

# New Algorithms for the Rectilinear Steiner Tree Problem

JAN-MING HO, GOPALAKRISHNAN VIJAYAN, AND C. K. WONG, FELLOW, IEEE

**Abstract**—We discuss a new approach to constructing the rectilinear Steiner tree (RST) of a given set of points in the plane, starting from a minimum spanning tree (MST). The main idea in our approach is to find layouts for the edges of the MST, so as to maximize the overlaps between the layouts, thus minimizing the cost (i.e., wire length) of the resulting rectilinear Steiner tree. We describe two algorithms for constructing rectilinear Steiner trees from MST's, that are optimal under the conditions that the layout of each edge of the MST is (1) a L-shape, or (2) any staircase, respectively. The first algorithm has linear time complexity and the second algorithm has a higher polynomial time complexity. Steiner trees produced by the second algorithm have a property called stability, which enables the rerouting of any segment of the tree, while maintaining the cost of the tree, and not causing overlaps with the rest of the tree. Stability is a desirable property in VLSI global routing applications.

## I. INTRODUCTION

GIVEN a set of  $n$  points on the plane, the rectilinear Steiner Tree (RST) problem is to find the rectilinear tree in the plane, of minimum total wire length, which connects the given set of points. The total wire length of a RST is referred to as the *cost* of the RST. RST's have many applications in VLSI physical design. The problem of constructing the minimum cost RST has been shown to be NP-complete [4]. The RST problem has been studied extensively [1], [4], [6]–[8], [10], and many heuristic algorithms have been proposed. Special cases of the RST problem have been shown to have polynomial time algorithms [1], [3].

Hwang [7] has shown that the ratio of the cost of a rectilinear minimum spanning tree (MST) to that of an optimum RST is no greater than  $3/2$ . Therefore, the rectilinear MST is a suitable starting point for deriving low cost RST's. Many heuristic algorithms [8], [10], [11] take this approach. They start with a sequence of the input points and edges as given by a rectilinear MST algorithm, and insert points and edges from this sequence into a growing RST. Local optimization, such as finding the shortest path from a newly added point to a growing RST is performed when inserting a new edge from the rectilinear MST sequence. A main drawback of such heuristics is that they have no provable performance guarantee other than Hwang's general bound. We introduce a more global

approach, in which the rectilinear MST is transformed into an RST, which is optimum under the condition of fixing the shapes of the routes taken by the edges of the MST.

Consider a rectilinear MST  $T$  of a point set  $S$ . A rectilinear shortest path between the end points  $i$  and  $j$  of a MST edge  $e = (i, j)$  is called a staircase layout of the edge. Suppose we select staircase layouts for each MST edge and merge the overlaps among them (if any), the resulting structure will be a RST. To obtain a low cost RST the staircase layouts have to be selected so as to maximize the overlaps among them. Steiner tree heuristics for VLSI global routing, such as in [11], [12], use this idea of trying to maximize the overlaps among the layouts of the MST edges. However, no attempt has been previously made to investigate, whether it is possible, in polynomial time, to produce Steiner trees, which are *optimal*, subject to the restrictions of the approach, i.e., we start with a MST and then lay out the edges of the MST. In this paper, we formalize this approach, and describe two algorithms that produce optimal RST's under differing conditions on the layouts of the MST edges.

The first algorithm restricts the staircase layout of the rectilinear MST edges to L-shapes, and produces the optimal RST under this condition. Given an MST as input, the algorithm produces such a RST in linear time. The second algorithm produces optimal RST's, under the very relaxed condition that arbitrary staircase shapes can be used to layout the MST edges. In fact, we prove that it is sufficient to consider only Z-shaped layouts in order to construct the optimal RST over all arbitrary staircase layouts of the MST edges. This second algorithm has a higher order polynomial time complexity.

We give below some simple definitions that will enable us to discuss the model and algorithms more formally.

**Definition 1 (Underlying Grid):** The underlying grid  $G(S)$  of point set  $S$  (on an oriented plane) is the grid obtained by drawing horizontal and vertical lines through each point of  $S$  (see Fig. 1).

The grid in Fig. 1(a) is the underlying grid of the points  $p_1, p_2, \dots, p_6$ . The optimal RST of a point set  $S$  is known to be a subgraph of its underlying grid  $G(S)$  [5]. The nodes of a RST that are not in  $S$  and have degree at least 3 are called *Steiner points*. The nodes of  $S$  are called *original points*. The length of a vertical or horizontal edge of  $G(S)$  is defined to be the distance between its end-points. The *rectilinear distance*  $\text{dist}(i, j)$  on the oriented plane between two points  $i$  and  $j$  on the grid  $G(S)$  is equal

Manuscript received March 20, 1989. This paper was recommended by Associate Editor R. H. J. M. Otten.

J.-M. Ho is with the Institute of Information Science, Academia Sinica, Nankung, Taipei, Taiwan, Republic of China.

G. Vijayan and C. K. Wong are with IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

IEEE Log Number 8931877.

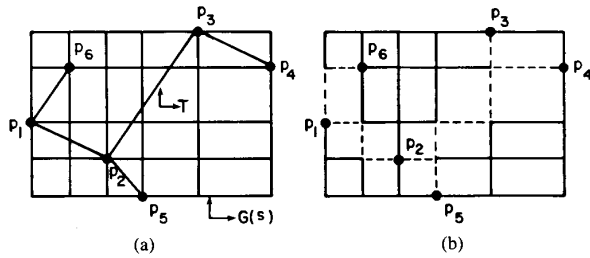


Fig. 1. Underlying grid, MST, and edge layouts.

to the length of any shortest path between the points on the grid  $G(S)$ . The *cost of a RST* is the sum of the lengths of the edges of  $G(S)$  that are used in the RST.

**Definition 2 (Rectilinear MST):** A *rectilinear MST* of a point set  $S$  on an oriented plane is a MST of the complete graph  $G_c(S)$  of the point set, in which the weight of each edge  $(i, j)$ ,  $i \in S, j \in S$  is the rectilinear distance  $\text{dist}(i, j)$ .

**Definition 3 (Edge Layouts):** A *staircase layout* of an edge  $(i, j)$  of a rectilinear MST  $T$  of a point set  $S$  is a shortest path between the points  $i$  and  $j$  on the underlying grid  $G(S)$ . A staircase layout having at most two turns on the grid  $G(S)$  is defined to be a *Z-shaped layout*. A staircase layout having at most one turn is defined to be a *L-shaped layout*.

The tree  $T$  of Fig. 1(a) is a rectilinear MST of the points  $p_1, p_2, \dots, p_6$ . The layouts of the edges of the MST  $T$  shown (as dashed paths) in Fig. 1(b) are all staircase layouts. In particular the layout of the edge  $(p_1, p_2)$  is a Z-shaped layout, and that of  $(p_1, p_6)$  is a L-shaped layout.

**Definition 4 (Degenerate Edge):** An edge between two points  $i$  and  $j$  of a rectilinear MST of a point set is said to be *degenerate* if  $i$  and  $j$  lie on the same horizontal line or on the same vertical line. A degenerate edge has a unique staircase layout.

