以下である．従ってすべての実現に対して

$$U(R)\leqq mc(\mathcal{N}),\quad L(R)\leqq mc(\mathcal{N})$$

である．

与えられたネット集合$\mathcal{N}$に対して，

(i) $L(R)=\mathrm{a}=0$である実現Rが存在するとき
$U(R)\leqq mc(\mathcal{N})$より$R$を$R'$とすればよい．

(ii) $L(R)=\mathrm{a}\ (\mathrm{a}>0)$である実現$R$が存在して$U(R)=mc(\mathcal{N})$のとき（$U(R)\leqq mc(\mathcal{N})-1$のときは$R$を$R'$とすればよい）

その実現Rである頂点vにおいて上チャネルに，$mc(\mathcal{N})$本の直線Xに平行な線分が存在するとき，この頂点vにおいて下チャネルに線分が存在することはない（直線Xに平行な線分の数は$mc(\mathcal{N})$以下であるから）．従って上チャネルにおいて，一番下の線分（直線Xの最も近くにある線分）を下チャネルに移すことができる．このようにして求められる実現R'は

$$L(R')=L(R),\ U(R')=mc(\mathcal{N})-1$$

である．　（証明終）

〔片側トラック数一定値以下の実現を与えるアルゴリズム〕

ネット集合$\mathcal{N}$と非負整数kが与えられたとき，以下に示すアルゴリズム[2]を用いて$\mathcal{N}$のネットに順番をつける．このときアルゴリズム中の(2)において，空でないSETに対して常に$\mathrm{cut}_2(N, \mathrm{SET})\leqq k$であるネット$N$が存在すれば（以下，この条件を正常終了条件と呼ぶ），すべてのネットを(4)で取り除くことができる．このとき配列orderには取り除かれた順番にネットが並んでいる．この順番を区間グラフの順序づけとみなしたとき，対応するネット集合$\mathcal{N}$の実現が得られる．

〔アルゴリズム１〕 下チャネルのトラック数k以下の実現を求めるアルゴリズム

／＊　入力として与えられるもの
　　$\mathcal{N}$：ネット集合
　　m：$\mathcal{N}$に含まれるネットの個数
　　k：下チャネルのトラック数の上限
　変数の内容
　　SET：ネット集合（$\mathcal{N}$の部分集合）
　　order[1..m]：ネットの順番を入れる配列
＊／

[1]　SET：$=\mathcal{N}$；

〔2〕 <u>for</u> j ：＝1 <u>to</u> m <u>do</u>

<u>begin</u>

(1) SETの各ネットNに対してcut_2 (N, SET) を計算する；

(2) <u>if</u> すべてのネット$N \in$ SETに対して cut_2 (N, SET) $> k$

<u>then</u> 失敗 (下チャネルのトラック数k以下の実現は存在しない)

<u>else</u> cut_2 (N, SET) $\leqq k$であるネット$N \in$ SETのうちの1つをN_iとする；

(3) order 〔j〕：＝N_i ；

(4) SET：＝SET－$\{N_i\}$

<u>end</u>

(注) orderの内容は(2)でのネットN_iの選び方により一意に定まらないが，得られたorderはすべて下チャネルのトラック数がk以下の実現の区間グラフ的表現に対応している．

〔補題2〕 ネット集合$\mathcal{N}$と非負整数kが与えられたとき，上記のアルゴリズムで正常終了条件が成立すれば，$L(R) \leqq k$である$\mathcal{N}$の実現Rが存在する．

(証明) 正常終了条件が成立すれば，上記のアルゴリズムで任意のネットNはcut_2 (N, SET) $\leqq k$となったときに(4)で取り除かれる．(4)でNが取り除かれる直前を考えると，cut_2 (N, SET) $\leqq k$より，Nに属する各頂点v_iに対してcut_1 (v_i, SET) $\leqq k$である．配列orderに求められる区間グラフの順序づけにおいてNより下にくるネットはSET-$\{N\}$の要素に限られる．従ってNの各頂点v_iの下に存在する，他のネットの区間グラフの個数はcut_1 (v_i, SET) であるのでk以下である (図5(a))．

任意の頂点に対して，下に存在する他のネットの区間グラフの数がk以下なので，この順序づけに対応する，$L(R) \leqq k$である実現Rが存在する (図5(b))．

(証明終)

〔定理2〕 任意のネット集合$\mathcal{N}$に対して，$L(R)= \alpha(\mathcal{N})$である$\mathcal{N}$の実現$R$が存在して，

$$\max(\alpha(\mathcal{N}),\ mc(\mathcal{N})-\alpha(\mathcal{N})) \leqq U(R)$$
$$\leqq mc(\mathcal{N})-\min(\alpha(\mathcal{N}),\ 1)$$

