CHANNEL DENSITY MINIMIZATION BY PIN PERMUTATION

Yang Cai and D. F. Wong

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712-1188

TR-92-09 March 1992

Channel Density Minimization by Pin Permutation *

Yang Cai and D.F. Wong
Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

Abstract

We present in this paper a linear time optimal algorithm for minimizing the density of a channel (with exits) by permuting the terminals on the two sides of the channel. This compares favorably with the previously known near-optimal algorithm presented in [6] that runs in super-linear time. Our algorithm has important applications in hierarchical layout design of intergrated circuits. We also show that the problem of minimizing wire length by permuting terminals is NP-hard in the strong sense.

1 Introduction

Channel routing is an important problem in VLSI layout design and has been extensively studied before [2, 9, 12, 22, 26]. Conventional channel routers assume the positions of the terminals on each side of the channel are fixed. However, it is typical in practice that after the placement phase, the positions of the terminals are not completely fixed, and there is some degree of freedom to choose positions for the terminals. This freedom should be used to our advantage to make the subsequent routing task easier and hence obtain reduction in routing area. This type of problems have been studied by many researchers before [3, 4, 5, 8, 11, 13, 14, 15, 16, 17, 18, 20, 23, 24, 25]. We study in this paper the problem of permuting the terminals on the two sides of a channel to minimize the channel density. An important application of routing channels with permutable (interchangeable) terminals is for

*This work was partially supported by the National Science Foundation under grant MIP-8909586, by an IBM Faculty Development Award, and by an ACM SIGDA scholarship.

solving the pin assignment and global routing problem in building-block layout design, as it was shown that the combined pin assignment and global routing problem can be reduced to routing a set of channels with permutable terminals [7, 21].

Several special cases of the problem of minimizing channel density by permuting terminals have been studied before. In [19], an optimal algorithm was presented for the case where the channel has no exit. Another optimal algorithm was given in [7] for basic channels, *i.e.*, channels containing only two-terminal nets with one terminal on each side of the channel. No optimal polynomial time algorithm for the general case was known before. Recently, a near-optimal polynomial time algorithm was presented in [6], which can guarantee to produce results that are within one of the optimal channel density. We show in this paper that the general problem can be solved optimally in polynomial time by presenting a linear time exact algorithm for this problem. We also show that the closely related problem of permuting the terminals so that the channel can be routed in minimum wire length is NP-hard in the strong sense.

The remainder of this paper is organized as follows. Section 2 introduces notations and terminologies. Some preliminary results are presented in Section 3. Section 4 describes the alternate packing algorithm whose use simplifies the presentation of our algorithm. Section 5 presents our algorithm in details. Section 6 proves the optimality of our algorithm. Section 7 gives the NP-hardness result of the wire length minimization problem. Finally, Section 8 concludes the paper with some remarks.

2 Notations and Terminologies

We assume there is a grid superimposed on the layout. All terminals must be placed on grid points and all wires must follow the grid lines. Routing is done in the *two-layer Manhattan model*, in which there are two layers for routing. All horizontal wires are routed in one layer, all vertical wires are routed in the other layer and no wire overlappings are allowed. Wires on different layers are connected by *contact holes* or *vias*. We shall consider a horizontal channel, in which the terminals are placed on its two sides, called the *top* and *bottom* of the channel. A *column* of a channel is a vertical grid line having its endpoints on the two sides of the channel. The columns of a channel are numbered consecutively from left to right. The *length* of a channel is the number of columns in the channel. A net is said to *intersect* a column if it has one terminal on or to the left of the column, one terminal on or to the right of the column, and at least one terminal outside of the column. Two nets are said to *cross* each other if they intersect a common column, and they are said to cross each other at that column. The *local density* of a channel at a column is the total number of nets that intersect the column. The *density* of a channel is the maximum number of nets intersecting a column, *i.e.*, the maximum local density of the channel at a column over all columns of the channel. The *width* of a channel is the minimum number of *tracks* (horizontal grid lines) needed to complete the routing in the two-layer Manhattan model. It is clear that the density of a channel is a lower bound of the width of a channel.

A *channel routing solution* or a *routed channel* is a channel with its connections routed in the two-layer Manhattan model. The *width* of a routed channel is the number of tracks used in the routed channel. There are many channel routing solutions for the same channel, the width of the channel is equal to the minimum width of its channel routing solutions.

With respect to a channel, a net is said to have a *left exit* (respectively, *right exit*) if it has a terminal to the left (respectively, right) of the channel (outside the channel). A channel is said to have an *exit* if it has a net with either a left or a right exit. Each net N_k is specified by an ordered pair (t_k, b_k), where t_k is the number of terminals on the top of the channel (called *top terminals*), and b_k is the number of terminals on the bottom of the channel (called *bottom terminals*). We say N_k is of the *form* (t_k, b_k) and use the notation $N_k = (t_k, b_k)$ A net N_k is called a *positive net* if $t_k \geq b_k$, a *negative net* if $t_k \leq b_k$. If either $t_k = 0$ or $b_k = 0$, then N_k is called a *one-sided net*, otherwise it is a *two-sided* net. If $t_k = 0$, then N_k is a *bottom-sided* net. If $b_k = 0$, then N_k is a *top-sided* net.

The Density_Minimization problem is the problem of permuting the terminals on the two sides of a channel to minimize its density. An instance of the Density_Minimization problem is a 3-tuple $\Pi = (N, L, R)$, where N is the set of nets to be routed, $L, R \subseteq N$ are, respectively, the set of nets in N with left, right exits. We use $n = |N|$ to denote the number of nets in N. For any subset $N' \subseteq N$, we define[1]

$$\begin{aligned} t_{N'} &= \sum_{N_k \in N'} t_k; \\ b_{N'} &= \sum_{N_k \in N'} b_k; \\ t'_{N'} &= \min_{N_k \in N'} t_k; \\ b'_{N'} &= \min_{N_k \in N'} b_k; \\ m_{N'} &= \sum_{N_k \in N'} \max\{t_k, b_k\}. \end{aligned}$$

We may assume that $t_N = b_N = l$, where l is the length of the channel. This assumption is possible because we can always realize it by introducing *trivial nets, i.e.*, nets of the form $(0,1)$ or $(1,0)$ without exits. Trivial nets represent nets requiring no connections. Without loss of generality, we may also assume no net in N has the form $(0,0)$. It is easy to see that no matter how the terminals are permuted, the density of the channel is at least $\max\{|L|, |R|\}$ because all the nets in L cross the leftmost column of the channel, and all the nets in R cross the rightmost column of the channel.

