# AN OPTIMAL STEINER TREE ALGORITHM FOR A NET WHOSE TERMINALS LIE ON THE PERIMETER OF A RECTANGLE

J. P. Cohoon
D. S. Richards
J. S. Salowe

Computer Science Report No. TR-88-11
June 7, 1989

# AN OPTIMAL STEINER TREE ALGORITHM FOR A NET WHOSE TERMINALS LIE ON THE PERIMETER OF A RECTANGLE†

*J.P. Cohoon, D.S. Richards, J.S. Salowe*

*Department of Computer Science*
*University of Virginia*
*Charlottesville, Virginia 22903*

**Abstract**

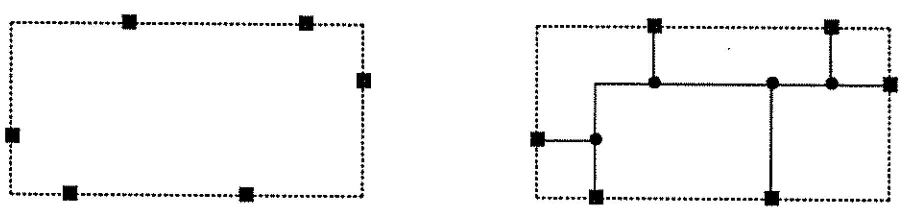
Given a set of input points, the rectilinear Steiner tree problem is to find a minimal length tree consisting of vertical and horizontal line segments that connects the input points, where it is possible to add new points to minimize the length of the tree. The restricted Steiner tree problem in which all the input points lie on the boundary of a rectangle frequently occurs in VLSI physical design. Since the fastest published algorithm is cubic in the size of the point set, VLSI designers have been forced to use heuristic approximations to the length of the Steiner tree for this problem. We present a simple, practical, linear-time exact algorithm to find the Steiner tree for points lying on the boundary of a rectangle, obviating the need for some heuristic algorithms in VLSI design. The analysis of the algorithm is based on the use of a novel tie-breaking rule that should prove useful for other Steiner tree problems.

† Submitted to *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. A conference version of this paper appears in the 1988 *IEEE International Conference on Computer-Aided Design*.

## I. Introduction

The Steiner tree problem seeks to construct a minimal length tree that interconnects a set of $n$ given points. These input points are called *terminals.* In constructing this minimal length tree, it is permissible to introduce other points, called *Steiner points,* if their inclusion allows a tree with less length. The Steiner tree problem has many different applications—communication networks, interstate highway systems, and VLSI physical design. However, it is the latter domain that is our concern here. In the VLSI physical design domain, the additional constraint of rectilinearity is imposed (i.e., all line segments are either horizontal or vertical).

Although the general rectilinear Steiner problem is NP-Hard [GARE77], a number of interesting special cases occur in the VLSI domain for which polynomial-time algorithms are known to exist. In this paper, we consider the problem where the set of $n$ terminals are constrained to lie on the perimeter of a rectangle. An example instance of the problem and the corresponding solution are given respectively in Figure 1.a and 1.b. In the figure, a terminal is represented by a "■" and a Steiner point by a "•."



(a) Problem instance (b) Steiner routing of instance

**Figure 1** — Problem instance and its solution.

The rectangular Steiner problem manifests itself in a variety of different ways in VLSI physical design. In its wiring phase, a common physical design activity is to find the minimal length routing of multi-terminal nets in a channel or switchbox [HU85, PREA88]. In its placement phase, a crucial component in min-cut placement algorithms is to evaluate the quality of a possible placement configuration by estimating the amount of wire that will be needed to subsequently route the chip [BREU77, DUNL85].