を満足する．

(証明) アルゴリズム1において，$k=\alpha(\mathcal{N})$としたときに正常終了条件が成立すること，すなわちアルゴリズムの(2)において空でないSETに対して，常にcut_2 (N, SET) $\leqq \alpha(\mathcal{N})$であるネット$N \in$ SETが存在することを背理法により説明する．

ある空でない$\mathcal{N}$の部分集合$\mathcal{N}'$が存在して，すべてのネット$N \in \mathcal{N}'$が$\mathrm{cut}_2(N, \mathcal{N}') > \alpha(\mathcal{N})$を満足していると仮定する．このとき，ネット集合のカット数の定義より，$\mathrm{cut}_3(\mathcal{N}') > \alpha(\mathcal{N})$となる．$\mathcal{N}'$は$\mathcal{N}$の部分集合なので，これは$\alpha(\mathcal{N})$の定義

$$\alpha(\mathcal{N}) = \max_{\mathcal{N}' \subseteq \mathcal{N}} \mathrm{cut}_3(\mathcal{N}')$$

に反する．

従って補題2より，$L(R) \leqq \alpha(\mathcal{N})$である$\mathcal{N}$の実現$R$が存在する．$L(R) \leqq \alpha(\mathcal{N})$である実現は定理1より存在しないので，この実現は$L(R)= \alpha(\mathcal{N})$である．

この実現Rの上チャネルのトラック数$U(R)$に対しては，

$$U(R) \geqq L(R) = \alpha(\mathcal{N})$$

であり，補題1より，

$$U(R) \leqq mc(\mathcal{N}) - \min(\alpha(\mathcal{N}),\ 1)$$

が成立する．また，命題2より，

$$mc(\mathcal{N}) \leqq U(R) + L(R) = U(R) + \alpha(\mathcal{N})$$

が成立する．すなわち，この実現Rに対しては，

$$\max(\alpha(\mathcal{N}),\ mc(\mathcal{N}) - \alpha(\mathcal{N})) \leqq U(R)$$
$$\leqq mc(\mathcal{N}) - \min(\alpha(\mathcal{N}),\ 1)$$

である． (証明終)

定理2より，任意のネット集合$\mathcal{N}$に対して，

$$L(R) = \alpha(\mathcal{N})$$
$$U(R) \leqq mc(\mathcal{N}) - \min(\alpha(\mathcal{N}),\ 1)$$

である実現Rが存在する．$\alpha(\mathcal{N}) \leqq mc(\mathcal{N}) - 1$が成立つので，定理2は命題1より強い結果である．

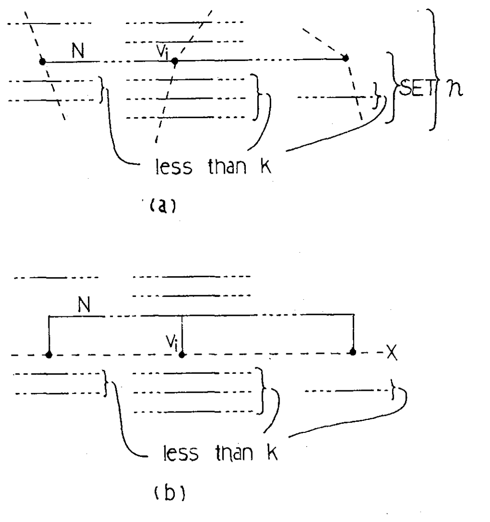


図5 SETからのネットNの除去
Fig. 5－Deletion of net N from SET.

〔例〕

$V=\{v_1, v_2, \cdots, v_{11}\}$

$\mathcal{N}=\{N_1, N_2, N_3, N_4, N_5\}$

$N_1=\{v_1, v_4, v_8\}$, $N_2=\{v_2, v_9\}$

$N_3=\{v_3, v_6\}$, $N_4=\{v_5, v_{11}\}$

$N_5=\{v_9, v_{10}\}$

この例に対して，$\max_{\mathcal{N}'\subseteq\mathcal{N}} \mathrm{cut}_3(\mathcal{N}')$を与える$\mathcal{N}'$の一つとして，$\mathcal{N}''=\{N_1, N_2, N_4, N_5\}$があり，$\alpha(\mathcal{N})=\mathrm{cut}_3(\mathcal{N}'')=2$である．従ってこのネット集合$\mathcal{N}$には$L(R)=2$である実現$R$が存在する．実現$R$の一例を図6に示す．このネット集合$\mathcal{N}$に対して$mc(\mathcal{N})=4$であるので，命題1では$U(R)\leqq 3$，$L(R)\leqq 3$である実現$R$が存在することしか保証されない．また，文献(5)のアルゴリズムでも$U(R)\leqq 3$，$L(R)\leqq 3$である実現Rが存在することのみが求められるにすぎない．