Given an instance $\Pi = (N, L, R)$ of the Density_Minimization problem, we use $B = L \cap R$ to denote the set of nets with both left and right exits, and $M = N - L \cup R$ to denote the set of nets without exits. The set of trivial nets M_T is a subset of M, and we let $M_P = M - M_T$. Also, we let $L^* = L - B$ and $R^* = R - B$. L is said to be *top critical* if

$$t'_{L^*} > l - b_{M_P \cup R^*} = b_{L \cup M_T} = b_{L^*} + b_{B \cup M_T},$$

bottom critical if

$$b'_{L^*} > l - t_{M_P \cup R^*} = t_{L \cup M_T} = t_{L^*} + t_{B \cup M_T},$$

and *critical* if it is either top critical or bottom critical. That R is *top critical*, *bottom critical* and *critical* are similarly defined. Let

$$\delta_L = \begin{cases} 1 & \text{if } L \text{ is critical} \\ 0 & \text{otherwise,} \end{cases}$$

[1] For convenience of presentation, we define

$$\sum_{x \in \phi} x = \min_{x \in \phi} x = \max_{x \in \phi} x = 0.$$

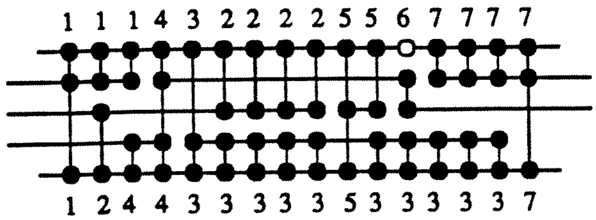


Figure 1: Illustration of definitions

and similarly define δ_R. We also make use of a variable δ, such that $\delta = 1$ if $|L| = |R|$, $\delta_L = \delta_R = 0$, and either

$$b_{M_T \cup B} < (t'_{L^*} - b_{L^*}) + (t'_{R^*} - b_{R^*})$$

or

$$t_{M_T \cup B} < (b'_{L^*} - t_{L^*}) + (b'_{R^*} - t_{R^*}),$$

and $\delta = 0$ otherwise.

Example 1: Consider the channel shown in Figure 1, we have $n = 7$, $l = 16$, $N_1 = (3,1)$, $N_2 = (4,1)$, $N_3 = (1,10)$, $N_4 = (1,2)$, $N_5 = (2,1)$, $N_6 = (1,0)$, $N_7 = (4,1)$, $L = \{N_1, N_2, N_4\}$, $L^* = \{N_1, N_2\}$, $R = \{N_4, N_7\}$, $R^* = \{N_7\}$, $B = \{N_4\}$, $M = \{N_3, N_5, N_6\}$, $M_T = \{N_6\}$, $M_P = \{N_3, N_5\}$. Since $|L| = 2 > |R| = 1$, we have $\delta = 0$. Since $b_{L \cup M_T} = 1 + 1 + 2 + 0 = 4 > t'_{L^*} = 3$ and $t_{L \cup M_T} = 3 + 4 + 1 + 1 = 9 > b'_{L^*} = 1$, L is not critical and hence $\delta_L = 0$. Since $b_{R \cup M_T} = 2 + 1 = 3 < t'_{R^*} = 4$, R is top critical and hence $\delta_R = 1$. □

The significance of L being top critical is that no matter how the terminals are permuted, the density of the resulting channel is at least $|L| + 1$. To see this, consider the net $N_k \in L^*$ with the property that the rightmost column that contains a terminal of N_k is the leftmost among all such rightmost columns of the nets in L^*. Since $t'_{L^*} > b_{L \cup M_T}$, among the $t_k \geq t'_{L^*}$ columns with a top terminal of N_k on it, at least one of them has a bottom terminal of a net in $R^* \cup M_P$ on it. Hence this net crosses every net in L at this column, making the local density at this column at least $|L| + 1$. We can also claim that if $\delta = 1$, then the minimum channel density that can be achieved by permuting terminals is at least $|L| + 1$ or $|R| + 1$. Suppose

$$b_{M_T \cup B} < (t'_{L^*} - b_{L^*}) + (t'_{R^*} - b_{R^*}),$$

then $t'_{L^*} - b_{L^*} > 0$ and $t'_{R^*} - b_{R^*} > 0$, for otherwise we have either $t'_{L^*} > b_{M_T \cup B} + b_{L^*} = b_{L \cup M_T}$ or $t'_{R^*} > b_{R \cup M_T}$, contradicting the fact that $\delta_L = \delta_R = 0$. To avoid a net in $R^* \cup M_P$ crossing all the nets in L at some column, we need $t'_{L^*} - b_{L^*}$ bottom terminals from the nets in $B \cup M_T$ to be assigned to columns with a terminal of a net in L^* on them . Similarly, to avoid a net in $L^* \cup M_P$ crossing all the nets in R at some column, we need $t'_{R^*} - b_{R^*}$ bottom although there are enough bottom terminals from the nets in $B \cup M_T$ to accomplish either one of the goals, there are not enough such bottom terminals to accomplish both goals at the same time. Hence the claim follows.

3 Preliminaries

We present in this section some preliminary results. Theorem 3.1 provides a lower bound for the minimum channel density achievable by permuting terminals. Lemma 3.2 states a result used in establishing the optimality of our algorithm.

Theorem 3.1 *Given an instance $\Pi = (N, L, R)$ of the Density_Minimization problem, the minimum channel density achievable by permuting terminals is at least*

$$D_\Pi = \max\{|B| + d^*_\Pi, \max\{|L| + \delta_L, |R| + \delta_R\} + \delta\},$$

where

$$d^*_\Pi = \begin{cases} 0 & \text{if } t_k, b_k \leq 1 \text{ for all } N_k \in M \\ 1 & \text{if } l \geq m_{L^* \cup M_P \cup R^*} \\ 2 & \text{otherwise.} \end{cases}$$

In other words, the channel cannot be routed in fewer than D_Π tracks.

Proof: Let d_Π be the minimum channel density achievable by permuting terminals. It is obvious that $d_\Pi \geq |B|$ because every net in B intersects every column of the channel. If $d^*_\Pi = 1$, then there exists a net in M with at least two terminals on the same side of the channel. Hence this net crosses all the nets in B at some column c, therefore $d_\Pi \geq |B| + 1$. If $d^*_\Pi = 2$, then $l < m_{L^* \cup M_P \cup R^*}$. In this case, there exist two distinct nets in $L^* \cup M_P \cup R^*$ having their terminals assigned to the same column. Hence these two nets together with all nets in B cross this column and the density of the channel is therefore at least $|B| + 2$. Thus $d_\Pi \geq |B| + d^*_\Pi$. To show that $d_\Pi \geq \max\{|L| + \delta_L, |R| + \delta_R\} + \delta$, observe that $d_\Pi \geq \max\{|L|, |R|\}$. Hence it suffices to consider the case where $\max\{\delta_L, \delta, \delta_R\} = 1$. We consider the following cases:

- Case 1: $\delta = 0$.
Without loss of generality assume $|L| + \delta_L \geq |R| + \delta_R$. If $\delta_L = 0$, then

$$d_\Pi \geq \max\{|L|, |R|\} = \max\{|L| + \delta_L, |R| + \delta_R\} + \delta.$$

Suppose $\delta_L = 1$ and L is top critical, then

$$d_\Pi \geq |L| + 1 = \max\{|L| + \delta_L, |R| + \delta_R\} + \delta.$$

The case where L is bottom critical can be proved in a similar way.