Nevertheless, the fastest published result is an $O(n^3)$ time algorithm due to Aho, Garey and Hwang [AHO77], so VLSI designers have been forced to settle for heuristic approximations to the length of the Steiner tree. Although this algorithm does have a polynomial running time, it is impractically slow because tens of thousands of multi-terminal nets must be considered. Our contribution to this problem is a fast and practical $O(n)$ time exact algorithm. In an independent investigation, Agarwal and Shing have presented a similar algorithm with a linear running time [AGAR86]. The proof and algorithm in their report contains a substantial error, but a correction has been claimed. In any event, their proof is very complicated and depends on nontrivial results of Hwang [HWAN76] and Aho, Garey and Hwang [AHO77]. The analysis of our result is much simpler than that of Agarwal and Shing and far easier to prove correct, since we only rely on first principles and the use of two tie-breaking rules.

Our algorithm achieves it running time by greatly restricting the appearances of optimal Steiner trees. This is done by the introduction of two tie-breaking rules, exterior wire and leftness, which are described in the next section. It appears that leftness is novel and should prove useful for other Steiner tree problems. In Section III, we describe our $O(n)$ time algorithm which exploits these restrictions to quickly find a rectilinear Steiner tree for problem sets where the terminals lie on the perimeter of a rectangle. In Section IV, we present some empirical results, and in Section V we make some comments regarding possible efficiencies.

## II. Topologies of Steiner Trees

Given a set of $n$ terminals lying on rectangular boundary $B$, let $\tau$ be the set of minimal Steiner trees. Each of these trees is composed of horizontal and vertical line segments, and it is easy to see that no segment lies outside $B$. Furthermore, let $len(s)$ be the sum of the lengths of the segments in $s$, and assume that the lower left corner of the rectangle is at the origin. We break ties in $\tau$ by preferring Steiner trees in $\tau_1 = \{ s \mid s \in \tau \text{ , } len(s \cap B) \text{ is maximized } \}$. This rule maximizes the use of the boundary (exterior wire). We break ties in $\tau_1$ by choosing a tree as far to the "left" as possible; i.e. if $len(s \cap x{=}t)$ is the length of the intersection of Steiner tree $s$ with vertical line $x = t$, we prefer trees in $\tau_2 = \{ s \mid s \in \tau_1 \text{ , } \sum t \cdot len(s \cap x{=}t) \text{ is minimized } \}$. The weighted sum $\sum t \cdot len(s \cap x{=}t)$ is called the *leftness* of tree $s$. We call a Steiner tree in $\tau_2$ an *optimal Steiner tree.*

In the remainder of this section, we prove the main theorem which bounds the number of topologies of optimal Steiner trees. To simplify the exposition, we define the following notation. A *corner-vertex* is a point in the interior of the rectangle incident to exactly one horizontal edge and exactly one vertical edge. The two edges incident to the corner-vertex form a *corner*; in the paper we will refer to these edges as the *legs* of the corner. If both legs of a corner intersect the boundary, the corner is called a *complete interior corner.* A T-*vertex* is an interior point incident to precisely three edges; these three edges together are called a T. The two collinear edges together are called the *head* of the T, and the remaining edge is called the *body.* A *cross-vertex* is an interior point incident to precisely four edges. The T- and cross-vertices are Steiner points, but the corner-vertex is not. An *interior line* is a maximal sequence of adjacent, collinear edges inside the rectangular boundary, and a *complete interior line* is an interior line whose endpoints lie on opposite sides of the rectangle $B$. An *exterior line* is a maximal sequence of adjacent, collinear edges on the boundary of $B$. A line $e$ incident to a second line $l$ is said to *point* towards the left if all of $e$ lies to the left of $l$. A similar definition holds for the other direction. The edges incident to an interior line are said to *alternate* if no edges on that line are incident to a cross-vertex and no two neighboring edges point in the same direction.

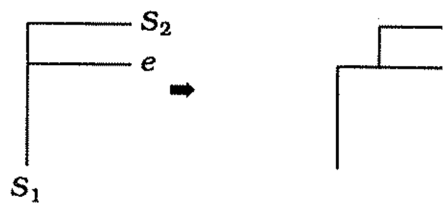
The following fact is used several times in the proof of the main theorem and is implied by the tie-breaking rule of leftness.