The number of staircase layouts of an edge is an exponential function of the number of points on the subgrid within the bounding rectangle of the two endpoints of the edge. Choosing different staircase layouts of the edges of the MST will result in different amounts of overlaps among these edge layouts. If we merge these overlapping portions, and introduce appropriate Steiner points, the resulting structure is a RST of the given point set. Depending on the types of layouts we allow for the MST edges, we define the following different types of such RST's.

**Definition 5 (S-RST, Z-RST, L-RST):** An RST that is obtained from a MST  $T$  of a point set  $S$ , by using staircase layouts on  $G(S)$  for the edges of the MST is referred to as an *S-RST* of  $T$ . An S-RST of  $T$ , in which the layout of each MST edge is a Z-shaped layout is called a *Z-RST* of  $T$ . A S-RST of  $T$ , in which the layout of each MST edge is a L-shaped layout is called a *L-RST* of  $T$ . An optimal

S-RST (Z-RST, L-RST) is an S-RST (Z-RST, L-RST) of the least cost among all S-RST's (Z-RST's, L-RST's).

Fig. 2 shows a rectilinear MST of a set of 8 points and two different S-RST's of the MST. The points  $p_1, p_2, \dots, p_8$  are the points in the given set. The points  $s_1, s_2, s_3, s_4$  are Steiner points. The S-RST  $\Psi_1$  has a lower cost than  $\Psi_2$ .

Optimal S-RST's have an interesting and useful property called *stability under rerouting*. We describe this below. A *staircase segment* of an S-RST is a path between two points (original or Steiner) such that this path does not contain any other original or Steiner points. The operation of replacing a staircase segment by another staircase segment between the same endpoints is called *staircase rerouting*. Staircase rerouting may cause overlaps. Merging the overlaps results in a new S-RST of strictly lower cost. For example, consider the two S-RST's  $\Psi_1$  and  $\Psi_2$  of Fig. 2. *Rerouting* any staircase in the S-RST  $\Psi_1$  does not cause overlaps. In the S-RST  $\Psi_2$ , the staircase segment (actually L-shaped) between  $s_1$  and  $p_3$  can be rerouted to cause overlaps with the staircase between  $p_3$  and  $p_4$ . In fact when this overlap is merged, the resulting lower cost S-RST has the same cost as  $\Psi_1$ . An S-RST is said to be *stable under rerouting* if the staircase rerouting of any subset of its staircase segments does not cause any overlaps. In Fig. 2, the S-RST  $\Psi_1$  is stable, while the S-RST  $\Psi_2$  is not. *Stability under rerouting* is a desirable property for Steiner trees of nets in VLSI global routing [11], because it would enable a segment of the Steiner tree of a net to be rerouted without causing any overlaps with other segments of the same Steiner tree. Such reroutings may be necessary, in order to alleviate congestions in the underlying global routing grid. Also, note that a stable S-RST corresponds to a local optimum under the operation of staircase rerouting. An optimal S-RST must be stable, otherwise a lower cost S-RST may be derived by rerouting some of the staircase segments and merging the resulting overlaps, which is a contradiction. From the Z-shape sufficiency result in Section V, an optimal Z-RST is also a optimal S-RST. Therefore, optimal Z-RST's are also stable. However, note that rerouting some staircase segments in a Z-RST may result in S-RST in which one or more of the original MST edges, no longer have Z-shaped layouts. The stability property does not hold for optimal L-RST's. Flipping an L-shaped segment of an optimal L-RST can cause overlaps with other portions of the L-RST.

Our approach uses rectilinear MST's that have a special property, that for any pair of nonadjacent edges in the rectilinear MST, any staircase layouts of the two edges will not intersect or overlap. This property is called the *separability*, and a rectilinear MST which has this property is called a *separable MST* (or SMST for short). When we delete an edge from a SMST, the two resulting subtrees cannot intersect or overlap each other. Overlaps can occur only between edges that are incident on a common point. This property enables the use of dynamic programming techniques. Separability is an interesting property

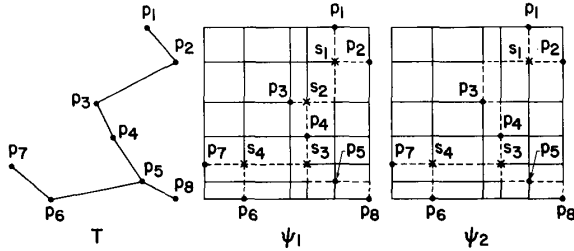


Fig. 2. A rectilinear MST and two S-RST's.

by itself, and may have other interesting applications. In Section II we present an algorithm for constructing a separable MST of any set of  $n$  points.

In Section III we prove that the degree of any point in a rectilinear MST is at most 8, and that at most 6 edges incident on a common point can be nondegenerate. We also show that the neighbors of a point in a rectilinear MST are restricted to lie in specific areas. The degree boundedness of a rectilinear MST is used to bound the worst-case time complexities of our two algorithms.

In Section IV we present a linear time algorithm for constructing an optimal L-RST from a given SMST. In Section V we show that Z-shaped layouts of the edges are sufficient for the case of general staircase layouts. In Section VI we describe a polynomial time algorithm for constructing an optimal Z-RST from a SMST, which by the sufficiency theorem is also an optimal S-RST. As much as is possible, we have tried to describe the two algorithms in a manner so that they can be read and understood without having to read the proofs of the various lemmas and theorems of the paper. Experimental results of the two algorithms are presented in Section VII.

In comparing the two algorithms, the first for optimal L-RST and the second for optimal S-RST, we note that the first algorithm has a much lower running time than the second. However, staircase layouts are much more general than L-shaped layouts, therefore the second algorithm produces lower cost RST's. In addition, the RST's trees produced by the second algorithm have the property of being stable under rerouting.

## II. SEPARABLE MST

The input to the two RST algorithms described in this paper is a separable rectilinear MST (SMST) of a given point set  $S$ .

**Definition 6 (Separable MST (SMST)):** A rectilinear MST of a point set is said to be separable if arbitrary staircase layouts of any two nonadjacent edges of the tree do not intersect or overlap.

Consider a nondegenerate edge  $e$  of a rectilinear MST between the points  $i$  and  $j$  of the given point set. Any staircase layout of the edge  $e$  will be contained within the smallest rectangle enclosing the two points  $i$  and  $j$ . We call this smallest rectangle the *enclosing box* of the edge  $e$ . Fig. 3(a) shows the enclosing boxes of two edges  $e_1$

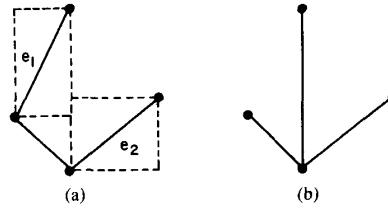


Fig. 3. A nonseparable MST and a separable MST.

and  $e_2$ . It is obvious that a rectilinear MST is separable if and only if the intersection of the enclosing boxes of any pair of nonadjacent edges is empty. The property that the layouts of two nonadjacent edges of a SMST cannot interact with each other makes it possible to design polynomial time dynamic programming algorithms for both the L-RST and Z-RST problems. Not all rectilinear MST's are separable. In Fig. 3 we show a nonseparable and a separable MST of the same set of points. The first MST shown in the figure is nonseparable because the enclosing boxes of the edges  $e_1$  and  $e_2$  overlap.