- Case 2: $\delta = 1$.
In this case, $|L| = |R|$ and $\delta_L = \delta_R = 0$. Hence either $d_\Pi \geq |L| + 1$ or $d_\Pi \geq |R| + 1$. In either case we conclude that

$$d_\Pi \geq \max\{|L|, |R|\} + \delta = \max\{|L| + \delta_L, |R| + \delta_R\} + \delta.$$

Therefore, the theorem follows. □

Theorem 3.1 provides a lower bound on the minimum density of a channel achievable by permuting terminals. For the example shown in Figure 1, we have

$$m_{L^* \cup M_P \cup R^*} = (3 + 4) + (10 + 2) + 4 = 23 > l,$$

hence $d_\Pi^* = 2$. Since $|B| = 1$, $|L| = 3$, $|R| = 2$, $\delta_R = 1$ and $\delta_L = \delta = 0$, we have $D_\Pi = 3$. Figure 1 shows a terminal permutation that actually achieves channel density 3. Later on, we will show that this bound is always achievable by presenting an algorithm that constructs a channel that achieves it. The following lemma is used in showing the optimality of the algorithm.

Lemma 3.2 *Let $X = (x_1, x_2, \ldots, x_u)$ and $Y = (y_1, y_2, \ldots, y_v)$ be sequences of non-ngative integers, such that $u \geq v$, $x_1 \leq x_2 \leq \ldots \leq x_u$, $y_1 \geq y_2 \geq \ldots \geq y_v$ and*

$$S = \sum_{i=1}^{u} x_i = \sum_{j=1}^{v} y_j,$$

then for $1 \leq k \leq v$,

$$\alpha_k = \sum_{i=1}^{k} x_i \leq \sum_{j=1}^{k} y_j = \beta_k.$$

Proof: Since X is nondecreasing, we have

$$\begin{aligned} S - \alpha_k &= \sum_{j=k+1}^{u} x_j \\ &\geq (u-k)x_{k+1} \\ &\geq \frac{u-k}{k}\alpha_k, \end{aligned}$$

from which we get

$$\alpha_k \leq \frac{k}{u}S.$$

Similarly, we can show that

$$\beta_k \geq \frac{k}{v}S.$$

It now follows from the assumption that $u \geq v$ that for $1 \leq k \leq v$, $\alpha_k \leq \beta_k$. □

4 Alternate Packing

It is convenient to present our algorithm using a technique called *alternate packing* [6, 19], which computes a terminal permutation by "packing nets" one at a time. After a net is "packed", the positions of all its terminals on one side of the channel are determined. The algorithm proceeds in such a way that after the packing of a net, there is at most one *partially assigned net* (*PAN* for short), *i.e.*, a net for which the positions of some but not all of its terminals are determined, and all the unassigned terminals of the PAN are on the same side of the channel. If these terminals are on the top of the channel, then the next net to be packed is a negative net, if there is any, otherwise the next net to be packed is a positive net. This technique is described formally in the following algorithm.

Algorithm: Alternate_Packing (Ω);
 (* $\Omega = (N_1, N_2, \ldots, N_n)$ is an ordered sequence of the nets to be routed *)
 Begin
  pan := 0;   (* the index of the PAN *)
  σ := 0;   (* $|\sigma|$ is the number of unassigned terminals of the PAN *)
  current := 1;   (* the index of the next net to be packed *)
  column := 1; $t := t_1$; $b := b_1$;
  while $\Omega \neq \phi$ **do**
   begin
    Remove the current net from Ω;
    if $\sigma \geq 0$

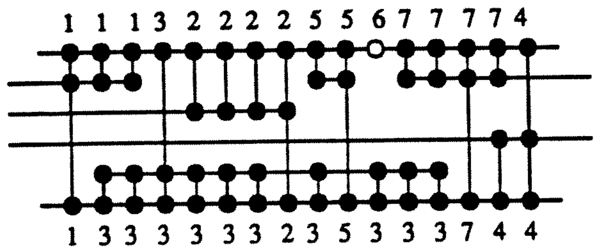


Figure 2: Alternate Packing

```
        (* the PAN has σ unassigned top terminals and *)
        (* the current net is *)
        then for k := 1 to min{t + σ, b} do
                if k ≤ min{σ, b − t}
                   then Assign_Column (column + k − 1, pan, current)
                   else Assign_Column (column + k − 1, current, current)
        (* σ < 0, the PAN has −σ unassigned bottom terminals, and *)
        (* the current net is positive *)
        else for k := 1 to min{t, b − σ} do
                if k ≤ min{t − b, σ}
                   then Assign_Column (column + k − 1, current, pan)
                   else Assign_Column (column + k − 1, current, current);
      if σ + (t − b) > 0
      (* the new PAN has only unassigned top terminals *)
        then N_j := the next negative net in Ω
        else N_j := the next positive net in Ω;
      if |t − b| ≥ |σ|
        then pan := current;
      column := column + k; σ := σ + (t − b); current := j; t := t_j; b := b_j
    end
End.
```

A call to procedure Assign_Column (c, i, j) assigns a top terminal of net N_i to column c, a bottom terminal of net N_j to column c. Since we can use two linked lists to separately store the positive nets and the negative nets, the algorithm can be implemented to run in $O(l)$ time, where l is the length of the channel. Figure 2 shows a channel obtained by applying Algorithm Alternate_Packing on the example shown in Figure 1 with $\Omega = (N_1, N_2, N_3, N_4, N_5, N_6, N_7)$. The nets are actually packed in the following order: $(N_1, N_3, N_2, N_5, N_6, N_7, N_4)$. The density of the channel is equal to 4, instead of the optimal value 3.

Note that if the channel has no exits, then the density of the channel obtained by Algorithm Alternate_Packing is at most two. This is because at most two nets intersect each column, one of them is the PAN when the column is under consideration, the other is the net under packing at that time. By carefully ordering the nets in L^* and R^*, the algorithm can also be used to compute a terminal permutation for a given channel with exits so that the resulting channel has density within one of the optimal value [6].

Observe that for any net of the form (k, k), Algorithm Alternate_Packing assigns all of its terminals to k consecutive columns. In particular, any net of the form $(1, 1)$ have both its terminals assigned to the same column.

5 The Optimal Algorithm

We present in this section our linear time exact algorithm for the Density_Minimization problem. Our algorithm computes a terminal permutation that achieves minimum channel density, which is always equal to the lower bound given in Theorem 3.1.