*Fact 1:* A corner vertex in the interior of $B$ cannot be incident to a horizontal edge on its left. That is, the corner must bend towards the right.

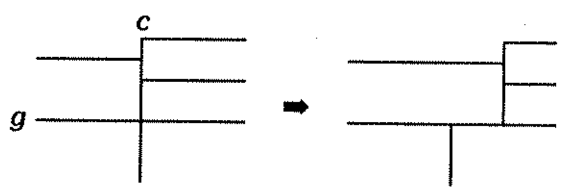
Please note that in the proofs that follow that the diagrams are used for illustrative purposes and do not necessarily represent the only possible configuration concerning the discussion.

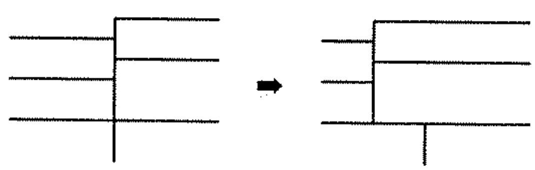
*Lemma 1:* In an optimal Steiner tree, edges incident to the legs of a corner must alternate. The edge closest to the corner-vertex must point in the opposite direction of the other leg of the corner.

*Proof:* Let $C$ be a corner with legs $S_1$ and $S_2$, and let $e$ be the interior line intersecting $S_1$ which is closest to $C$'s corner-vertex $c$. If $e$ and $S_2$ are not on opposite sides of $S_1$, either $e$ points in the same direction as $S_2$ or $e$ intersects $S_1$ at a cross-vertex. In either case, "flipping" $C$, as in the figure below, decreases the length of the Steiner tree, contrary to the claim of optimality.



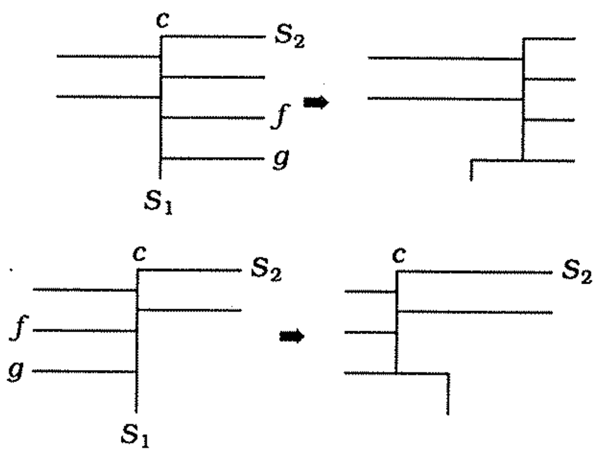

To prove that edges emanating from the legs of corner $C$ must alternate, suppose they do not. Then there is either a single line intersecting $C$ at a cross-vertex or T-vertex, or else two adjacent edges point in the same direction. Let $g$ be the edge closest to corner-vertex $c$ that does not alternate with the preceding one. Assume that $g$ and $S_1$ intersect at a T- or cross-vertex, and let $u$ be the adjacent sequence of edges joining $g$ and $c$. Then $u$ can be shifted appropriately to either decrease wire length or decrease leftness, depending on whether the number of edges on one side of $u$ equals the number of edges on the other side of $u$. Note that in the figures, all corners must bend towards the right because of Fact 1.



Now suppose that $g$ points in the same direction as the preceding edge $f$ and that these edges intersect side $S_1$. If $g$ and $f$ point in the same direction as $S_2$, then there is one more edge incident to $u$, including the endpoint at $c$ but not including the endpoint at $g$, pointing in the direction of $S_2$. Sliding $u$ in the appropriate direction will decrease total wire length. If $g$

and $f$ point away from $S_2$, then there is an equal number of edges pointing in either direction along $u$, including the endpoint at $c$ but not including the endpoint at $g$. Because Fact 1 states that all corners bend towards the right, $f$ and $g$ may either point towards the left, or they may be vertical. If $f$ and $g$ point towards the left, $u$ can be slid towards the left to decrease leftness without increasing the cost of the Steiner tree. If $f$ and $g$ are vertical, $u$ can be slid appropriately to decrease leftness.

