P-17

An Optimal Algorithm for Optimal Routing Around a Rectangle †

Teofilo F. Gonzalez and Sing-Ling Lee
Programs in Mathematical Sciences
The University of Texas at Dallas
Richardson, Texas 75080

ABSTRACT

The problem of connecting a set of terminals that lie on the sides of a rectangle to minimize the total area is discussed. We present an O(n log n) algorithm to solve this problem. Our strategy is to reduce the problem to several subproblems that have the property that no matter what instance we start with, at least one of these subproblems can be solved optimally by the greedy method. If the set of terminals is initially sorted, then the time complexity of our algorithm reduces to O(n α (n)).

I. INTRODUCTION

Let T be a rectangle and S be a set of points on the sides of T. Each point in set S has to be connected by a wire to exactly one point in set S. The wires can be assigned to two different layers. It is assumed that all wires in one layer are parallel to the x-axis and all wires in the other layer are parallel to the y-axis. Wires in different layers can be connected at any given point z by a wire perpendicular to the layers if both wires cross point z in their respective layers. Every pair of distinct and parallel wires must be at least λ units apart and every wire must be at least λ units from each side of T, except in the region where the wire joins the point in S it connects. Also, no wire is allowed inside T in any of the layers.

Problem WR consists of specifying paths for all the connecting wires in such a way that the total area is minimized. That is, to place T together with all the connecting wires (that must satisfy the restrictions imposed above) inside a rectangle of least possible area. This problem has applications in the layout of integrated circuits ([L] and [R]) and conforms to a set of design rules for VLSI systems [MC].

In [HS] and [GLL] an O(n log n) algorithm is presented to solve the WR problem for the case when all the points in S lie in one side of T, where n is the number of elements in set S. In [L] an O(n^3) algorithm is presented to solve the WR problem. If more than two layers are allowed, then the problem becomes NP_hard [SBR]. Other generalizations of the WR problem have also been shown to be NP-hard [La]. In this paper we present an O(n log n) algorithm to solve the WR problem.

Initially our algorithm follows the steps of LaPaugh's algorithm [L]. These steps are the divide-and-conquer and the final assignment of the direction for the connecting paths of all the local terminals (a terminal is said to be local if it is to be connected to another terminal that is located on the same side or on an adjacent side of T). After these steps, it is only required to solve a restricted version of the WR problem. In this restricted version the only non-local terminals appear on the top and bottom sides of T. It is at this point that our algorithm will differ from the one given in [L].

† This work was supported in part by ORG grant A6495-51

First we show the existence of an optimal solution that satisfies the following properties:

a) Balanced. The number of paths crossing two predefined points on the top side of T differs by at most one.

and b) The maximum number of paths crossing any point on the top side of T will not differ by more than one from the minimum, taken over all feasible solutions, of the maximum number of paths crossing any point on the top side of T.

Two suboptimal solutions satisfying conditions (a) and (b) are defined. Both of these solutions can be easily generated. We then show that there is an optimal solution that differs from one of these suboptimal solutions by a set of connecting paths that can be easily characterized. Our algorithm generates these suboptimal solutions and interchanges the direction of several sets of paths producing two solutions. At least one of these solutions will be an optimal solution.

In section II we present some initial definitions and the steps from LaPaugh's [L] algorithm that our procedure follows. Then in section III we present a series of lemmas that show the existence of an optimal solution that can be generated by the greedy method. The final algorithm and complexity issues relating to the WR problem are discussed in section IV.

II. Definitions and Problem Reductions

In this section we redefine the WR problem. Our algorithm can be easily explained under this new definition. We also define some terms and present the steps of LaPaugh's algorithm that our procedure follows.