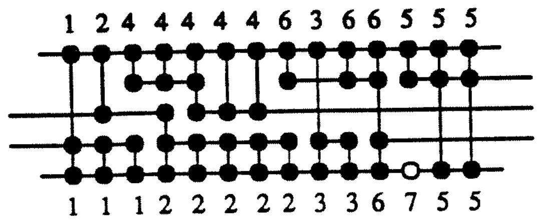
As we have seen in the last section, Algorithm Alternate_Packing is not optimal in general. In order to achieve optimality, the terminals of the nets have to be carefully distributed, so that, for example, no net in $M_P \cup R^*$ is to cross all nets in L at some column, if this is necessary and possible (*i.e.*, when $D_\Pi = |L|$ and hence $\max\{\delta, \delta_L\} = 0$). A convenient way of doing this is to partition the nets into a number of "smaller" nets, each containing a subset of the set of terminals of the original net. This is done in the following procedure. (We use $X.Y$ to denote the concatenation of two sequences X and Y. This notation extends to the case that X or Y is a set because we can consider a set as a sequence by arbitrarily ordering its elements.)

Procedure: Distribute_Nets (Π);
  (* $\Pi = (N, L, R)$ is an instance of the Density_Minimization problem *)
  Begin
    (* decompose the nets in L^* and R^* *)
    $L' := \epsilon$; $R' := \epsilon$;   (* ϵ is the empty sequence *)
    for $N_k \in L^*$ **do**
      if $t_k * b_k = 0$
        then $L' := L'.(N_k)$
        else begin
            $N_k' := (t_k, 0)$; $N_k'' := (0, b_k)$; $L' := L'.(N_k', N_k'')$

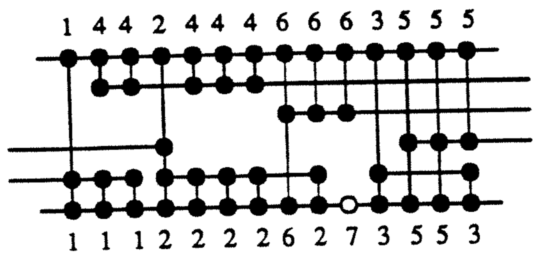
```
            end ;
    for N_k ∈ R* do
      if t_k * b_k = 0
        then R' := R'.(N_k)
        else begin
               N'_k := (t_k, 0); N''_k := (0, b_k);  R' := R'.(N'_k, N''_k)
             end ;
    S := M_T;
    (* decompose nets in B into trivial nets *)
    for N_i ∈ B do
      for j := 1 to t_i + b_i do
        begin
          if j ≤ t_i
            then N_{i_j} := (1, 0)
            else N_{i_j} := (0, 1);
          S := S.(N_{i_j})
        end;
    (* distribute the nets in S *)
    S_L := φ; S_R := φ;
    if D_Π = |L|
      then if t'_{L*} ≥ b_{L*}
               then S_L := the first (t'_{L*} - b_{L*}) negative nets of S
               else S_L := the first max{0, b'_{L*} - t_{L*}} positive nets of S;
    if D_Π = |R|
      then if t'_{R*} ≥ b_{R*}
               then S_R := the last (t'_{R*} - b_{R*}) negative nets of S
               else S_R := the last max{0, b'_{R*} - t_{R*}} positive nets of S;
    S_M := S - S_L ∪ S_R
  End.
```

A net of the form (t, b) is said to have *disparity* $|t - b|$. A *tag* of L^* is a net $N_k \in L^*$ such that $t_k = t'_{L^*}$ if $t_{L^*} \geq b_{L^*}$, and $b_k = b'_{L^*}$ otherwise. The tag of R^* is similarly defined.

Procedure Distribute_Nets decomposes each net in $L^* \cup R^*$ into two one-sided nets, if it is not itself one sided. It also partitions each net in B into nets of the form $(1, 0)$ and $(0, 1)$. These nets together with the trivial nets in M_T form the set S. Note that the set of new nets obtained from decomposing the same net are grouped together and the ordering of the new nets are consistent with the original ordering of the old nets (*i.e.*, if an old net N_i appeared before N_j, then all the new nets obtained from decomposing N_i appear before all the nets obtained from decomposing N_j). The set of new nets S is partitioned into three subsets S_L, S_R and S_M. The set S_L is introduced to avoid having some net in $M_P \cup R^*$ crossing



(a) An optimal solution



(b) A near-optimal solution

Figure 3: Optimal Packing

every net in L at some column. This is necessary if $D_\Pi = |L|$, because otherwise the density of the channel is at least $|L| + 1 > D_\Pi$. It is also possible in this case because $D_\Pi = |L|$ implies $\max\{\delta, \delta_L\} = 0$, hence there are enough terminals in $L \cup M_T$ to pad the columns with terminals of the tag of L^* on it. The set S_R is similarly introduced to avoid having a net in $L^* \cup M_P$ crossing every net in R. The set S_M is introduced to keep positive nets in M_P from crossing negative nets in M_P. This is necessary if $D_\Pi = |B| + 1$, for otherwise the density of the channel would be at least $|B| + 2$. It is possible if $d^*_\Pi \leq 1$.

With the nets decomposed as described in Procedure Distribute_Nets, we can distribute the terminals to where we want them to be in applying Algorithm Alternate_Packing by carefully ordering the nets. This is enough to achieve optimality.

Algorithm: Optimal_Packing (Π);
  (* $\Pi = (N, L, R)$ is an instance of the Density_Minimization problem *)
  Begin
    Sort L^* into increasing order of net disparities;
    Sort R^* into decreasing order of net disparities;
    if $D_\Pi = |L|$ **and** $(t'_{L^*} > b_{L^*}$ **or** $b'_{L^*} > t_{L^*})$
      then Select a tag of L^* as its first net;
    if $D_\Pi = |R|$ **and** $(t'_{R^*} > b_{R^*}$ **or** $b'_{R^*} > t_{R^*})$
      then Select a tag of R^* as its last net;

$M_P^+ := \{N_k \in M_P : t_k \geq b_k\}$; $M_P^- := M_P - M_P^+$; Distribute_Nets (Π);
if $t_{L^*} > b_{R^*}$
then $\Omega := S_L.L'.M_P^-.S_M.M_P^+.R'.S_R$
else $\Omega := S_L.L'.M_P^+.S_M.M_P^-.R'.S_R$;
Alternate_Packing (Ω)
End.

