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, 8, 11, 22]. 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, 10, 12, 13, 14, 15, 16, 17, 19, 20, 21]. 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 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].

Several special cases of the problem of minimizing channel density by permuting terminals have been studied before. In [18], 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. Due to space limitation, proofs of the results are not included, details can be found in [5].

2 Preliminaries

With respect to a horizontal 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 (DM) problem is the problem of permuting the terminals on the two sides of a channel to minimize its density. An instance of the DM problem is a 3-tuple $= (N, L, R)$, 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] $t_{N'} = \sum_{N_k \in N'} t_k$; $b_{N'} = \sum_{N_k \in N'} b_k$; $t'_{N'} = \min_{N_k \in N'} t_k$; $b'_{N'} = \min_{N_k \in N'} b_k$; $m_{N'} = \sum_{N_k \in N'} \max\{t_k, b_k\}$. 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

[1] For convenience of presentation, we define

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

*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.

1063-6404/92 $3.00 © 1992 IEEE

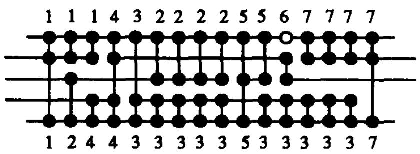


Figure 1: Illustration of definitions

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 $= (N, L, R)$ of the DM 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

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

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.

Theorem 2.1 *Given an instance $= (N, L, R)$ of the DM problem, the minimum channel density achievable by permuting terminals is at least*

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

where

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

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

Theorem 2.1 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 $d^* = 2$. Since $|B| = 1, |L| = 3, |R| = 2, \delta_R = 1$ and $\delta_L = \delta = 0$, we have $D = 3$. 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 2.2 *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 $1 \leq k \leq v$, $\alpha_k = \sum_{i=1}^{k} x_i \leq \sum_{j=1}^{k} y_j = \beta_k$.*

3 Alternate Packing

It is convenient to present our algorithm using a technique called *alternate packing* [6, 18], 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

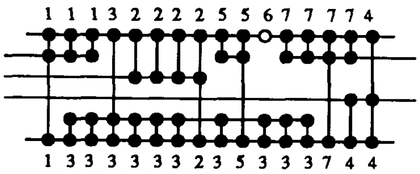


Figure 2: Alternate Packing

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 (W);
  (* W = (N_1, N_2, ..., N_n) is an ordered sequence of *)
  (* the nets to be routed *)
  Begin
    pan := 0; σ := 0;
    current := 1;     (* index of next net to be packed *)
    column := 1; t := t_1; b := b_1;
    while W ≠ φ do
      begin
        Remove the current net from W;
        if σ ≥ 0
        (* the PAN has σ unassigned top terminals *)
        (* and the current net is negative *)
        then for k := 1 to min{t + σ, b} do
                 if k ≤ min{σ, b − t}
                 then AC(column + k − 1, pan, current)
                 else AC(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 AC(column + k − 1, current, pan)
                 else AC(column + k − 1, current, current);
        if σ + (t − b) > 0
        (* next PAN has only unassigned top terminals *)
          then N_j := the next negative net in W
          else N_j := the next positive net in W;
        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 AC(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 $W = (N_1, N_2, N_3, N_4, N_5, N_6, N_7)$. 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.

4 The Optimal Algorithm

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 $D = |L|$ 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 ();
  (* = (N, L, R) is an instance of the DM problem *)
  Begin
    (* decompose the nets in L* and R* *)
    L' := ε; R' := ε;        (* ε is the empty sequence *)
    for N_k ∈ 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 = |L|
    then if t'_{L*} ≥ b_{L*}
```

    then S_L := first $(t'_{L^*} - b_{L^*})$ negative nets of S
    else S_L := first $\max\{0, b'_{L^*} - t_{L^*}\}$ positive nets of S;
  if $D = |R|$
  then if $t'_{R^*} \geq b_{R^*}$
    then S_R := last $(t'_{R^*} - b_{R^*})$ negative nets of S
    else S_R := last $\max\{0, b'_{R^*} - t_{R^*}\}$ positive nets of S;
  $S_M := S - S_L \cup S_R$
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.

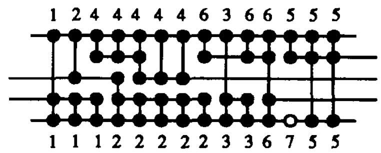
Procedure Distribute_Nets decomposes each net in $L^* \cup R^*$ into two one-sided nets, if it is not itself one sided. It also partitions each net in B into nets of the form $(1, 0)$ and $(0, 1)$. These nets together with the trivial nets in M_T form the set S. Note that the set of new nets obtained 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 every net in L at some column. This is necessary if $D = |L|$, because otherwise the density of the channel is at least $|L| + 1 > D$. It is also possible in this case because $D = |L|$ 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 $D = |B| + 1$, for otherwise the density of the channel would be at least $|B| + 2$. It is possible if $d^* \leq 1$.

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.

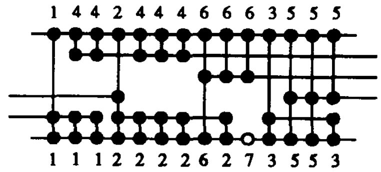
Algorithm: Optimal_Packing ();
  (* $= (N, L, R)$ is an instance of the DM problem *)
  Begin
    Sort L^* into increasing order of net disparities;
    Sort R^* into decreasing order of net disparities;
    if $D = |L|$ **and** $(t'_{L^*} > b_{L^*}$ **or** $b'_{L^*} > t_{L^*})$
      then Select a tag of L^* as its first net;
    if $D = |R|$ **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 ();
    if $t_{L^*} > b_{R^*}$
      then $W := S_L.L'.M_P^-.S_M.M_P^+.R'.S_R$
      else $W := S_L.L'.M_P^+.S_M.M_P^-.R'.S_R$;
    Alternate_Packing (W)
  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 $D = 3 > |L| = 2$ 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

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



(a) An optimal solution



(b) A near-optimal solution

Figure 3: Optimal Packing

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]. □

5 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 DM problem. We first state the correctness of the algorithm, which follows from the correctness of Algorithm Alternate_Packing.

Theorem 5.1 *Algorithm Optimal_Packing computes a valid terminal permutation of a given instance of the DM 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 5.2 *In the channel produced by Algorithm Alternate_Packing, no net in $L^* \cup R^*$ crosses two nets in M_P at some column.*

Lemma 5.3 *If* $L^* = R^* = \phi$, *then Algorithm Optimal_Packing produces a minimum density channel.*

Lemma 5.4 *If* $M_P = \phi$, *then Algorithm Optimal_Packing produces a minimum density channel.*

Based on the above lemmas, we can now show:

Theorem 5.5 *Algorithm Optimal_Packing computes a minimum density channel.*

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 5.6 *Algorithm Optimal_Packing runs in* $O(l)$ *time, where* l *is the length of the channel.*

6 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 6.1 *The WLM problem is NP-complete in the strong sense even for channels with no exits.*

We prove Theorem 6.1 by a polynomial time transformation from the following 3-Partition problem, which is known to be NP-complete in the strong sense [9]. We now have

Corollary 6.2 *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.*

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, "Minimizing channel density by shifting blocks and terminals", *Proc. of the 1991 IEEE International Conference on Computer-Aided Design*, 524-527.

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

[5] Y. Cai and D.F. Wong, *Channel density minimization by pin permutation*, Technical Report #TR-92-09, Department of Computer Sciences, University of Texas at Austin, 1992.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

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

[18] 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.

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

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

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

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