Let T be a rectangular component of size h by w (height by width). There are 2n terminals (T_1 , T_2 , ... , T_{2n}) on its sides. It is assumed that every pair of terminals is at least λ units apart. The function $C(i)$, for $1 \leq i \leq 2n$, indicates that terminal T_i is to be connected to terminal $T_{C(i)}$. If $C(i) = j$ then $C(j) = i$, i.e., C is a symmetric function. T_i is to be connected to $T_{C(i)}$ by a path that starts at terminal T_i and ends at terminal $T_{C(i)}$. Each connecting path can be partitioned into a finite number of straight line segments. Each of these line segments must lie on the same plane as T, be on the outside of T and be parallel to a side of T. Perpendicular line segments can intersect at any point, but parallel line segments must be at least λ units apart. Also, all line segments must be at least λ units away from every side of rectangle T except in the vicinity where a line segment connects a terminal. The WR problem consists of specifying paths for all the interconnections subject to the rules mentioned above in such a way that the total area is minimized, i.e., place the component together with all the interconnecting paths inside a rectangle of least possible area.

Label the sides of the component (in the obvious way) left, top, right and bottom. Starting in the left-bottom corner of T, traverse the sides of the rectangle clockwise. The ith corner to be visited is labeled S_{i-1}. Assume that the ith terminal visited is terminal T_i . The close interval $[x , y]$, where x and y are the corners of T or the terminals T_i, consists of all the points on the sides of T that are visited while traversing the sides of T in the clockwise direction starting at point x and ending at point y. Parentheses are used instead of square brackets when it is desired to specify an open interval. We use $[S_0 , S_1]$, $[S_1$,

$S_2]$, $[S_2 , S_3]$ and $[S_3 , S_0]$ to represent the left, top, right and bottom sides of T, respectively. Terminal T_i is said to belong to side ℓ, $\underline{S(i)} = \ell$, if T_i is located in $[S_\ell , S_{(\ell+1)mod(4)}]$.

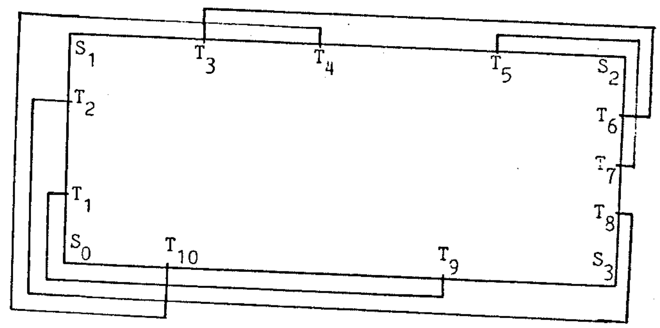

figure 1.1

Set $D = \{d_1 , d_2 ,..., d_k\}$ is said to be an assignment of directions if the cardinality of set D is n and $\{ d_i , C(d_i) \mid 1 \le i \le n \} = \{1,2,...,2n\}$.

D is said to be a partial assignment if the cardinality of set D is $\le$ n,

$$\{ d_i , C(d_i) \mid 1 \le i \le k \} \subseteq \{1 , 2 ,..., 2n\},$$

and $|\{ d_i , C(d_i) \mid 1 \le i \le k \}| = 2 * k$.

An assignment D indicates the direction of the connecting path for all terminals. The connection of terminal T_i, for any i in D, to terminal $T_{C(i)}$ starts at terminal T_i moving perpendicular to side S(i), it then continues moving in the clockwise direction with respect to T until it can be joined to a line segment (all of it on the outside of T) perpendicular to S(C(i)) that starts at terminal $T_{C(i)}$. In a partial assignment, the direction of some connecting paths might not be specified. The assignment for the layout given by figure 1.1 is {3 , 5 , 8 , 9 , 10}. For any $\ell \in$ D, we say that the connecting path for ℓ as given by D crosses point z if $z \in [\ell , C(\ell)]$.

For any assignment (or partial assignment) D we define the layer function H_D for $x,y \in \{T_1,T_2,..., T_{2n}\} \cup \{S_0,S_1,S_2,S_3\}$ as follows:

$H_D(x,y)$ = max{number of paths given by D that cross point z | $z \in [x,y]$}.