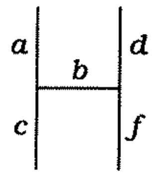



□

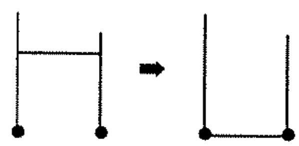
*Lemma 2:* In an optimal Steiner tree, the body of a T must hit the boundary. Furthermore, no interior edges are incident to the body of a T.

*Proof:* Consider a T with T-vertex $t$ and body $b$. If no interior edges are incident to $b$, then $b$ must hit the boundary. Now suppose that some edges are incident to $b$. There are three ways in which $b$ can end: it can hit the boundary, it can end at a corner vertex, or it can end at a T-vertex. Interior line $b$ cannot be a leg of a corner since Lemma 1 implies that the leg of a corner cannot intersect a T-vertex. If $b$ ends in a T-vertex and there is at least one edge incident to $b$ between the two heads, then there must be an equal number of edges intersecting $b$ in either direction; otherwise, one may slide $b$ appropriately to reduce wire length. However, with an equal number of edges on either side of $b$, one may slide $b$ appropriately to improve leftness, leading to the contradiction.

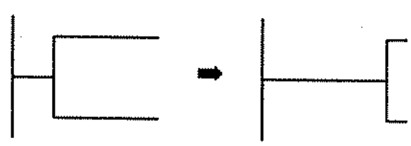
If $b$ ends in a T-vertex but no edge is incident to the interior of $b$, the proof is slightly more complicated. If $b$ is vertical, it can be slid to the left to decrease leftness. If $b$ is horizontal, then it must be the crossbar of an H-shaped figure.




The leftness property implies that either $a$ must extend above $d$, or both $a$ and $d$ hit the boundary. Specifically, if $a$ ends at a corner-vertex below $d$, then this corner must bend towards the right. If this is the case, $b$ can be shifted to decrease wire length. Similarly, if $d$ ends below $a$ it cannot end in a T. Symmetry implies that either $c$ must extend below $f$ with $f$ not ending at a T-vertex, or both $c$ and $f$ hit the boundary. If either $a$ and $d$ or $c$ and $f$ hit the boundary, $b$ can be slid to the boundary to increase the use of exterior wire.

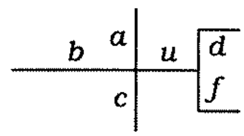


If this is not the case, both $d$ and $f$ end at corner-vertices. Lemma 1 implies that the edges incident to a leg of a corner must alternate, with the edges closest to the corner vertices pointing towards the left. As a result, there must be one more rightward edge than leftward edge incident to the union of $d$ and $f$. Shifting the union of $d$ and $f$ towards the right decreases wire length.



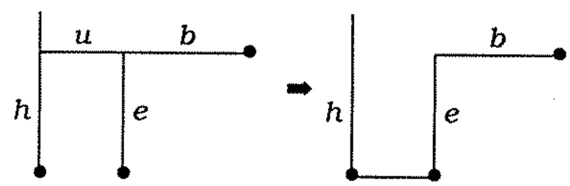
Therefore, $b$ must hit the boundary, regardless of the number of incident edges.