2.4 総和トラック数問題

定理1，2より，ネット集合$\mathcal{N}$を実現したときの総和トラック数に対して次の定理が成立つ．

〔定理3〕 任意のネット集合$\mathcal{N}$に対し，$U(R)+L(R)$を最小にする$\mathcal{N}$の実現Rに対して

$$\max(mc(\mathcal{N}), 2\alpha(\mathcal{N}))\leqq U(R)+L(R)$$
$$\leqq \alpha(\mathcal{N})+mc(\mathcal{N})-\min(\alpha(\mathcal{N}), 1)$$

が成立つ．

（証明） 命題2より，$U(R)+L(R)\leqq mc(\mathcal{N})$が成立つ．定理1より，$U(R)+L(R)\geqq 2\alpha(\mathcal{N})$が成立つ．また，定理2より，$U(R)+L(R)\leqq \alpha(\mathcal{N})+mc(\mathcal{N})-\min(\alpha(\mathcal{N}), 1)$が成立つ． （証明終）

定理3より，ネット集合$\mathcal{N}$の任意の実現Rに対して，$U(R)+L(R)\geqq\max(mc(\mathcal{N}), 2\alpha(\mathcal{N}))$が成立つ．これは命題2より強い結果である．

2.5 ネット集合の交錯度を求めるアルゴリズム

本節ではネット集合$\mathcal{N}$が与えられたときに$\alpha(\mathcal{N})$を求める$O(\sum_{v\in T(\mathcal{N})} \mathrm{cut}_1(v, \mathcal{N}))$時間のアルゴリズムを示す．

ネット集合$\mathcal{N}$と非負整数kに対して，$S_k(\mathcal{N})$を$\mathcal{N}'\subseteq\mathcal{N}$かつ$\mathrm{cut}_3(\mathcal{N}')\geqq k$である要素数最大のネット集合として定義する．$S_k(\mathcal{N})$は以下の性質を満たす．

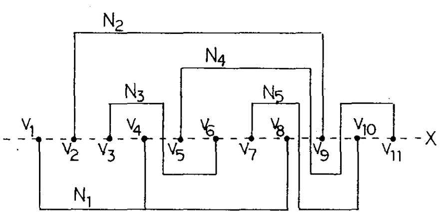


図6 $L(R)=\alpha(\mathcal{N})=2$である実現$R$

Fig. 6-A realization R such that $L(R)=\alpha(\mathcal{N})=2$.

(1) $\mathcal{N}$, kが与えられたとき$S_k(\mathcal{N})$は一意に定まる．なぜならば，$\mathcal{N}'\subseteq\mathcal{N}$かつ$\mathrm{cut}_3(\mathcal{N}')\geqq k$である要素数最大の相異なるネット集合$\mathcal{N}_1$，$\mathcal{N}_2$が存在したと仮定すると$\mathcal{N}_1\cup\mathcal{N}_2\subseteq\mathcal{N}$かつ$\mathrm{cut}_3(\mathcal{N}_1\cup\mathcal{N}_2)\geqq k$であり，$\mathcal{N}_1\cup\mathcal{N}_2\supsetneqq\mathcal{N}_1$，$\mathcal{N}_1\cup\mathcal{N}_2\supsetneqq\mathcal{N}_2$である．これは$\mathcal{N}_1$,$\mathcal{N}_2$が要素数最大であることに反する．

(2) $S_k(\mathcal{N})\supseteq S_{k+1}(\mathcal{N})$（$k=0, 1, \cdots$）が成立つ．なぜならば$S_k(\mathcal{N})\not\supseteq S_{k+1}(\mathcal{N})$と仮定すると，$\mathrm{cut}_3(S_{k+1}(\mathcal{N}))\geqq k+1$より，$\mathrm{cut}_3(S_k(\mathcal{N})\cup S_{k+1}(\mathcal{N}))\geqq k$であり，$S_k(\mathcal{N})\cup S_{k+1}(\mathcal{N})\supsetneqq S_k(\mathcal{N})$である．これは$S_k(\mathcal{N})$の要素数の最大性に反する．