Let D be the assignment for the layout shown in figure 1.1. For assignment D we have that $H_D(S_0 , S_1)$ is 3, $H_D(T_5 , T_5)$ is 2 and $H_D(S_2 , S_3)$ is 2. We shall refer to $H_D(x , y)$ as the height of the interval [x , y] in T for assignment D.

The next two lemmas establish that the WR problem reduces to the problem of finding an assignment D with least

$$(H_D(S_1 , S_2) + (H_D(S_3 , S_0)) * (H_D(S_0 , S_1) + (H_D(S_2 , S_3))$$

and then in O(n log n) time (O(n) time if the set of terminals is initially sorted) one can construct a final layout for the connecting paths.

Lemma 2.1: For every assignment D, there is a rectangle Q of size h_Q by w_Q, where

$h_Q = h + (H_D(S_1,S_2)+H_D(S_3,S_0)) * \lambda$ and $w_Q = w + (H_D(S_0,S_1)+H_D(S_2,S_3)) * \lambda$

with the property that rectangle T together with the interconnecting paths defined by D can be made to fit inside Q.

Proof: The proof appears in [L]. []

Lemma 2.2: A final layout with the area given by lemma 2.1 can be obtained in O(n log n) time (O(n) time if the set of terminals is initially sorted) for any assignment D.

Proof: The proof of this lemma appears in [L]. The procedure that constructs the final layout, uses as a subalgorithm the procedures given in [GLL] and [HJ]. []

Terminal T_i is said to be a global terminal if $|S(i) - S(C(i))| = 2$, i.e., terminal T_i is global if it is to be connected to a terminal located on the opposite side of the rectangle. Terminal T_i is said to be local otherwise, i.e., if it is to be connected to a terminal located on the same side or on an adjacent side of the rectangle T. The problem shown in figure 1.1 has T_2 , T_8 , T_4 and T_{10} as the only global terminals. For assignment D we define the function A(D) as

$(h + (H_D(S_1, S_2) + H_D(S_3, S_0)) * \lambda) * (w + (H_D(S_0, S_1) + H_D(S_2, S_3)) * \lambda)$

i.e., the total area required by the layout of T together with all the interconnections specified by D.

Definition 2.1: D'

Let D' be the partial assignment in which all the local terminals are connected by paths crossing the least number of corners of T. []

Lemma 2.3: There is an optimal assignment, D, such that $D' \subseteq D$.

Proof: The proof appears in [L]. []

Lemma 2.3 shows that given any instance of the WR problem it is possible to find an optimal solution in which all local terminals are connected by paths that cross at most one corner of T. The WR problem has been reduced to the problem of finding the direction for the connecting paths of all global terminals in the presence of the partial assignment D'. The next lemma partitions the WR problem into two separate problems: the problem of finding an optimal solution to the WR problem in which all global terminals appear in the top and bottom sides of T and the one in which all global terminals appear in the left and right sides of T. In both these subproblems, local terminals are connected by the paths specified in D'.

Lemma 2.4: D is an optimal assignment if and only if both

$h_Q = h + (H_D(S_1,S_2)+H_D(S_3,S_0)) * \lambda$ and $w_Q = w + (H_D(S_0,S_1)+H_D(S_2,S_3)) * \lambda$

are optimal.

Proof: The proof of this lemma appears in [L]. []

III. The restricted problem

In this section we show that given any instance of the restricted WR problem, it is always possible to obtain an optimal solution by solving one of its subproblems with the greedy method.

From now on we shall restrict our attention to the solution of the WR problem in which all global terminals are located on the top and bottom sides of T and all local terminals are connected by the paths specified in D'. Note that the second subproblem mentioned in the previous section can be transformed to this one by rotating the rectangle 90 degrees. If the number of global terminals located on the top side of T is zero, then D' is an optimal assignment (lemma 2.3). In what follows we assume there is at least one global terminal located on the top side of T.