We now present an  $O(n^2)$  algorithm for constructing a separable MST of any given point set. Its correctness, proved in Theorem 1, implies that any point set always has a MST which is separable. It is possible to derive an  $O(n \log n)$  algorithm using the notion of *Voronoi diagrams* [9]. We wish to describe the  $O(n^2)$  algorithm in this paper, because it is easy to implement being a minor modification of the well-known MST algorithm of Prim [2]. We use  $x(i)$  and  $y(i)$  to denote the  $x$ - and  $y$ -coordinates of a point  $i$ .

### Algorithm SMST

- 1) Construct the complete graph  $G_c(S)$  of the point set  $S$ .
- 2) For each edge  $e = (i, j)$  of  $G_c(S)$ , construct the three-tuple

$$3\text{tuple}(i, j) = (\text{dist}(i, j), -|y(i) - y(j)|, -\max(x(i), x(j)))$$

and assign it to be the weight of the edge.

- 3) Run Prim's MST algorithm [2] on  $G_c(S)$  using the three-tuples as the weights, to generate a MST  $T_s(S)$ . Output  $T_s(S)$  as the separable MST of the point set  $S$ .

Note that in comparing two 3-tuples, we have  $(a_1, b_1, c_1) < (a_2, b_2, c_2)$ , if and only if  $(a_1, b_1, c_1)$  precedes  $(a_2, b_2, c_2)$  in nondecreasing lexicographic order. The first element of each three-tuple is the rectilinear length of the corresponding edge, and therefore Prim's algorithm will indeed produce a MST of the graph  $G_c(S)$ . The remaining two elements of the three-tuple are used only to break ties among edges of the same rectilinear length. The time complexity is dominated by the execution of Prim's algorithm, which takes  $O(n^2)$  time. Hence, we have the following lemma:

**Lemma 1:** The tree  $T_s(S)$  constructed by Algorithm SMST is a rectilinear MST of the point set  $S$ . The time complexity of Algorithm SMST is  $O(n^2)$ .

It remains to show the tree  $T_s(S)$  has the separability property.

**Theorem 1 (Separability Theorem):** The rectilinear MST  $T_s(S)$  generated by Algorithm SMST is separable.

*Proof:* The proof is by contradiction. Suppose there exists nonadjacent edges  $e_1 = (i_1, j_1)$  and  $e_2 = (i_2, j_2)$  in  $T_s(S)$ , whose enclosing boxes  $B_1$  and  $B_2$ , respectively, have a nonempty intersection. Without loss of generality, assume that  $i_1$  and  $i_2$  both belong to the same connected component of  $T_s(S) - \{e_1, e_2\}$ .

First, we claim that the points  $i_2$  and  $j_2$  do not lie either on the boundary or in the interior of  $B_1$ . Suppose not, then let  $i_2$  be on the boundary or interior of  $B_1$ . In this case, replacing the edge  $e_1$  by the edge  $(i_2, j_1)$ , results in a rectilinear spanning tree whose cost is strictly less than that of  $T_s(S)$ . This is a contradiction to Lemma 1 that  $T_s(S)$  is a MST. A similar argument holds when  $j_2$  is on the boundary or interior of  $B_1$ . Similarly the two points  $i_1$  and  $j_1$  do not lie on the boundary or interior of  $B_2$ .

We may assume  $x(i_1) \geq x(i_2)$  else exchange the point labels  $i_1, i_2$  and point labels  $j_1, j_2$ . Since the elements of the three-tuple weights are invariant under reflection across the  $x$ -axis, we may also assume that  $y(i_1) \geq y(i_2)$ . Since the endpoints  $e_1(e_2)$  cannot lie on  $B_2(B_1)$ ,  $x(j_1) \neq x(i_2)$ . If we assume  $x(j_1) > x(i_2)$ , there remain six possible intersection patterns between the enclosing boxes  $B_1$  and  $B_2$  as depicted in Fig. 4. If  $x(j_1) < x(i_2)$ , then we have six similar patterns obtained by rotating the patterns of Fig. 4 by  $180^\circ$  and exchanging labels  $i_1, i_2$  and labels  $j_1, j_2$ . A similar argument as in the following will hold for these patterns.

Symbols  $a_1, b_1, a_2, b_2, \alpha, \beta$  are used to denote the lengths marked in the figure. Since  $T_s(S)$  is a MST and  $i_1$  and  $i_2$  are in the same component of  $T_s(S) - \{e_1, e_2\}$ , the inequalities  $\text{dist}(i_1, j_2) \geq \text{dist}(i_2, j_2)$  and  $\text{dist}(i_2, j_1) \geq \text{dist}(i_1, j_1)$  must hold. In cases (a) and (f) they are equivalent to  $a_1 + b_2 \geq a_2 + \alpha + \beta + b_2$  and  $a_2 + b_1 \geq a_1 + \alpha + \beta + b_1$  implying that  $a_1 = a_2$ , and  $\alpha = \beta = 0$ . In cases (b) and (c) we can deduce that  $a_1 = a_2$  and  $\beta = 0$ . In cases (d) and (e) we can only deduce that  $a_1 = a_2$ . The six cases thus reduce to corresponding six cases of Fig. 5.

In cases (a)–(e) of Fig. 5 we have  $\text{dist}(i_1, j_2) = \text{dist}(i_2, j_2)$  and  $-|y(i_1) - y(j_2)| < -|y(i_2) - y(j_2)|$  which implies that  $3\text{tuple}(i_1, j_2) < 3\text{tuple}(i_2, j_2)$ . Replacing the edge  $(i_2, j_2)$  by a new edge  $(i_1, j_2)$  in the tree  $T_s(S)$  results in a spanning tree of strictly lesser 3-tuple weight than  $T_s(S)$  which is a contradiction. In case (f) we have  $\text{dist}(i_1, j_2) = \text{dist}(i_2, j_2)$  and  $-|y(i_1) - y(j_2)| = -|y(i_2) - y(j_2)|$ . However,  $-\max(x(i_1), x(j_2)) < -\max(x(i_2), x(j_2))$ . Therefore,  $3\text{tuple}(i_1, j_2) < 3\text{tuple}(i_2, j_2)$  is still true and a contradiction follows. This completes the proof of the theorem.  $\square$ .