Example 2: Consider the channels shown in Figure 3. We have $n = 7$, $N_1 = (1,3)$, $N_2 = (1,5)$, $N_3 = (1,2)$, $N_4 = (5,0)$, $N_5 = (3,2)$, $N_6 = (3,1)$, $N_7 = (0,1)$; $L = \{N_1, N_2\}$, $R = \{N_4, N_5, N_6\}$, $B = \phi$; $S = M_T = \{N_7\}$, $M_P = \{N_3\}$. Since $D_\Pi = 3 > |L| = 2$ and $t'_{R^*} = 3 = b_{R^*}$, $b'_{R^*} = 0 < t_{R^*} = 8$, we have $S_L = S_R = \phi$, and L is sorted into (N_1, N_2), R is sorted into (N_4, N_6, N_5). After decomposing the nets, we obtain

$$\Omega = (N_1', N_1'', N_2', N_2'', N_3, N_7, N_4, N_6', N_6'', N_5', N_5'').$$

The channel constructed by our algorithm is shown in Figure 3(a) which achieves optimal density 3. Figure 3(b) shows the channel constructed by the algorithm in [6] which has density equal to 4, one more than the optimal value. The difference is that because the nets in L^* and R^* are decomposed in our algorithm, net N_3 does not cross every net in R in the channel constructed by our algorithm, whereas it does in the channel constructed by the algorithm in [6]. □

6 The Optimality of Our Algorithm

We show in this section that Algorithm Optimal_Packing constructs a channel with minimum density given an instance of the Density_Minimization problem. We first state the correctness of the algorithm, which follows from the correctness of Algorithm Alternate_Packing.

Theorem 6.1 *Algorithm Optimal_Packing computes a valid terminal permutation of a given instance of the Density_Minimization problem,* i.e., *each terminal is assigned a unique position and no two terminals are assigned to the same position.*

To establish the optimality of the algorithm, we need the following lemmas.

Lemma 6.2 *In the channel produced by Algorithm Alternate_Packing, no net in $L^* \cup R^*$ crosses two nets in M_P at some column.*

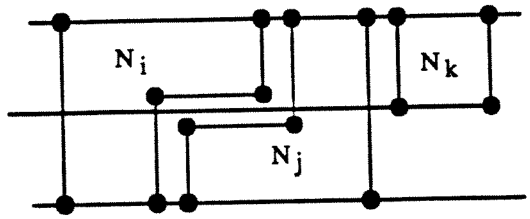


Figure 4: Illustration of the proof of Lemma 6.2

Proof: Assume, to the contrary, that $N_i, N_j \in M_P$ and $N_k \in L^*$ all intersect certain column c of the channel, and that N_i is packed before N_j. Then one of N_i, N_j must be positive and the other negative. Without loss of generality, assume N_i is positive and N_j is negative. This scenario is illustrated in Figure 4. Since the nets in L^* are placed before the nets in M_P, the fact that N_j is packed before N_k (for otherwise N_k would not cross N_j) implies that N_k is positive. This, however, implies that N_k should have been packed before N_i, a contradiction. □

Lemma 6.3 *If $L^* = R^* = \phi$, then Algorithm Optimal_Packing produces a minimum density channel.*

Proof: We show that in this case the channel density is equal to $D_\Pi = |B| + d^*_\Pi$. Since the nets in M_P are not splitted in the algorithm, at most two nets in M_P intersect a column. Hence the density of the channel is $\leq |B| + 2$. Thus if $d^*_\Pi = 2$, then the lemma holds. If $d^*_\Pi = 0$, then $t_k, b_k \leq 1$ for any net $(t_k, b_k) \in M_P$. Since Algorithm Alternate_Packing assigns both terminals of any net of the form $(1, 1)$ to the same column, no net in M_P intersects any column of the channel. Hence the density of the channel is $|B|$. In the remaining case we have $d^*_\Pi = 1$. Since the nets in M_P^+ precede the nets in $S = S_M$ precede the nets in M_P^-, two nets $N_i \in M_P^+$ and $N_j \in M_P^-$ cross each other only when all negative nets in S have been exhausted. This implies that $t_{M_P^+} > b_{M_P^+} + b_S = b_{M_P^+ \cup S} = b_{N - M_P^-}$. However, because $d^*_\Pi = 1$, we have $l \geq m_{M_P} = t_{M_P^+} + b_{M_P^-}$, *i.e.*, $t_{M_P^+} \leq l - b_{M_P^-} = b_{N - M_P^-}$, which is a contradiction. Hence the density of the channel is $|B| + 1$. This completes the proof of the lemma. □

Lemma 6.4 *If $M_P = \Phi$, then Algorithm Optimal_Packing produces a minimum density channel.*

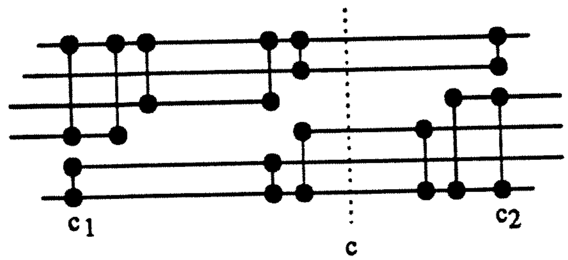


Figure 5: Illustration of the proof of Lemma 6.4

Proof: If no net in L^* crosses any net in R^*, then the density of the channel is equal to $\max\{|L|, |R|\}$, which is clearly optimal. Otherwise, due to the fact that the nets in $L^* \cup R^*$ are decomposed into one-sided nets, there must exist columns c_1 and c_2, $c_1 \leq c_2$, such that

- All terminals of the nets in $M_T \cup B$ are on either the leftmost $c_1 - 1$ columns or the rightmost $l - c_2$ columns of the channel;
- All terminals of the nets in L^* on one side of the channel (say, the bottom of the channel) are on the leftmost $c_1 - 1$ columns of the channel;
- All terminals of the nets in R^* on the other side of the channel (*i.e.*, the top of the channel) are on the rightmost $l - c_2$ columns of the channel;
- No terminal of the nets in R^* is on the leftmost $c_1 - 1$ columns of the channel; and
- No terminal of the nets in L^* is on the rightmost $l - c_2$ columns of the channel.

(In the example shown in Figure 3(a), we have $c_1 = 1$ and $c_2 = 8$.) Hence the maximal local density of the leftmost $c_1 - 1$ columns of the channel is $|L|$ and the maximal local density of the rightmost $l - c_2$ columns of the channel is $|R|$. Let $L' = \{N_{i_1}, N_{i_2}, \ldots, N_{i_u}\}$ and $R' = \{N_{j_1}, N_{j_2}, \ldots, N_{j_v}\}$ be the set of nets (from left to right in that order) obtained from L^* and R^*, respectively, by excluding their terminals on the leftmost $c_1 - 1$ and rightmost $l - c_2$ columns of the channel, then $N_{i_k} = (t_{i_k}, 0)$ for $1 \leq k \leq u$ and $N_{j_k} = (0, b_{j_k})$ for $1 \leq k \leq v$. Since the nets in L^*, R^* are sorted into increasing, decreasing order of disparities, respectively, we have $t_{i_1} \leq t_{i_2} \leq \ldots \leq t_{i_u}$, $b_{j_1} \geq b_{j_2} \geq \ldots \geq b_{j_v}$ and