First the points T_α and T_β will be defined. It will then be shown the existence of an optimal solution in which the global terminals located in the interval $[S_1, T_\alpha)$ and $(T_\beta, S_2]$ are connected as in assignment D" (def 3.2).

<u>Definition 3.1</u>: α and β.

Let

L = D' (_) {all global terminals are connected by a path crossing the left side of T};

$\alpha = \max\{ k \mid H_L(T_k, T_k) = H_L(S_1, S_2) \text{ and } T_k \in [S_1, S_2]\}$;

R = D' (_) {all global terminals are connected by a path crossing the right side of T};

and $\beta = \min\{ k \mid H_R(T_k, T_k) = H_R(S_1, S_2) \text{ and } T_k \in [S_1, S_2]\}$. []

<u>Definition 3.2</u>: D"

Let D" = D' (_) {all global terminals located in the interval $[S_1, T_\alpha)$ connected by a path crossing the left side of T} (_) {all global terminals located in the interval $(T_\beta, S_2]$ connected by a path crossing the right side of T} []

<u>Definition 3.3</u>: α' and β'

Let $\alpha' = H_{D''}(T_\alpha, T_\alpha)$ and $\beta' = H_{D''}(T_\beta, T_\beta)$. []

It will now be shown that there is an optimal assignment such that D" is a subset of it.

<u>Lemma 3.1</u>: Let D" be as defined above. There is an optimal assignment, D, such that $D'' \subseteq D$.

<u>Proof</u>: The proof of this lemma appears in [GL]. []

<u>Definition 3.4</u>: t

Let t be the number of global terminals in the interval $[T_\alpha, T_\beta]$. []

If t = 0 then D" is a complete assignment, i.e., we have specified the direction for the connection of all terminals, and by lemma 3.1 we conclude that D" is an optimal assignment. In what follows we shall assume that $t > 0$.

A lower bound for the height on the top side of T for any optimal solution satisfying the conditions of lemma 3.1 is given by Δ which is defined below.

<u>Definition 3.5</u>: Δ

$$\Delta = \lceil (\alpha' + \beta' + t) / 2 \rceil \quad []$$

In what follows we show that there is an optimal solution satisfying lemma 3.1 whose height on the top side of T is Δ or $\Delta + 1$ and with the property that the number of paths crossing at points T_α and T_β differs by at most one.

Lemma 3.2: There is an optimal solution, D, such that

(a) $D'' \subseteq D$;

(b) $\left| H_D(T_\alpha , T_\alpha) - H_D(T_\beta , T_\beta) \right| \leq 1$

and (c) $\Delta \leq H_D(S_1 , S_2) \leq \Delta + 1$.

Proof: The proof of this lemma appears in [GL]. []

There are instances for which there is no optimal solution satisfying (a) and (b) of lemma 3.2 and whose height on the top side of T is Δ. Also, there are instances that do not have an optimal solution satisfying (a) and (b) of lemma 3.2 and whose height on the top side of T is $\Delta + 1$. When $\alpha' + \beta' + t$ is odd, one can show that there always exists an optimal solution whose height on the top side of T is Δ. The proof of this fact is similar to the one of lemma 3.2. In the next two lemmas we do not take advantage of this fact, however it will be used by the final algorithm.

We now define assignments E_1 and E_2. These assignments have the property that there is an optimal assignment that differs from them by a set of connecting paths that can be easily characterized.