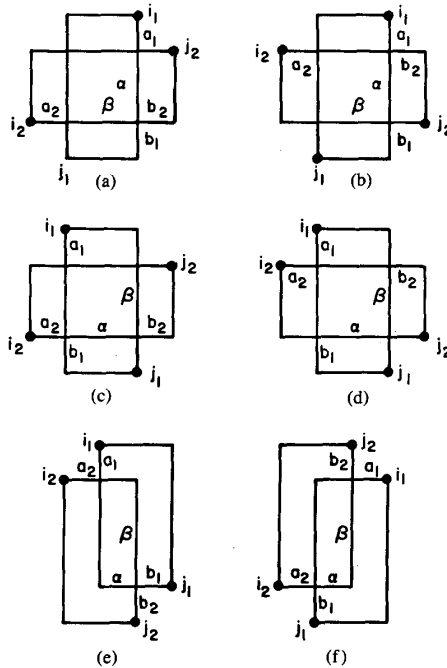


Fig. 4. Intersection patterns between two enclosing boxes—I.

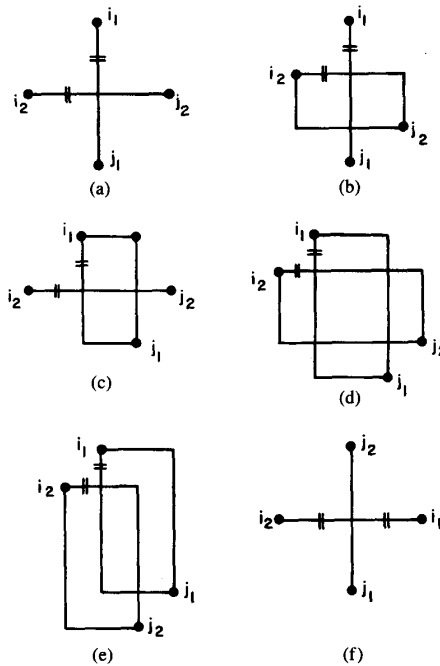


Fig. 5. Intersection patterns between two enclosing boxes—II.

### III. DEGREE BOUNDEDNESS OF RECTILINEAR MST'S

We now show that there are restrictions on the location and number of the neighbors of any point in a rectilinear MST of a point set.

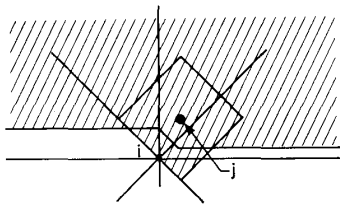


Fig. 6. Location lemma.

**Lemma 2 (Location Lemma):** Let  $T$  be a rectilinear MST of the point set  $S$ , and let  $i \in S$ . There is at most one neighbor in  $T$ , of  $i$ , in the interior of each of the 4 regions formed by the two  $\pm 45^\circ$  lines passing through the point  $i$ .

*Proof:* Let  $j$  and  $k$  be two neighbors of a point  $i$  in  $T$ . Since  $T$  is a rectilinear MST, the inequalities  $\text{dist}(k, j) \geq \text{dist}(i, j)$  and  $\text{dist}(k, j) \geq \text{dist}(i, k)$  must be true. Suppose  $j$  is located in the upper quadrant formed by the two  $\pm 45^\circ$ -lines passing through the point  $i$ , as shown in Fig. 6. The horizontal jagged line in the figure separating  $i$  and  $j$  is the rectilinear bisector of  $i$  and  $j$ . Any point on this line is equidistant from  $i$  and  $j$ . If  $k$  is a point above this bisector, then  $\text{dist}(i, k) > \text{dist}(k, j)$ . Therefore, no neighbor of  $i$  other than  $j$  can be located above this bisector. The square shown in the figure is centered at the point  $j$  with rectilinear radius  $\text{dist}(i, j)$ . Any point on the boundary of this square is at a rectilinear distance of  $\text{dist}(i, j)$  from  $j$ . If  $k$  is a point in the interior of this square then,  $\text{dist}(i, j) > \text{dist}(k, j)$ . Therefore, no neighbor of  $i$  other than  $j$  can be located in the interior of this square. Thus, the shaded region shown in Fig. 6 is a forbidden region for any other neighbor of  $i$ . Since this shaded region includes the upper quadrant formed by the two  $\pm 45^\circ$  lines through  $i$ , there can be at most one neighbor of  $i$  in this quadrant. The proof of the lemma follows by symmetry.  $\square$

**Lemma 3 (Degree Boundedness):** Let  $T$  be a rectilinear MST of the point set  $S$ , and let  $i \in S$ . Then the point  $i$  has at most 8 neighbors in  $T$ , and at most 6 of these neighbors form nondegenerate edges with the point  $i$ .

*Proof:* By Lemma 2 there are at most 4 neighbors of  $i$  which are not located on the two  $\pm 45^\circ$  lines. It is easy to see that no two neighbors of  $i$  can be located on the same side of  $i$  on the two  $\pm 45^\circ$  lines. Therefore,  $i$  can have no more than 8 neighbors. We leave it to the reader to establish that at most 6 of these neighbors can be present outside the two  $\pm 90^\circ$  lines.  $\square$

#### IV. ALGORITHM FOR OPTIMAL L-RST

We now present a linear time algorithm, called *Algorithm Optimal-L-RST*, for obtaining an optimal L-RST from a given separable MST  $T$  of a point set  $S$ . The algorithm has to select for each nondegenerate edge one of its two possible L-shaped layouts, so as to maximize the

total amount of overlap, thus minimizing the total cost of the resulting L-RST.

Hang the input separable MST  $T$  by any leaf edge  $r$  resulting in the rooted tree  $T_r$ . For each edge  $e$  in  $T$ , let  $T_e$  denote the subtree of  $T_r$  that hangs by the edge  $e$ . Let  $e = (a, b)$  with the point  $b$  being a child of  $a$  in the rooted tree  $T_r$ . The other edges of  $T_r$  incident on point  $b$  are called the *child edges* of the edge  $e$ . For example, in Fig. 2, suppose the tree is hung by the leaf edge  $(p_8, p_5)$  then the edges  $(p_5, p_4)$  and  $(p_5, p_6)$  are child edges of the edge  $(p_8, p_5)$ .

Let  $L(e)$  and  $U(e)$  denote the upper and lower L-shaped layouts, respectively, of any edge  $e$ . Let  $\Psi_L(e)$  ( $\Psi_U(e)$ ) denote the L-RST of the subtree  $T_e$ , which has *minimum cost* among all L-RST's of  $T_e$ , in which the layout of the edge  $e$  is *constrained* to be the L-shape  $L(e)$  ( $U(e)$ ).