In addition, $b$ cannot intersect a cross-vertex $v$. Let $u$ be the portion of $b$ between $t$ and $v$. If any edges are incident to $u$, then $u$ can be slid appropriately to decrease leftness. Otherwise, a horizontal crossbar is present as in the case above. Using the same logic as in the previous paragraph, $d$ and $f$ must end at a corner vertices. By Lemma 1, the leg of a corner cannot intersect a cross-vertex, and by Fact 1, corners must bend towards the right. Therefore, the following topology must be present:

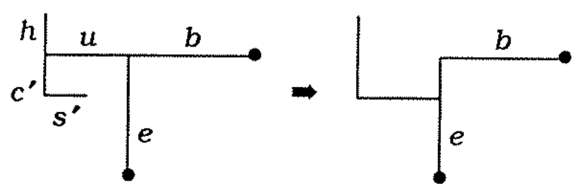


As before, the union of $d$ and $f$ can be slid towards the right to decrease total wire length.

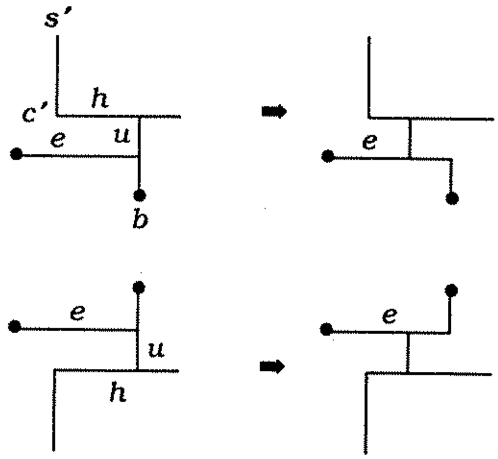
Now consider the edge $e$ closest to $t$ along $b$; the argument in the previous paragraphs proves that $e$ must hit the boundary. Let the portion of the head pointing in the same direction as $e$ be called $h$, and let the portion of $b$ between $h$ and $e$ be called $u$. If $h$ hits the boundary, then $u$ can be shifted to the boundary to increase exterior wire without increasing the cost of the Steiner tree, contradicting optimality.



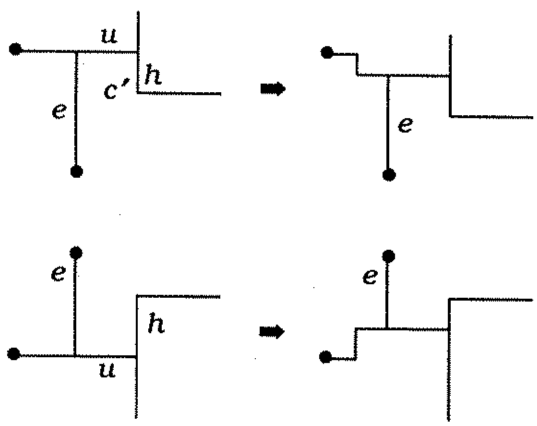
If $h$ does not hit the boundary, then either $h$ ends at a corner-vertex or another T-vertex. If $h$ ends at a corner-vertex $c'$ where the other leg $s'$ of the corner points towards $e$, then $u$ can be shifted towards $c'$ until total wire length is decreased.



The situation when $h$ ends in a T is identical. If $h$ ends in a corner-vertex $c'$ where the other side $s'$ of the corner points away from $e$, the situation is slightly more complicated. Fact 1 shows that corners are of only two types. One type originates from the top and then bends towards the right, and the other originates from the bottom and bends towards the right. Line $b$ can emanate from either side of the corner, and since $s'$ points away from $e$, $b$ can either point leftwards, upwards or downwards. If $b$ is vertically oriented, then $e$ must point towards the left. Edge $u$ can be moved to the left, decreasing leftness.




If $b$ points towards the left, then a portion of $b$ beginning just to the left of $e$ and ending at $h$ can be slid appropriately to decrease leftness. In any event, $h$ cannot end in a corner- or T-vertex, so no edge is incident to $b$.



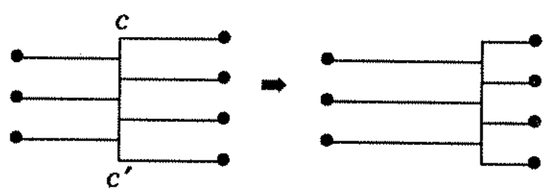

□

*Lemma 3:* In an optimal Steiner tree, the legs of a corner must hit the boundary, forming a complete interior corner. Each of the legs must also be anchored by interior lines; i.e. the interior lines incident to the corner must be connected to the other two sides of the boundary.