Definition 3.6: E_1 and E_2

Let

$M = \{ \ell \mid T_\ell$ is a global terminal located on the bottom side of T and $\ell , C(\ell) \notin D'' \}$;

$M_1 = \{ \ell \mid \ell$ is in M, ℓ is the ith largest value in M and $i \leq \lceil (\alpha' + \beta' + t) / 2 \rceil - \alpha' \}$;

$M_2 = \{ \ell \mid \ell$ is in M, ℓ is the ith largest value in M and $i \leq \lfloor (\alpha' + \beta' + t) / 2 \rfloor - \alpha' \}$;

$E_1 = D'' \cup \{ \ell \mid \ell \in M_1 \} \cup \{ C(\ell) \mid \ell \in (M - M_1) \}$;

and $E_2 = D'' \cup \{ \ell \mid \ell \in M_2 \} \cup \{ C(\ell) \mid \ell \in (M - M_2) \}$. []

We should point out that if $\alpha' + \beta' + t$ is even then $E_1 = E_2$. However, if this is not the case, then E_1 and E_2 will differ only on the direction of one connecting path.

In what follows we define some terms for assignment E which can be either E_1 or E_2. When we make use of the terms defined this way, it will be explicitly indicated which of E_1 or E_2 was used in the definition.

Definition 3.7: R , T'_ℓ and T''_ℓ

Let $R = H_E(S_1 , S_2) - \Delta$.

For $\ell = 1 , 2 , \ldots , R$, let T'_ℓ (T''_ℓ) represent the terminal located in the interval $[S_1 , S_2]$ with largest (smallest) index whose height is $\Delta + \ell$. Let T'_0 (T''_0) represent T_β (T_α). []

We say that assignment D <u>conforms</u> to E_1 (or E_2) if $H_D(T_\alpha , T_\alpha) = H_{E_1}(T_\alpha , T_\alpha)$ (or $H_D(T_\alpha , T_\alpha) = H_{E_2}(T_\alpha , T_\alpha)$).

It is required to define both E_1 and E_2 because there are instances for which there is no optimal solution that conforms to E_1. The same is also true for E_2. Note that this happens only when $\alpha' + \beta' + t$ is odd.

Let D be an optimal assignment that satisfies the conditions of lemma 3.2 and let $E \in \{E_1 , E_2\}$ conform to D. A connecting path is said to be of <u>type I</u> (<u>type II</u>) if it crosses the right (left) side of T in D but not in E.

<u>Lemma 3.3</u>: There is an optimal solution D such that $E \in \{E_1 , E_2\}$ conforms to it and

a) $D'' \subseteq D$;

b) $\left| H_D(T_\alpha , T_\alpha) - H_D(T_\beta , T_\beta) \right| \leq 1$;

c) $\Delta \leq H_D(S_1 , S_2) \leq \Delta + 1$;

d) all type I (type II) paths connect a terminal located in the interval $[T'_R , T_\beta]$ ($[T_\alpha , T''_R]$)

and e) The number of type I and type II paths is

$\lceil R / 2 \rceil$ if $H_D(S_1, S_2) = \Delta$ and $\lfloor R / 2 \rfloor$ otherwise.

<u>Proof</u>: The proof of this lemma appears in [GL]. []

Sets of terminals will be defined and subsets of them will be labeled A, A', B and B'. This sets will be used in lemma 3.4 where it will be shown that there is always an optimal solution that differs from E_1 or E_2 by the set of paths that connect the terminals labeled A and A' or B and B'. Consequently in order to construct an optimal assignment it is only required to construct E_1 and E_2 and then interchange some set of connecting paths. One of the assignments obtained this way will be an optimal assignment.

<u>Definition 3.8</u>: Q_ℓ , Q'_ℓ , P_ℓ and P'_ℓ.

For $\ell = 1 , 2 , \ldots , R$, let Q_ℓ (Q'_ℓ) be the set of indices of the global terminals located in the interval $(T'_\ell , T_\beta]$ ($[T_\alpha , T''_\ell)$) if ℓ is odd and $[T'_\ell , T_\beta]$ ($[T_\alpha , T'_\ell]$) otherwise, such that these terminals are connected by a path that crosses the left (right) side of T in assignment E.