$\Psi_L(e)$  can be computed recursively as follows: let  $e_i$ :  $i = 1, 2, \dots, d$  be the  $d$  child edges of  $e$  in the rooted tree  $T_r$ . For each child edge  $e_i$  and for each of the two possible L-shaped layouts  $L(e_i)$  and  $U(e_i)$  of the edge  $e_i$ , recursively compute the constrained optimal L-RST's  $\Psi_L(e_i)$  and  $\Psi_U(e_i)$  of the subtrees  $T_{e_i}$ . Taking one of these L-RST's for each subtree  $T_{e_i}$ , and merging these subtree L-RST's with the layout  $L(e)$  of the edge  $e$  results in a L-RST of  $T_e$ . Since the tree  $T$  has the separability property, the only new overlaps that can occur during this merging are among the edges  $e, e_1, \dots, e_d$ , which are all incident on a common point. Therefore, the total amount of overlap in the resulting L-RST of  $T_e$  is the sum of the overlaps among the layouts of the edges  $e, e_1, \dots, e_d$ , added to the sum of overlaps in the selected L-RST's of the subtrees  $T_{e_i}$ . Enumerate all combinations of selecting one of the two L-RST's  $\Psi_L(e_i)$  or  $\Psi_U(e_i)$  for each subtree  $T_{e_i}$ . For each such combination compute the resulting L-RST of  $T_e$ . The constrained optimal L-RST  $\Psi_L(e)$  of the subtree  $T_e$  is simply the one with the least cost.  $\Psi_U(e)$  can be computed in a similar fashion.

To compare the optimal L-RST of the entire rooted tree  $T_e$ , recursively compute (as explained above) the constrained optimal L-RST's  $\Psi_L(r)$  and  $\Psi_U(r)$ , and select the one with the smaller cost.

The proof of correctness of the above algorithm follows the separability of the input tree  $T$ . The linear time complexity of the algorithm can be established as follows: From Lemma 3, we know that the degree of any point in the SMST  $T$  is at most 8, and at most 6 of the incident edges are nondegenerate. Therefore, the total number of combinations enumerated in the computations of  $\Psi_L(e)$  and  $\Psi_U(e)$  for any edge  $e$  is at most  $2^6$ , which is a constant. The new overlaps occurring in each combination can be processed in constant time. Since there are  $n - 1$  edges in  $T$ , the above algorithm has linear time complexity. Hence we have the following theorem.

**Theorem 2:** Algorithm Optimal-L-RST constructs the optimal L-RST of the SMST  $T$  of a point set  $S$  of cardinality  $n$  in  $O(n)$  time.

### V. SUFFICIENCY OF Z-SHAPED LAYOUTS

We now prove that there exists a Z-RST whose cost is equal to the cost of the optimal S-RST. Thus Z-shaped layouts are *sufficient* to consider when constructing an optimal S-RST. As shown in Fig. 7, an optimal L-RST may have strictly lower cost than an optimal S-RST thus making Z-shaped layouts necessary.

Let  $l(e)$  denote the number of edges on the underlying grid  $G(S)$  traversed by any staircase layout of an edge  $e$  of a MST  $T$  of a point set  $S$ . If the enclosing box of  $e$  has  $w(e)$  grid edges on its horizontal side and  $h(e)$  edges on its vertical side, then  $l(e) = w(e) + h(e)$ . Note that  $l(e)$  is independent of the actual staircase layout chosen for  $e$ . The number of different staircase layouts of a non-degenerate edge  $e$ , in the worst case, could be exponential in  $l(e)$ . Considering only Z-shaped layouts of  $e$ , observe that starting at one of the endpoints, a Z-shaped layout can go either horizontally and make its first turn at any one of the  $w(e)$  points, or go vertically and turn at any one of the  $h(e)$  points. After making a turn the layout must go directly to the opposite side and make the second turn (if any) towards the other endpoint. Thus there are  $l(e) = w(e) + h(e)$  choices to make the first turn which implies that there are  $l(e)$  distinct Z-shaped layouts. This observation together with the sufficiency result enables the design of a polynomial time algorithm for constructing the optimal S-RST. This algorithm is described in Section VI.

We first introduce several notations. Given an edge  $e = (i, j)$  of a SMST  $T$ , the angle formed by the intersection of the line  $\overline{ij}$  and a horizontal line is denoted as  $\theta(i, j)$ . Let  $R(i, j)$  denote the enclosing box of the two points  $i$  and  $j$ . Let  $R_I(i, j)$  and  $R_B(i, j)$  denote the interior and the boundary respectively of  $R(i, j)$ . In a SMST  $T$ , only adjacent edges can overlap and cause Steiner points to be generated in a S-RST, and thus a Steiner point  $s$  in a S-RST must lie on the layouts of at least two edges of the SMST, which are incident on a common point, say  $k$ . We say then that the Steiner point  $s$  is *adherent* to the original point  $k$ . The layout of the edge  $e = (i, j)$  in a S-RST is denoted as  $P(i, j)$ .

Given two points  $i$  and  $j$  of the input set  $S$ , one can partition the plane into nine regions, which are labeled as illustrated in Fig. 8, formed by the horizontal and vertical lines passing through the two points  $i$  and  $j$ . We will be referring to these regions in the following discussion.

**Lemma 4:** Let  $e = (i, j)$  be an edge of a SMST  $T$  of a point set  $S$  with  $45^\circ \leq \theta(i, j) < 90^\circ$  and  $x(i) > x(j)$ ,  $y(i) < y(j)$ . For any optimal S-RST of  $T$  the set  $A$  of Steiner points in  $P(i, j) \cap R_I(i, j)$  satisfies the following propositions:

- 1) Each point of  $A$  is either adherent to  $i$  or to  $j$ .
- 2) If  $s \in A$  is on the layout  $P(b, i)$  of the edge  $(b, i)$  of  $T$  (or  $P(b, j)$  of edge  $(b, j)$ ) then the point  $b$  of  $S$  lies in region IV (or region II). Moreover the portion of the layout  $P(b, i)$  ( $P(b, j)$ ) that lies in  $R_I(i, j)$  must be horizontal.

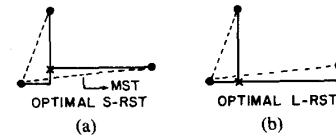


Fig. 7. Insufficiency of L-shaped layouts.

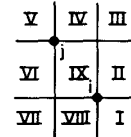


Fig. 8. Labeling of the nine regions.

- 3) The Steiner set  $A$  has at most two points. If so, one (say)  $s_i$  must adhere to  $i$  and the other (say)  $s_j$  must adhere to  $j$ . Moreover the portion of  $P(i, j)$  between  $s_i$  and  $s_j$  is vertical.