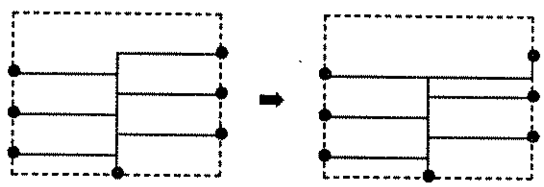
*Proof:* By Fact 1, all corners must bend towards the right. Therefore, the horizontal leg cannot end at a second corner-vertex. The horizontal leg cannot intersect a T-vertex since Lemma 2 implies that the body of a T must hit the boundary and have no edges incident to it.

As a result, the horizontal leg must hit the boundary.

We now consider the vertical leg $S$ of the corner $C$ with corner-vertex $c$. Both Lemmas 1 and 2 imply that $S$ cannot end in a T-vertex, and leftness implies that if $S$ ends in another corner-vertex $c'$, the bend at $c'$ must be towards the right. By Lemma 1, edges intersecting $S$ must alternate, and the edges closest to $c$ and $c'$ must point towards the left. Since the edges alternate and both corners bend towards the right, there is one more rightward edge than leftward edge incident to $S$. Shifting $S$ towards the right decreases wire length.




To show that each incident interior line must be anchored, suppose this is not the case. Then the corner can be flipped to increase exterior wire without increasing total wire length.



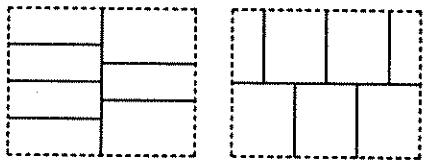
□

*Lemma 4:* In an optimal Steiner tree, the head of a T is either a leg of a complete interior corner or a complete interior line.

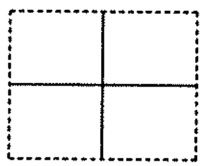
*Proof:* If one side of the head ends at a corner-vertex, then Lemma 3 implies that the head is one leg of a complete interior corner. Lemma 2 states that it is not possible for the head of a T to intersect another T, so the only other possibility is that both sides of the head reach the boundary, forming a complete interior line. □

*Theorem 1:* The interior lines of an optimal Steiner tree must have one of the following ten topologies:

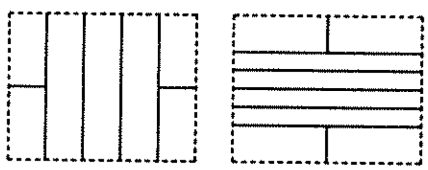
1. Boundary Tree -- No interior edges.



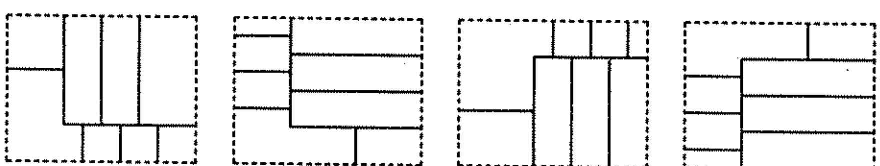
2. Complete Interior Lines -- Exactly one complete interior line.



3. Cross -- Two intersecting complete interior lines.



4. Earthworms -- Two or more parallel complete interior lines.

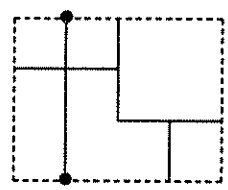


5. Complete Interior Corners -- An odd number of interior lines incident to one leg and exactly one interior line incident to the other leg.