$$\sum_{k=1}^{u} t_{i_k} = \sum_{k=1}^{v} b_{j_k} = c_2 - c_1 + 1.$$

If $u \geq v$, then for $c_1 \leq c \leq c_2$, let

$$x_c = \min\{h : \sum_{k=1}^{h} t_{i_k} \geq c - c_1 + 1\}$$

and

$$y_c = \min\{h : \sum_{k=1}^{h} b_{j_k} \geq c - c_1 + 1\},$$

then $d_c \leq (u - x_c + 1) + y_c = (u+1) - (x_c - y_c)$. (See Figure 5 for illustration.) According to Lemma 3.2, we have

$$\sum_{k=1}^{h} t_{i_k} \leq \sum_{k=1}^{h} b_{j_k}$$

for $1 \leq h \leq v$, implying $x_c \leq y_c$. Hence $d_c \leq u+1$. Similarly, we can show that $d_c \leq v+1$ if $u \leq v$. Therefore, in either case we have $d_c \leq \max\{u+1, v+1\}$. We claim that $d_c \leq D_\Pi$ for $c_1 \leq c \leq c_2$. It suffices to show that $u < D_\Pi$ and $v < D_\Pi$. Assume, to the contrary, that $u = D_\Pi$, then $|L| = D_\Pi$ because $u \leq |L|$. This implies that there exists a net in R^* crosses every net in L, and hence $b'_{L^*} > t_{L^*}$ and $S_L = \phi$. But this is impossible according to Procedure Distribute_Nets. Hence $u < D_\Pi$. Similarly, we can show that $v < D_\Pi$. Therefore, the density of the channel is $\leq D_\Pi$. □

Theorem 6.5 *Given a Density_Minimization instance* Π, *Algorithm Optimal_Packing produces a minimum density channel.*

Proof: Let columns x, y, be, respectively, the rightmost, leftmost column with a terminal of a net in L^*, R^* on it. To show that the channel has minimum density, we consider the following two cases:

- Case 1: $x < y$.
  In this case, there is no terminal of nets in R^* on the leftmost x columns of the channel. Since the rightmost net in L^* intersects all of the leftmost x columns of the channel, none of these columns is intersected by two nets in M_P by Lemma 6.2. Therefore, the maximal local density at the leftmost x columns of the channel is $\leq |L| + 1$, with equality holds only if a net in M_P crosses the leftmost net in L^*. This can happen only when $D_\Pi > |L|$. Hence the maximal local density at the leftmost x columns of the channel is $\leq D_\Pi$. Similarly, the maximal local density at the rightmost $l - y + 1$

columns of the channel is at most D_Π. We proceed to show that the maximal local density at the middle $y-x-1$ columns of the channel (from column $x+1$ to column $y-1$) is $\leq D_\Pi$. Note that no net in $L^* \cup R^*$ intersects any of these columns. As in the proof of Lemma 6.3, only the case where $d^*_\Pi = 1$ is of interest (for the maximal local density at these columns of the channel is equal to $|B| + d^*_\Pi$ in the remaining cases). It suffices to consider the case where $D_\Pi = |B|+1$. Without loss of generality assume $t_{L^*} \leq b_{L^*}$, then $\Omega = S_L.L'.M_P^+.S_M.M_P^-.R'.S_R$ and $b_{S_L} = 0$. If a net $N_i \in M_P^+$ crosses a net $N_j \in M_P^-$, then we have $t_{L' \cup S_L \cup M_P^+} > b_{L' \cup S_L \cup M_P^+ \cup S_M}$. If $L^* = \phi$, then $S_L \cup L' = \phi$, otherwise since $|L^*| = 1$ we have $t_{S_L} = b_{L^*} - t_{L^*}$. In either case $t_{L' \cup S_L} = b_{L' \cup S_L}$. We can also show that $b_{S_R \cup R'} = m_{R'}$. Therefore

$$
\begin{aligned}
t_{M_P^+} &> b_{L' \cup S_L \cup M_P^+ \cup S_M} - t_{L' \cup S_L} \\
&= b_{M_P^+ \cup S_M} - (t_{L' \cup S_L} - b_{L' \cup S_L}) \\
&= b_{M_P^+ \cup S_M}.
\end{aligned}
$$

On the other hand, since $l \geq m_{L' \cup M_P \cup R'}$, we have

$$
\begin{aligned}
t_{M_P^+} &= l - b_{L'} - b_{M_P^-} - m_{R'} \\
&= b_{S_L \cup M_P^+ \cup S_M} + (b_{S_R \cup R'} - m_{R'}) \\
&= b_{S_L \cup M_P^+ \cup S_M} \\
&= b_{M_P^+ \cup S_M},
\end{aligned}
$$

which is a contradiction. Hence no two nets in M_P cross each other. Therefore, the maximal local density at the middle $y-x-1$ columns of the channel is also $\leq D_\Pi$. We conclude that in this case the density of the channel is $\leq D_\Pi$.

- Case 2: $x \geq y$.
  In this case, the part of the channel from column y to column x must of the form as shown in Figure 5. Similar to the proof of Lemma 6.3, we can show that the maximal local density at these columns of the channel is at D_Π. According to to the analyses in Case 1, the maximal local density at the leftmost $y-1$ columns and the rightmost $l-x$ columns of the channel is at most D_Π. Hence the density of the channel is $\leq D_\Pi$.

Thus in both case we conclude that the density of the channel is D_Π, which is optimal according to Theorem 3.1. □

Since both L^* and R^* can be sorted in linear time by bucket sort [1], the running time of the algorithm is easily seen to be linear, as stated in the following theorem.

Theorem 6.6 *The running time of Algorithm Optimal_Packing is $O(l)$, where l is the length of the channel.*

7 NP-Hardness Results

In this section, we show that the problem of permuting the terminals of a channel so that it can be routed in minimum wire length is NP-hard in the strong sense by proving the following decision version of the problem is NP-complete in the strong sense.

Wire_Length_Minimization (WLM)
INSTANCE: A set of nets $N = \{N_1, N_2, \ldots, N_n\}$ to be routed with

$$\sum_{k=1}^{n} t_k = \sum_{k=1}^{n} b_k = l,$$

and a positive integer W.
QUESTION: Is there a permutation of the terminals such that the channel can be routed in total wire length $\leq W$?

Theorem 7.1 *The WLM problem is NP-complete in the strong sense even for channels with no exits.*

We prove Theorem 7.1 by a polynomial time transformation from the following 3-Partition problem, which is known to be NP-complete in the strong sense [10]. (I^+ stands for the set of positive integers.)