*Proof:*

- 1) Since  $T$  is separable, only the layout of an edge incident to either  $i$  or  $j$  can intersect with the layout of  $e$ . Thus any Steiner point on  $P(i, j)$  must be adherent to either  $i$  or  $j$ .
- 2) Suppose a Steiner point  $s \in A$  also lies on  $P(b, i)$ , where  $b$  is a neighbor of  $i$ , then  $s$  must be in  $R(b, i)$ , which forces  $b$  to be in either region IV or region VI. However, the *Location Lemma* (Lemma 2) of Section III requires  $b$  to be in region VI. In addition, it is easy to see that the optimality of the S-RST forces the portion of  $P(b, i)$  in  $R_I(i, j)$  to be on a horizontal line segment. Similar argument holds when  $b$  is a neighbor of  $j$ .
- 3) Since  $45^\circ \leq \theta(i, j) < 90^\circ$ , by the *Location Lemma* (Lemma 2) there can be at most one neighbor of  $i$  in region VI. Similarly there is at most one neighbor  $j$  in region II. Since each Steiner point in the set  $A$  must be adherent to either  $i$  or  $j$ , we have  $|A| \leq 2$ . It also follows that if  $|A| = 2$ ,  $A = \{s_i, s_j\}$  then  $s_i$  is adherent to  $i$  and  $s_j$  is adherent to  $j$ . In addition, the optimality of the S-RST forces the segment between  $s_i$  and  $s_j$  to be vertical.  $\square$

**Theorem 3 (Z-Sufficiency Theorem):** Given a SMST  $T$  of a point set  $S$  of cardinality  $n$ , there exists a Z-RST of  $T$  whose cost is equal to the cost of an optimal S-RST of  $T$ .

*Proof:* We claim that for any integer  $k$ ,  $0 \leq k \leq n - 1$ , there is an optimal S-RST  $\Psi_k$  of  $T$  in which all edges of a  $k$ -edge subtree of  $T$  is laid out using only Z-shapes. The proof of this claim is by induction. Clearly the basis, when  $k = 0$  is obviously true. Assume that there is an optimal S-RST in which a subtree  $T_{k-1}$  of  $T$ ,  $k > 0$ , is implemented using only Z-shaped layouts for its  $k - 1$  edges. Let  $e = (i, j)$  be an edge in  $T$  such that  $j$  is in the subtree  $T_{k-1}$ , and  $i$  is outside this subtree. We will now show that the optimal S-RST  $\Psi_{k-1}$  can be modified into

another optimal S-RST  $\Psi_k$  in which the subtree  $T_k = T_{k-1} \cup e$  is implemented using only Z-shaped layouts for its  $k$  edges. Note that in the case of  $k = 1$ , the subtree  $T_{k-1}$  is simply the isolated point  $j$ , and has zero edges.

If the layout of the edge  $e$  in the S-RST  $\Psi_{k-1}$  is either degenerate, or already a Z-shape, then we are done. So let us assume otherwise. Without loss of generality, we assume  $y(i) < y(j)$ ,  $x(i) > x(j)$ , and  $45^\circ \leq \theta(i, j) < 90^\circ$ . The remaining cases are covered by the fact that the cost of an S-RST is invariant under rotation by a multiple of  $90^\circ$  or by reflection across any vertical or horizontal line.

Let  $\bar{s}_j$  be the Steiner point on  $R_B(i, j)$  which is the farthest away from  $j$  among all such Steiner points *adherent* to  $j$ . If there are no Steiner points adherent to  $j$  on  $R_B(i, j)$  we let  $\bar{s}_j = j$ . Let  $\bar{s}_i$  be similarly defined. Let  $A$  denote the set of Steiner points in  $R_I(i, j)$ . Since by Lemma 4,  $|A| \leq 2$ , we have the following three cases on the cardinality of  $A$ . In each of the cases, we show how to alter the layout of the edge  $e = (i, j)$  into a Z-shaped layout. For convenience, we let  $L(x, y)$  denote the lower L-shape between the two points  $x$  and  $y$ , and  $U(x, y)$  to be the upper L-shape.

- 1)  $|A| = 0$ : Use any one of the L-shapes  $L(\bar{s}_i, \bar{s}_j)$  or  $U(\bar{s}_i, \bar{s}_j)$  for laying out the portion of the edge  $e$  between  $\bar{s}_i$  and  $\bar{s}_j$ . Different cases on the locations of  $\bar{s}_i$  and  $\bar{s}_j$  are shown in Fig. 9. The layout of the edge  $e = (i, j)$  is now a Z-shape.
- 2)  $|A| = 1$ : Without loss of generality assume that  $s_i \in A$  is adherent to  $i$ . Since  $\Psi_{k-1}$  is optimal  $\bar{s}_j$  and  $s_i$  must lie on the same vertical line. There are two subcases depending on the location of  $\bar{s}_i$ .

(a)  $\bar{s}_i$  is on the bottom side of  $R_B(i, j)$ : Use the lower L-shape  $L(\bar{s}_i, \bar{s}_j)$  for laying out that portion of the edge  $e$ , between  $\bar{s}_i$  and  $\bar{s}_j$  as shown in Fig. 10(a). The layout of the edge  $e = (i, j)$  is now a Z-shape.

(b)  $\bar{s}_i$  is on the right side of  $R_B(i, j)$ , or  $\bar{s}_i = i$ : Use the upper L-shape  $U(s_i, \bar{s}_i)$  for laying out the portion of the edge  $e$ , between  $s_i$  and  $\bar{s}_i$ , as shown in Fig. 10(b). As seen in the figure, the layout of  $e = (i, j)$  is not yet a Z-shape. Denote the edge, of the SMST  $T$  that shares the Steiner point  $s_i$  with the edge  $e$ , as  $e' = (i, n_i)$ . According to Lemma 4 the portion of the layout of the edge  $e'$  between  $s_i$  and  $n_i$  in  $R_I(i, j)$  is a horizontal line segment. Therefore the vertical segment between  $s_i$  and  $\bar{s}_i$  can be slid to the left until it is flush against the left side of  $R_B(i, j)$  without increasing the cost of the S-RST. This is shown in Fig. 10(c). The layout of the edge  $e = (i, j)$  is now a Z-shape.

- 3)  $|A| = 2$ : Let  $s_i \in A$  be adherent to  $i$  and  $s_j \in A$  be adherent to  $j$ . Let  $e'_i = (i, n_i)$  be the edge of  $T$  which shares the Steiner point  $s_i$  with the edge  $e$ . Similarly, let  $e'_j = (j, n_j)$  be the edge of  $T$  which shares the Steiner point  $s_j$  with the edge  $e$ . According to Lemma 4,  $s_i$  and  $s_j$  lie on a vertical line segment with  $s_j$  on top of  $s_i$ . Also the portion of the layout of the edge

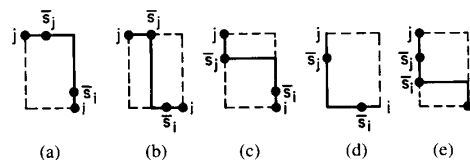


Fig. 9. The case of  $|A| = 0$ .

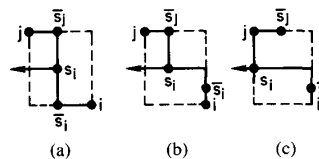


Fig. 10. The case of  $|A| = 1$ .