(3) $S_{\mathrm{cut}_3(\mathcal{N})}(\mathcal{N})=\mathcal{N}$である．

(4) $i\leqq\alpha(\mathcal{N})$のとき$S_i(\mathcal{N})$は存在するが，$j>\alpha(\mathcal{N})$のとき$S_j(\mathcal{N})$は存在しない．これは$\alpha(\mathcal{N})$の定義よりも明らかである．

性質(4)は，$S_i(\mathcal{N})$が存在し，$S_{i+1}(\mathcal{N})$が存在しないようなiが$\alpha(\mathcal{N})$である，と換言できる．

上記の性質より，以下に示すアルゴリズムで$\alpha(\mathcal{N})$は求められる．

(1) $k:=\mathrm{cut}_3(\mathcal{N})$; $S_k(\mathcal{N}):=\mathcal{N}$;

(2) <u>repeat</u>

　　$S_k(\mathcal{N})$から$\mathrm{cut}_2(N, S_k(\mathcal{N}))\leqq k$のネット$N$をすべて取り除くことにより$S_{k+1}(\mathcal{N})$を求める；

　　$k:=k+1$

　<u>until</u> $S_k(\mathcal{N})=\phi$

(3) /* repeat-until節から出たときのkの値-1が$\alpha(\mathcal{N})$である． */

このアルゴリズムを詳細化したものを以下に示す．

〔アルゴリズム2〕 $\alpha(\mathcal{N})$を求めるアルゴリズム

/* 入力として与えられるもの

　n：Vの頂点数

　m：$\mathcal{N}$のネット数

　$c[i]$：ネットN_iの頂点数

　$v[i, j]$：ネットN_iのj番目の頂点．ただし$v[i, j]<v[i, j+1]$

各変数の内容

　net[i]：頂点iを含むネットの番号

　count：$S_k(\mathcal{N})$の頂点数

　$\mathrm{cut}_1[i]$：$\mathrm{cut}_1(v_i, S_k(\mathcal{N}))$

　$\mathrm{cut}_2[i]$：$\mathrm{cut}_2(N_i, S_k(\mathcal{N}))$

　$q[j]$：$\mathrm{cut}_2(N_i, S_k(\mathcal{N}))=j$であるネットの集合

　cutnode[i]：ネットN_iがカットする頂点の集合

　nodes[i, j]：$\mathrm{cut}_1(v, S_k(\mathcal{N}))\geqq j$であ

```
            るネット N_i の頂点数      */
[1]      /*初期設定           */
(1) for i := 1 to m do
      begin
        net[i] := 0; cut_2[i] := 0;
        nodes[i, 1] := 0;
        nodes[i, 0] := c[i];
        q[i], cotnode[i]を空にする;
      end;
    for i := 1 to m do cut_1[i] := 0;
(2) for i := 1 to m do
      for j := 1 to c[i] do
        net[v[i, j]] := i;
      /* net[i]をセットする      */
(3) for i := 1 to m do
      for j := v[i, 1] to
              v[i, c[i]] do
        if net[j] ≠ i then  /* ネット
          iが頂点jをカットしているとき
                                      */
         begin
           cut_1[j]を1ふやす;
           cutnode[i]にjを加える;
           nodes[net[j], cut_1[j]]を1
           ふやす;
           if nodes[net[j], cut_1[j]] = 1
            then nodes[net[j], cut_1[j]
                        +1] := 0
         end;
(4) k := m;  count := m;
(5) for i := 1 to m do
     begin
      cut_2[i] := max (cut_1[v[i, 1]],
        …, cut_1[v[i, c[i]]]);
      q[cut_2[i]]にiを加える;
      k := min (k, cut_2[i])
     end; /*  k = cut_3(𝒩)である.     */
[2]      /*  主ルーチン      */
(1) repeat    /*  S_k(𝒩)からS_{k+1}(𝒩)を求める
                                      */
      while q[k] ≠ ∅ do
        begin  /*  cut_2(N, S_k(𝒩)) ≤ kのネ
                    ットを取り除く      */
(2)        q[k]から一つのネットを取り除き，それ
           をNに入れる;
           count を1へらす;
           while cutnode[N] ≠ φ do
            begin  /*  Nがカットしている頂点
                       のカット数をへらす  */
(3)          cutnode[N]から一つの頂点を取り
             除き，それをXに入れる;
(4)          if cut_2[net[X]] > k  then
              begin
                nodes[net[X], cut_1[X]]
                  を1へらす;
                cut_1[X]を1へらす;
                if nodes[net[X], cut_2[net
                  [X]]=0 then
                 begin
                  net[X]をq[cut_2[net[X]]]
                  からq[cut_2[net[X]]-1]
                  へ移す;
                  cut_2[net[X]]を1へらす;
                 end
              end
            end
        end;
      k := k + 1
    until  count = 0;
    /*  α(𝒩) = k − 1である          */
```