For $\ell = 1 , 2 , \ldots , R$, let P_ℓ (P'_ℓ) be the set of indices of the global terminals located in the interval $[T'_\ell , T_\beta]$ ($[T_\alpha , T''_\ell]$) if ℓ is odd and $(T'_\ell , T_\beta]$ ($[T_\alpha , T''_\ell)$) otherwise, such that these terminals are connected by a path that crosses the left (right) side of T in assignment E. []

Definition 3.9: A , A' , B and B' labeling.
We shall label terminals as follows:

A labeling:

for ℓ = 1 to R do
  Let f = $\lceil \ell / 2 \rceil - \lceil (\ell - 1) / 2 \rceil$
  Let X = { C(i) | i ∈ P_ℓ and i was not labeled when considering sets $P_1 , P_2 , \ldots , P_{\ell-1}$ }
  Let X' = the f elements of smallest value in set X
  Each index i such that C(i) ∈ X' is labeled A.
endfor

A' labeling:

This labeling procedure is identical to the previous one, except for the sets P_ℓ being replaced by P'_ℓ and "smallest" is replaced by "largest".

B labeling:

for ℓ = 1 to R do
  Let f = $\lfloor \ell / 2 \rfloor - \lfloor (\ell - 1) / 2 \rfloor$
  Let X = { C(i) | i ∈ Q_ℓ and i was not labeled when considering sets $Q_1 , Q_2 , \ldots , Q_{\ell-1}$ }
  Let X' = the f elements in set X with smallest value.
  Each index i such that C(i) ∈ X' is labeled B.
endfor

B' labeling:

This labeling procedure is identical to the previous one, except for the sets Q_ℓ being replaced by Q'_ℓ and "smallest" is replaced by "largest". []

The next lemma establishes that there is an optimal solution which can be obtained by starting from E_1 or E_2 and reversing the connecting path for the terminals labeled A and A' or B and B'.

Lemma 3.4: There is an optimal assignment D, such that E ∈ {E_1 , E_2} conforms to it and

a) D" ⊆ D;

b) $\left| H_D(T_\alpha , T_\alpha) - H_D(T_\beta , T_\beta) \right| \leq 1$;

c) All type I (type II) paths connect a terminal located in the interval $[T'_R , T_\beta]$ $([T_\alpha , T''_R])$;

d) The number of type I and type II paths in D is $\lceil R / 2 \rceil$ if $H_D(S_1, S_2) = \Delta$ and $\lfloor R / 2 \rfloor$ otherwise.

and e) if $H_D(S_1 , S_2) = \Delta$ then all the type I (type II) paths in D connecting a terminal in the interval $[T'_R , T_\beta]$ $([T_\alpha , T''_R])$ connect a terminal with label A (A'). if $H_D(S_1 , S_2) = \Delta + 1$ then all the type I (type II) paths in D connecting a terminal in the interval $[T'_R , T_\beta]$ $([T_\alpha , T''_R])$ connect a terminal with label B (B').

Proof: The proof of this lemma appears in [GL]. []

IV. Algorithm and Complexity Issues

In this section we present our algorithm to solve the WR problem. The algorithm is based on the lemmas presented in sections II and III. Our al-gorithm has worst case time complexity O(n log n) and O(n α (n)) when the set of terminals is initially sorted. In the last part of this section we discuss lower bounds on the worst case time complexity of deci-sion tree algorithms for the WR problem.