$e'_i$  between  $s_i$  and  $n_i$  in  $R_I(i, j)$  is a horizontal line segment. Similarly the portion of the layout of the edge  $e'_j$  between  $s_j$  and  $n_j$  in  $R_I(i, j)$  is also a horizontal line segment. There are now 4 subcases depending on the locations of the points  $\bar{s}_i$  and  $\bar{s}_j$ :

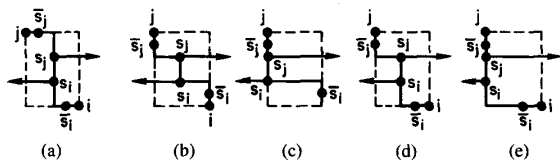
(a)  $\bar{s}_i$  and  $\bar{s}_j$  are on horizontal sides of  $R_B(i, j)$ : Use  $L(s_i, \bar{s}_i)$  and  $U(s_j, \bar{s}_j)$  for laying out the corresponding portions of the edge  $e = (i, j)$ . This is shown in Fig. 11(a). The layout of the edge  $e = (i, j)$  is now a Z-shape.

(b) (1)  $\bar{s}_i = i$ , or  $\bar{s}_i$  is on the right side of  $R_B(i, j)$ , and (2)  $\bar{s}_j = j$ , or  $\bar{s}_j$  is on the left side of  $R_B(i, j)$ : Use  $U(s_i, \bar{s}_i)$  and  $L(s_j, \bar{s}_j)$  as shown in Fig. 11(b). The layout of the edge  $e = (i, j)$  is not yet a Z-shape. According to Lemma 4 the two points  $s_i$  and  $s_j$  are on the same vertical line. Therefore this vertical segment between  $s_i$  and  $s_j$  can be slid until it is flush against the left side of  $R_B(i, j)$  without increasing the cost of the S-RST. This is shown in Fig. 11(c). The layout of the edge  $e = (i, j)$  is now a Z-shape.

(c) (1)  $\bar{s}_i$  is on the lower side of  $R_B(i, j)$ , and (2)  $\bar{s}_j = j$ , or  $\bar{s}_j$  is on the left side of  $R_B(i, j)$ : Use  $L(s_j, \bar{s}_j)$  and  $L(s_i, \bar{s}_i)$  as shown in Fig. 11(d). The layout of the edge  $e = (i, j)$  is not yet a Z-shape. As in the previous subcase, the vertical segment between  $s_i$  and  $s_j$  can be slid until it is flush against the left side of  $R_B(i, j)$  without increasing the cost of the S-RST. This is shown in Fig. 11(e). The layout of the edge  $e = (i, j)$  is now a Z-shape.

(d) (i)  $\bar{s}_i = i$ , or  $\bar{s}_i$  is on the right side of  $R_B(i, j)$ , and (ii)  $\bar{s}_j$  is on the upper side of  $R_B(i, j)$ : Similar to subcase (c).

The alterations of the layout of the edge  $e = (i, j)$  described in all these cases mould the layout of  $e$  into a Z-shape (an L-shape is considered to be a degenerate Z-shape) without any increase in the cost. Thus the resulting S-RST  $\Psi_k$  is optimal and has a subtree  $T_k = T_{k-1} \cup e$  which is implemented using only Z-shapes. This concludes the inductive proof of the claim. Applying the claim for  $k = n - 1$ , we have that the S-RST  $\Psi_{n-1}$  is optimal and uses Z-shaped layouts for all the  $n - 1$  edges of the SMST  $T$ , which completes the proof of the theorem.  $\square$

Fig. 11. The case of  $|A| = 2$ .

## VI. ALGORITHM FOR OPTIMAL S-RST

We now describe *Algorithm Optimal-Z-RST* for constructing an optimal Z-RST, which by Theorem 3 is also an optimal S-RST. The input to the algorithm is a separable MST  $T$  of a point set  $S$ .

Hang the input separable MST  $T$  by any leaf edge  $r$  resulting in the rooted tree  $T_r$ . For each edge  $e$  in  $T$ , let  $T_e$  denote the subtree of  $T_r$  that hangs by the edge  $e$ . Given a Z-shaped layout  $z$  of an edge  $e$ , we let  $\Psi_z(e)$  denote the Z-RST of the subtree  $T_e$ , which has the *minimum cost* among all Z-RST's of  $T_e$ , in which the layout of the edge  $e$  is *constrained* to be the Z-shape  $z$ .  $\Psi_z(e)$  can be computed recursively as follows: Let  $e_i: i = 1, 2, \dots, d$  be the  $d$  child edges of  $e$  in the rooted tree  $T_r$ . For each child edge  $e_i$  and for each possible Z-shaped layout  $z_{ij}$  of the edge  $e_i$ , recursively compute the constrained optimal Z-RST's  $\Psi_{z_{ij}}(e_i)$  of the subtrees  $T_{e_i}$ . Note that there are  $l(e_i)$  such constrained Z-RST's for each subtree  $T_{e_i}$ . Taking one such Z-RST for each subtree  $T_{e_i}$ , and merging these subtree Z-RST's with the layout  $z$  of the edge  $e$ , results in a Z-RST of  $T_e$ . Since the tree  $T$  has the separability property, the only new overlaps that can occur during this merging are among the edges  $e, e_1, \dots, e_d$ , which are all incident on a common point. Therefore, the total amount of overlap in the resulting Z-RST of  $T_e$  is the sum of the overlaps among the layouts of the edges  $e, e_1, \dots, e_d$ , added to the sum of overlaps in the selected Z-RST's of the subtrees  $T_{e_i}$ . Enumerate all combinations of selecting one of the Z-RST's  $\Psi_{z_{ij}}(e_i)$  for each subtree  $T_{e_i}$ , and for each such combination compute the resulting Z-RST of  $T_e$ . The constrained optimal Z-RST  $\Psi_z(e)$  of the subtree  $T_e$  is simply the one with the least cost.

To compute the optimal Z-RST of the entire rooted tree  $T_r$ , recursively compute (as explained above) the constrained optimal Z-RST's  $\Psi_z(r)$  for each Z-shaped layout  $z$  of the root edge  $r$ , and select that Z-RST of the smallest cost. The fact that above algorithm constructs the optimal S-RST follows from the separability of the input MST and from the Z-sufficiency theorem. Regarding time complexity, let  $l_{\max}$  be the maximum of  $l(e)$  over all edges  $e$ . It follows from Lemma 3 that the number of combinations enumerated for each subtree is at most  $l_{\max}^6$ . Each combination can be processed in constant time. Summing up over all the  $n - 1$  subtrees, the worst case time complexity of the algorithm is  $O(n \times l_{\max}^6)$ . Hence we have the following theorem.

**Theorem 4:** Algorithm Optimal-Z-RST constructs the optimal S-RST of the SMST  $T$  of a point set  $S$  of cardinality  $n$  in  $O(n \times l_{\max}^6)$  time.

No. of Points	Mean Ratio of Optimal L-RST to MST costs	Mean Ratio of Optimal S-RST to MST costs
5	.918	.912
10	.915	.910
15	.910	.907
20	.910	.905
25	.910	.905
30	.912	.906
50	.907	.902
100	.903	.898

Fig. 12. Mean ratios of costs.

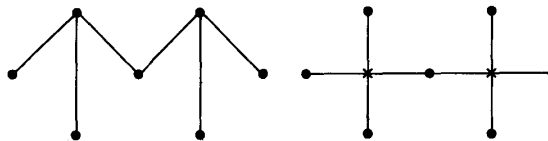


Fig. 13. A MST and its optimal L-RST: Example of 2/3 cost ratio.