3-Partition
INSTANCE: A finite set A of $3m$ elements, a bound $b \in I^+$, and a "size" $s(a) \in I^+$ for each $a \in A$, such that each $s(a)$ satisfies $b/4 < s(a) < b/2$ and such that

$$\sum_{a \in A} s(a) = mb.$$

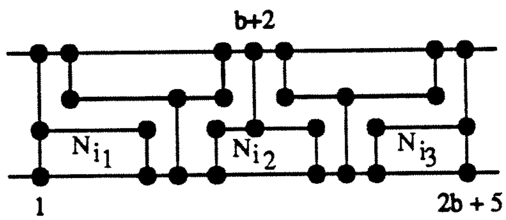


Figure 6: Illustration of the proof of Lemma 7.2

QUESTION: Can A be partitioned into m disjoint sets $A_1, A_2, \ldots, A_m$ such that, for $1 \leq i \leq m$,

$$\sum_{a \in A_i} s(a) = b?$$

Given an instance of the 3-Partition problem, we construct an instance of the WLM problem with $l = 2mb + 5m$, $W = 8mb + 15m$ and $n = 5m$ nets such that

$$\begin{aligned} N_k &= (b+1, 1) \qquad \text{for } 1 \leq k \leq 2m \\ N_k &= (1, 2s(a_{k-2m}) + 1) \qquad \text{for } 2m+1 \leq k \leq 5m. \end{aligned}$$

The first $2m$ nets are called *top nets*, the last $3m$ nets are called *bottom nets*. Each bottom net corresponding to an element of A. It is easy to check that

$$l = \sum_{k=1}^{n} t_k = \sum_{k=1}^{n} b_k = 2mb + 5m.$$

We claim that the given 3-Partition instance has a solution if and only if the answer to the corresponding WLM instance is yes.

Lemma 7.2 *If the given 3-Partition instance has a solution, then the corresponding WLM instance also has a solution.*

Proof: Let $A_1, A_2, \ldots, A_m$ be a solution to the given 3-Partition instance such that $A_i = \{a_{i_1}, a_{i_2}, a_{i_3}\}$ for $1 \leq i \leq m$. Then $s(a_{i_1}) + s(a_{i_2}) + s(a_{i_3}) = B$ for $1 \leq i \leq m$. We construct a solution for the WLM instance as follows. Partition the channel into m sub-channels

(a) A left-matched bottom net (b) A right-matched bottom net

Figure 7: Matched bottom nets

each of length $2b+5$, and assign two top nets and three bottom nets N_{i_1}, N_{i_2}, N_{i_3} to the ith subchannel and route it as shown in Figure 6. Let W_{h_i}, W_{v_i} be, respectively, the total horizontal, vertical wire lengths in the routing of the ith subchannel, then

$$\begin{aligned} W_{h_i} &= 2b + (2s(a_{i_1}) + 2s(a_{i_2}) + 2s(a_{i_3})) \\ &= 4b, \text{ and} \\ W_{v_i} &= 10 + 2(b+1) + (2s(a_{i_1}) + 2s(a_{i_2}) + 2s(a_{i_3}) + 3) \\ &= 4b + 15. \end{aligned}$$

Hence the total length in the routing of the ith subchannel is $W_i = W_{h_i} + W_{v_i} = 8b + 15$, and the total wire length in the routing of whole channel is

$$\sum_{i=1}^{m} W_i = 8mb + 15m = W.$$

Hence the terminal permutation so determined is a solution to the WLM instance □

A net $N_k \in N$ of a channel is said to be *proper* if all terminals of N_k are assigned to $\max\{t_k, b_k\}$ consecutive columns. A bottom net N_k is said to be *left-matched* (*right-matched*, respectively) if its top terminal is assigned to the leftmost (rightmost, respectively) column having a terminal of N_k on it. (See Figure 7 for illustration.) To prove the converse of Lemma 7.2, we need the following lemma.

Lemma 7.3 *For any terminal permutation of the WLM instance as constructed above, the resulting channel has to be routed in wire length $\geq W$, and equality holds only if all nets are proper.*

Proof: Let W_h, W_v be, respectively, the total horizontal, vertical wire lengths in a routed channel resulted from a terminal permutation of the WLM instance, and let w be the width

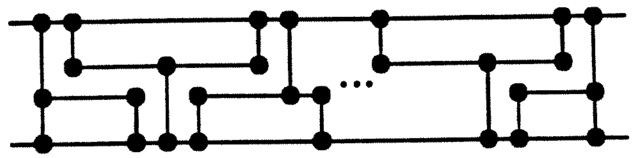


Figure 8: A segment of a channel

of this routed channel. Then we have

$$W_h \geq 2mb + \sum_{a\in A} 2s(a) = 4mb,$$

where equality holds if and only if for each net $N_k \in N$, all terminals of N_k are assigned to $\max\{t_k, b_k\}$ consecutive columns. We also have

$$\begin{aligned} W_v &= 2m(b+1) + 5mw + \sum_{a\in A}(2s(a)+1) \\ &= 4mb + 5m(w+1). \end{aligned}$$

Since

$$l = \sum_{k=1}^{n} t_k = \sum_{k=1}^{n} b_k = 2mb + 5m,$$

and there exists a net $N_k \in N$ with $t_k \neq b_k$, we must have $w \geq 2$. Therefore, the total wire length in this routed channel is at least $W_h + W_v \geq 8mb + 15m$, where equality holds if and only if $W_h = 4mb$ and $w = 2$. Since this holds for any routed channel resulted from any terminal permutation, the lemma follows. □

Lemma 7.4 *If the WLM instance as constructed above has a solution, then the given instance of the 3-Partition instance also has a solution.*

Proof: Suppose the WLM instance has a solution. According to Lemma 7.3, all nets in N are proper, Therefore, for any two bottom nets N_i and N_j, $2m+1 \leq i, j \leq 5m$, all the terminals of one of them are assigned the left of all the terminals of the other in the solution. Hence we can talk about a bottom net which is to the left (right) of another bottom net. It can be seen that the following conditions must hold in any solution to the WLM instance.

1. The leftmost bottom net is left-matched;
2. The rightmost bottom net is right-matched;

3. Each right-matched bottom net is followed immediately by a left-matched bottom net;

4. No left-matched bottom net is immediately followed by a right-matched bottom net.

The last statement is a consequence of the constraint $b/4 < s(a) < b/2$ on the sizes of elements of A. Therefore, the channel can be partitioned into *segments*, each starts with a left-matched bottom net and ends with the first right-matched bottom net after it (see Figure 8). Let $N_{k_1}, N_{k_2}, \ldots, N_{k_r}$ be the set of bottom nets in a segment of the channel from left to right in that order. Since for $1 \leq i \leq r$, $b_{k_i} = 2s(a) + 1$ for some $a \in A$, we have $b_{k_1} < b + 1 < b_{k_1} + b_{k_2}$. For $1 \leq i \leq r$, let

$$s_i = \sum_{j=1}^{i} b_{k_j},$$

then we have $s_i < i(b+1) < s_{i+1}$ for $1 \leq i < r-1$ and $s_r = (r-1)(b+1)$. This is because if $s_{j+1} < j(b+1)$ for some j, $1 \leq j < r-1$, then either the bottom net N_{k_j} or a top net is not proper, contradicting Lemma 7.3. Hence the number of top nets in a segment of the channel is exactly one fewer than the number of bottom nets in the same segment. Because there are m more bottom nets than top nets, the number of segments in the channel must be exactly m. Furthermore, according to the statment 4 above, the number of bottom nets in a segment is at least three. Since there are $3m$ bottom nets in total, each segment contains exactly three bottom nets. Let x, y, z be the elements of A corresponding to the bottom nets in a segment of the channel, then

$$(2s(x) + 1) + (2s(y) + 1) + (2s(z) + 1) + 2 = 2(b+1) + 3,$$

i.e., $s(x) + s(y) + s(z) = b$. Hence $A_1, A_2, \ldots, A_m$ is a solution to the 3-Partition instance, where A_i contains the elements of A corresponding to the bottom nets in the ith leftmost segment of the channel, $1 \leq i \leq m$. $\Box$.