We now present the algorithm:

```
algorithm ROUTING
 Rename the set of terminals in such a way that when traversing T in the
   clockwise direction starting at point S0, the terminals are visited
   in the order T1 , T2 , ... , T2n;
 Label the terminals local and global following the definitions that
   appear after lemma 2.2;
 Construct D';                                    //def 2.1//
 Partition the problem into the following two subproblems:
   P1 is the original problem after deleting all global terminals
          appearing in the left and right sides of T, and
   P2 is the original problem after deleting all global terminals
          appearing in the top and bottom sides of T;
 D1 <— SOLVE( P1 );
 D2 <— SOLVE( P2 );
 Combine D1 and D2 into the final assignment D;
 Construct and output the layout for D using the proof of lemma 2.2;
end of algorithm ROUTING;

procedure SOLVE( P );
 Construct D' for P;                          //def 2.1//
 if there are no global terminals then return( D' ) endif;
 Compute α and β;                           //def 3.1//
 Construct D";                                //def 3.2//
 Compute t;                                   //def 3.4//
 if t = 0 then return( D" ) endif;
 Compute Δ ;                                  //def 3.5//
 Construct E1 and E2;          //def 3.6//
 Compute R;                                   //def 3.7//
 Define T'1 , ... , T'R , T"1 , ... , T"R for E1 and E2;   //def 3.7//
 Perform the A , A' , B and B' labelings for E1 and E2;  //def 3.9//
 if α' + β' + t is even then D1 <— MODIFY(A , A' , E1);
                             D2 <— MODIFY(B , B' , E1);
                        else D1 <— MODIFY(A , A' , E1);
                             D2 <— MODIFY(A , A' , E2);
 endif
 return( D1 if A( D1 ) ≤ A( D2 ) and D2 otherwise );
end of procedure

procedure MODIFY( L , L' , E )
 D <— E except that the connecting paths for all terminals labeled L
       and L' is reversed;
 return( D );
end of procedure
```

Theorem 4.1: Algorithm ROUTING solves the WR problem.
Proof: The proof is based on lemmas 2.1 , 2.2 , 2.3 and 3.4. []

Theorem 4.2: The time complexity of procedure ROUTING is $O(n \log n)$.
Proof: The proof of this lemma appears in [GL]. []

Theorem 4.3: The time complexity of procedure ROUTING is $O(n \alpha(n))$ when the set of terminals is initially sorted.
Proof: The proof of this lemma appears in [GL]. []

In [GLL] it was shown that $\Omega(n \log n)$ comparisons are required by any decision tree algorithm that solves the 1-dimensional WR problem. This result holds even when comparisons among linear functions are allowed. A similar result can also be proven for the case when the input to the 1-dimensional WR problem is restricted to terminals located at a distance of at least λ units from each other. Clearly, this result also holds for the WR problem. Hence, the worst case time complexity for the WR problem is $\Theta(n \log n)$.

V. REFERENCES

[AHU] Aho, A. , Hopcroft, J. and Ullman, J. , " The Design and Analysis of Computer Algorithms ", Addison-Wesley, 1975.

[HS] Hashimoto, A. and Stevens J. E., "Wire Routing by Optimizing Channel Assignment without large Apertures", Proceedings of the 8th IEEE Design Automation Conference, 1971, pp 155-169.

[L] LaPaugh, A. S., "A Polynomial Time Algorithm for Optimal Routing around a Rectangle", Proceedings of the 21st IEEE Foundations of Computer Science, 1980, pp 282-29

[La] LaPaugh, A. S., "Algorithms for Integrated Circuit Layout, An Analytic Approach", Ph.D. dissertation, Massachusetts Institute of Technology, 1980.

[GL] Gonzalez, T and Lee, S, "An O(n log n) Algorithm for Optimal Routing Around a Rectangle," Technical Report #116, Programs in Mathematical Sciences, The University of Texas at Dallas, Oct. 1982.

[GLL] Gupta, U. I., Lee D. T. and Leung J., "An Optimal Solution for the Channel-Assignment Problem", IEEE Transactions on Computers, Vol. C-28, No. 11, 1979, pp. 807-810.

[MC] Mead, C. and Conway, L., " Introduction to VLSI Systems ", Addison-Wesley, 1980.

[R] Rivest, R.L., "The PI (Placement and Interconnect) System", Proceedings of the 19th IEEE Design Automation Conference, pp 475-481.

[SBR] Sahni, S., Bhatt, A. and Raghavan, R., "The Complexity of Design Automation Problems", Technical Report 80-23, Department of Computer Science, The University of Minnesota, Dec 1980.