## VII. EXPERIMENTAL RESULTS

The algorithms for optimal L-RST and optimal S-RST were implemented using the C language. In both programs, we used the  $O(n^2)$  Algorithm SMST of Section II to generate the separable MST needed as input to the two RST algorithms. The two programs were tested extensively on a large number of random input point sets with cardinalities ranging from 5 to 100. Fig. 12 shows the mean ratios of the cost of L-RST or S-RST over the MST. It is interesting to observe, that on the average, an optimal S-RST has a cost which is only 0.5-percent lower than that of an optimal L-RST. However on some test examples the optimal S-RST the reduction in the cost was observed to be as much as 10 percent.

Hwang [7] has shown that the lowest possible ratio of the cost of a RST to that of the MST of the same set of points is 2/3. However no such results are available for the average ratio for a given number of points. Referring to the table in Fig. 12, we observe that for all cardinalities of the input point sets, the average ratio of the cost of the RST's produced by our approach to the cost of MST is close to 9/10.

Both the algorithms produce the optimum Steiner trees for each member of the class of point sets whose optimal RST has a cost which is 2/3 that of the cost of the MST [7]. One such example is illustrated in Fig. 13.

The algorithm for optimal L-RST is quite fast taking less than 0.01 seconds for 10 input points, on an IBM 3090 processor. Therefore the algorithm is very well suited for VLSI global routing applications. The algorithm for optimal S-RST is only slightly slower in practice. This is because the number of combinations to enumerate for any subtree is usually much less than the worst case of  $l_{\max}^6$ . Empirically its average running time was observed to be between  $O(n^{1.5})$  and  $O(n^2)$ .

### ACKNOWLEDGMENT

The authors wish to thank one of the referees for suggesting simplifications in the separability and Z-sufficiency theorems. They are grateful to both referees for

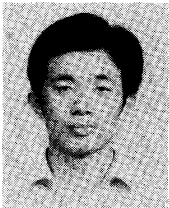


many valuable comments and suggestions, and also wish to thank Majid Sarrafzadeh and Kai-Win Lee for several useful discussions.

#### REFERENCES

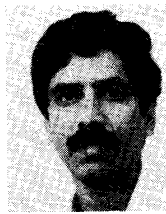
- [1] A. V. Aho, M. R. Garey, and F. K. Hwang, "Rectilinear Steiner trees: Efficient special-case algorithm," *Networks*, vol. 7, pp. 37-58, 1977.
- [2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*. Reading, MA: Addison-Wesley, 1983.
- [3] J. P. Cohoon, D. S. Richards, and J. S. Salowe, "A linear-time Steiner tree routing algorithm for terminals on the boundary of a rectangle," in *Dig. Technical Papers, ICCAD-88*, pp. 402-405, Nov. 1988.
- [4] M. R. Garey and D. S. Johnson, "The rectilinear Steiner tree problem in NP-complete," *SIAM J. Appl. Math.*, vol. 32, no. 4, pp. 826-834, 1977.
- [5] M. Hanan, "On Steiner's problem with rectilinear distance," *SIAM J. Appl. Math.*, vol. 14, no. 2, pp. 255-265, 1966.
- [6] J-M. Ho, G. Vijayan, and C. K. Wong, "A new approach to the rectilinear Steiner tree problem," in *Proc. 26th ACM/IEEE Design Automation Conf.*, pp. 161-166, June 1989.
- [7] F. K. Hwang, "On Steiner minimal trees with rectilinear distance," *SIAM J. Appl. Math.*, vol. 30, no. 1, pp. 104-114, 1976.
- [8] F. K. Hwang, "An  $O(n \log n)$  algorithm for suboptimal rectilinear Steiner trees," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 75-77, Jan. 1979.
- [9] D. T. Lee and C. K. Wong, "Voronoi diagrams in  $L_1$  ( $L_\infty$ ) metrics with 2-dimensional storage applications," *SIAM J. Computing*, vol. 9, no. 1, pp. 200-211, Feb. 1980.
- [10] J. H. Lee, N. K. Bose, and F. K. Hwang, "Use of Steiner's problem in suboptimal routing in rectilinear metric," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp. 470-476, July 1976.
- [11] K-W. Lee and C. Sechen, "A new global router for row-based layout," in *Dig. Tech. Papers, ICCAD-88*, pp. 180-183, Nov. 1988.
- [12] J. G. Xiong, "Algorithms for global routing," in *Proc. 23rd ACM/IEEE Design Automation Conf.*, pp. 824-830, June 1986.

\*



**Jan-Ming Ho** received the B.S. degree from Cheng-Kung University, Taiwan, in 1978, the M.S. degree from Chiao-Tong University, Taiwan, in 1980, both in electrical engineering, and the Ph.D. degree in computer science from Northwestern University, Evanston, IL, in 1989.

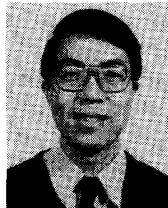
Currently, he is an Associate Research Fellow at the Institute for Information Science, Academia Sinica, Taiwan. His research interests include computational geometry, graph algorithms, and computer-aided design.



**Gopalakrishnan Vijayan** received the B.Sc. degree in mathematics from the University of Madras, India, in 1975, the B.E. degree in electronics and communication engineering in 1978 and the M.E. degree in automation in 1980 from the Indian Institute of Science, and the M.A. and Ph.D. degrees in electrical engineering and computer Science from Princeton University, in 1982 and 1983, respectively.

From 1983 to 1987, he was an assistant professor in the School of Information and Computer Science at the Georgia Institute of Technology. He joined IBM Research in June 1987 as a research staff member in the VLSI Design Algorithms group. His current research interests include VLSI design methodologies and algorithms, combinatorial optimization, and graph theory.

\*



**C. K. Wong** (M'71-SM'78-F'85) received the Ph.D. degree in mathematics from Columbia University in 1970.

He joined the IBM T. J. Watson Research Center in 1969 as a Research Staff Member and is currently manager of the VLSI Design Algorithms group in the Computer Sciences Department. He was Visiting Professor of Computer Science at University of Illinois, Urbana in 1972-1973 and at Columbia University in 1978-1979. His most recent research is on VLSI design algorithms. He holds four U.S. patents and has published over 100 papers in these areas. He is author of *Algorithmic Studies in Mass Storage Systems*. (Computer Science Press, 1983). He has received an Outstanding Invention Award, one Outstanding Technical Achievement Award and four other Invention Awards from IBM.

Dr. Wong is a member of the Association for Computing Machinery. He was Editor of *IEEE TRANSACTIONS ON COMPUTERS* from 1982 to 1985 and is an executive member of the IEEE Computer Society Technical Committee on VLSI. He is also on the editorial board of two international journals on Fuzzy Set theory and the founding Editor-in-Chief of *Algorithmica*.