〔$\alpha(\mathcal{N})$を求めるアルゴリズムの時間計算量〕

上記のアルゴリズムの計算時間は以下のようになる．ただし，ここでは簡単化のためにVのすべての頂点はいずれかのネットに含まれているとする．［1］の初期設定部は，(1)が$O(m+n)$，(2)が$O(n)$，(3)は外側のif文の実行回数が$O(n+\sum_{v\in T(\mathcal{N})}\mathrm{cut}_1(v,\mathcal{N}))$，(5)は$O(n)$であるので全体としては$O(\sum_{v\in T(\mathcal{N})}\mathrm{cut}_1(v,\mathcal{N}))$である．

［2］の主ルーチンでは(1)のrepeat−until節から出るまでにすべてのネットが(2)で取り出される．取り出された各ネットに対して(3)でそのネットがカットしているすべての頂点を取り出す．従って(4)以下の文は$O(\sum_{v\in T(\mathcal{N})}\mathrm{cut}_1(v,\mathcal{N}))$回実行される．

全体を通じての計算時間は$O(\sum_{v\in T(\mathcal{N})}\mathrm{cut}_1(v,\mathcal{N}))$である．各頂点のカット数は高々$m-1$である．従って計算時間は最悪時$O(m\cdot n)$である．

また，$L(R)=\alpha(\mathcal{N})$である実現Rを求めるアルゴリズムは上記のアルゴリズムを次のように変更することに

より得られる．新たな変数 order［i］を導入し，主ルーチンの(2)で取り除いたネットを順番に orderに入れる．このとき，この順序づけに対応する実現は$L(R)=\alpha(\mathcal{N})$を満たす．従って，$L(R)=\alpha(\mathcal{N})$を満たす実現Rも$O(\sum_{v\in T(\mathcal{N})}\mathrm{cut}_1(v,\mathcal{N}))$の計算時間で求めることができる．

〔定理 4〕 ネット集合$\mathcal{N}$に対し，$\alpha(\mathcal{N})$の計算および，$L(R)=\alpha(\mathcal{N})$である実現Rを求めることはいずれも$O(\sum_{v\in T(\mathcal{N})}\mathrm{cut}_1(v,\mathcal{N}))$の計算時間でできる．

3. むすび

本論文では一層一行配線問題の片側トラック数の最小値について述べた．残された問題としては，

(1) ネット集合$\mathcal{N}$を下チャネルのトラック数$\alpha(\mathcal{N})$で実現するときに上チャネルのトラック数を最小にするアルゴリズム

(2) ネット集合を多層で実現するために$\mathcal{N}_1, \mathcal{N}_2, \cdots, \mathcal{N}_k$に分割して$\max_{1\le i\le k}\alpha(\mathcal{N}_i)$を最小にする問題

等が考えられる．一層一行配線問題に対しては，この他にもまだ解かれていない問題（たとえば，総和トラック数最小問題）が多く，それらは今後の課題である．

謝辞 本研究にあたり多くの有益な御助言を頂きました大阪大学工学部白川功助教授ならびに築山修治博士および同基礎工学部柏原敏伸講師に感謝致します．また，御協力頂きました和田幸一博士，中内伸二氏，増澤利光氏，池田光二氏に感謝致します．なお，本研究の一部は文部省科研費（No. 57750304）によった．

文 献

(1) Arnold, P. B. : "Complexity Results Circuit Layout on Double-Sided Printing Circuit Boards", Under Graduate Thesis Submitted to the Department of Applied Mathematics, Harvard University (May 1982).

(2) Heinisch, J. : "Aiming at a General Routing Strategy", Proc. of the 18th Design Automation Conference, pp. 668-675 (June 1981).

(3) Kuh, E. S., Kashiwabara, T. and Fujisawa, T. : "On Optimum Single-Row Routing", IEEE Trans. Circuits & Syst., CAS-26, 6, pp. 361-368 (June 1979).

(4) 真鍋，萩原，都倉："一層一行配線問題の片側トラック数の最小性について"，信学技報，CAS83-05（1983-05）．

(5) Raghavan, R. and Sahni, S. : "Single Row Routing", Technical Reporr 80-22, Institute of Technology University of Minnesota, Minneapolis Minnesota (Nov. 1980).

(6) So, H. C. : "Some Theoretical Results on the Routing of Multilayer, Printed-Wiring Boards", Proc. of the 1974 IEEE International Symposium on Circuits and Systems, pp. 296-303 (1974).

（昭和58年9月5日受付，12月19日再受付）