Theorem 7.1 now follows from Lemmas 7.2 and 7.4 and the observation that the transformation can be carried out in polynomial time. From the definition of NP-hardness, we also have

Corollary 7.5 *The problem of computing a terminal permutation so that the resulting channel can be routed in minimum wire length is NP-hard in the strong sense even for channels without exits.*

Define the *wire span* of a channel routing solution to be the total length of the horizontal wires. We can show that the problem of computing a terminal permutation so that the resulting channel can be routed in minimum wire span is also NP-hard in the strong sense by showing the decision version of it is NP-complete in the strong sense. The proof is identical to that of Theorem 7.1 except that W replaced by $W_h = 4mb$. Hence we have

Corollary 7.6 *The problem of finding a terminal permutation so that the resulting channel can be routed in minimum wire span is NP-hard in the strong sense even for channels without exits.*

8 Concluding Remarks

We present in this paper an optimal linear time algorithm for computing a permutation of the terminals of a channel so that the resulting channel has minimum density. The previously known best result for this problem is a near optimal algorithm that runs in superlinear time. Hence our algorithm improves both the efficiency and optimality of the previous known algorithm. We also show that the problem of computing a terminal permutation so that the resulting channel can be routed in minimum wire length is NP-hard in the strong sense.

We believe that our algorithm can be modified to generate a channel that can be routed in minimum width as well. Several related problems are currently under investigation. These include the case where one or both ends of the channel are not aligned; the case where the sides of the channel are slidable; and the case where the positions of the terminals on one side of the channel are completely fixed. We believe that some of these problems can also be solved optimally in polynomial time.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The design and analysis of computer algorithms*, Addison Wesley, Reading, MA, 1974.

[2] M. Burstein and R. Pelavin, "Hierarchical channel router", *INTEGRATION, the VLSI journal, vol.* **1**, 21-38, 1983.

[3] Y. Cai and D.F. Wong, "An optimal channel pin assignment algorithm", *Proc. of the 1990 IEEE International Conference on Computer-Aided Design*, 10-13.

[4] Y. Cai and D.F. Wong, "Minimizing channel density by shifting blocks and terminals", *Proc. of the 1991 IEEE International Conference on Computer-Aided Design*, 524-527.

[5] Y. Cai and D.F. Wong, "Optimal channel pin assignment", *IEEE Trans. on CAD, vol. CAD*-**10**, 1413-1424, 1991.

[6] J. Cong and K.-Y. Khoo, "A provable near-optimal algorithm for the channel pin assignment problem", *Proc. of the 1991 IEEE International Conference on Computer Design*, 319-322.

[7] J. Cong, "Pin assignment with global routing", *Proc. of the 1989 IEEE International Conference on Computer-Aided Design*, 302-305.

[8] J. Deogun and B. Bhattacharya, "Via minimization in VLSI routing with movable terminals", *IEEE Trans. on CAD, vol. CAD*-**8**, 917-920, 1990.

[9] D.N. Deutsch, "A dogleg channel router", *Proc. of the 13rd ACM/IEEE Design Automation Conference*, 425-433, 1976.

[10] M.R. Garey and D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, W.H. Freeman & Co., New York, NY, 1979.

[11] I.S. Gopal, D. Coppersmith and C.K. Wong, "Optimal wiring for movable terminals", *IEEE Trans. on Computers, vol. C*-**32**, 845-858, 1983.

[12] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures", *Proc. of the Eighth ACM/IEEE Design Automation Workshop*, 155-163, 1971.

[13] T.W. Her, D.F. Wong and T.H. Freeman, "Optimal orientation of transistor chains", *Proc. of the 1990 IEEE International Conference on Computer-Aided Design*, 524-527.

[14] T.W. Her and D.F. Wong, "Optimal module implementation and its application to transistor placement", *Proc. of the 1991 IEEE International Conference on Computer-Aided Design*, 98-101.

[15] C.Y. Hou and C.Y.R. Chen, "A hierarchical methodology to improve channel routing by pin permutation", *Proc. of the 1991 IEEE International Conference on Computer-Aided Design*, 440-443.

[16] H. Kobayashi and C.E. Drozd, "Efficient algorithms for routing interchangeable terminals", *IEEE Trans. on CAD, vol. CAD-***4**, 204-207, 1985.

[17] A.S. LaPaugh and R.Y. Pinter, "On minimizing channel density by lateral shifting", *Proc. of the 1983 IEEE International Conference on Computer-Aided Design*, 121-122.

[18] H.W. Leong and C.L. Liu, "Permutation channel routing", *Proc. of the 1985 IEEE International Conference on Computer Design*, 579-584.

[19] H.W. Leong, *Routing problems in physical design of integrated circuits*, Ph.D. Dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, 1986.

[20] L.-S. Lin and S. Sahni, "Maximum alignment of interchangeable terminals" *IEEE Trans. on Computers, vol. C-***37**, 1166-1177, 1988.

[21] M. Pedram, M. Marek-Sadowska and E.S. Kuh, "Floorplanning with pin assignment", *Proc. of the 1990 IEEE International Conference on Computer-Aided Design*, 98-101.

[22] R.L. Rivest and C.M. Fiduccia, "A greedy channel router", *Proc. of the 19th ACM/IEEE Design Automation Conference*, 418-424, 1982.

[23] S. Sahni and S.Y. Wu, "Two NP-hard interchangeable terminal problems", *IEEE Trans. on CAD, vol. CAD-***7**, 467-472, 1988.

[24] I. Tollis and S. Tragoudas, "Interchanging terminals for improved channel routing", *Proc. of the 1990 IEEE International Symposium on Circuits and Systems*, 344-347.

[25] P. Widmayer and C.K. Wong, "An optimal algorithm for maximum alignment of terminals", *Information Processing Letters, vol.* **20**, 75-82, 1985.

[26] T. Yoshimura and E.S. Kuh, "Efficient algorithms for channel routing", *IEEE Trans. on CAD, vol. CAD-***1**, 25-35, 1982.