*Proof:* If there are no interior edges in an optimal Steiner tree, the topology must be of type 1. Now suppose that there is some interior line in an optimal Steiner tree. This interior line $F$ must have one endpoint $f_1$ on the boundary. The other endpoint $f_2$ of $F$ can appear in one of three forms. Either $f_2$ lies on the opposite boundary, it is a corner-vertex or it is a T-vertex. The first and second cases imply that $F$ is either a complete interior line or one leg of an complete interior corner. By Lemma 4, the third case implies that the head of the T incident to

$f_2$ is either a complete interior line or one leg of a complete interior corner.

Suppose there is a complete interior corner. Lemma 3 implies that this complete interior corner must be anchored to all four sides of the bounding rectangle. If there was also a complete interior line, it would intersect the complete interior corner at a cross-vertex. This intersection may be interpreted as four T's, and at least one of the T's must have an edge incident to its body. This contradicts Lemma 2, so a complete interior corner and a complete interior line cannot occur simultaneously. Similar reasoning shows that it is not possible to have two or more complete interior corners.



We divide the topologies into cases. From the discussion above, we know that any topology with interior edges must contain either one complete interior corner or at least one complete interior line, but not both. Suppose that there is one interior corner, $C$. Because of the leftness property, $C$ must bend to the right. By Lemma 3, at least one interior line must be incident to each leg of $C$. These lines anchor the corner to the other two boundaries of the rectangle. Using Lemma 1, Lemma 2 and the fact that the Steiner tree contains no cycles, at most one leg of $C$ can have more than one incident interior line. Since any optimal Steiner tree must maximize exterior wire length, one can show that the number of interior lines incident to either leg of $C$ must be odd. (If the number were even, then one can slide the side with an even number of incident interior lines appropriately either to decrease wire length or to increase exterior wire length without increasing the cost of the Steiner tree.) This accounts for the topologies listed in case 5.

Now suppose the topology contains a complete interior line $L$, and assume without loss of generality that $L$ is vertical. Furthermore, assume that $L$ is the only vertical complete interior line in the Steiner tree. $L$ may have zero, one, or more than one incident horizontal edge. If $L$ has more than one incident horizontal edge, these edges must either alternate or form a cross.

If this is not the case, there must be two adjacent horizontal interior lines anchored to the same boundary. The portion of $L$ between the adjacent interior lines can be shifted to the boundary, increasing exterior wire. If the interior lines alternate, Lemma 2 implies that there are no other interior lines, so one of the topologies in case 2 must occur. If there is a cross, Lemma 2 implies that there are no interior lines incident to any of the four implicit T's, so the cross topology of case 3 must occur.

If there is more than one vertical complete interior line, Lemma 2 implies that these lines must be arranged to form an earthworm, with only the leftmost vertical line possibly having a leftward horizontal edge and only the rightmost vertical line possibly having a rightward horizontal edge. This accounts for topology in case 4.

Since every topology must either have no interior edges, a complete interior line or a complete interior corner, this list exhausts all possibilities. □

## III. The Linear-Time Algorithm

In this section, we describe a straightforward version of the Steiner tree algorithm. The five topologies stated in Theorem 1 are considered independently; we present a linear-time algorithm for each case which returns an optimal Steiner tree with the topology of interest if such a tree exists, and it will return a suboptimal tree otherwise. The optimal Steiner tree is the global minimum. In practice, many of these cases are very constrained and can be simplified or eliminated by some inexpensive tests. Some of these simplifications are presented in Section V.

The input to the Steiner tree algorithm is the set of terminals on $B$ in clockwise order. Note that if the terminals were unsorted, we would have a trivial $\Omega(n \log n)$ lower bound on any Steiner tree algorithm. It is most convenient to store these points in a circular, doubly-linked list so that neighboring points can be found in constant time. We also assume that there is at least one point on each side of $B$. If this is not the case, we can redefine the problem so that the points are on the boundary of the smallest enclosing rectangle $B'$. The optimal Steiner tree for points on the boundary of $B'$ cannot be longer than the optimal Steiner tree for the points on the boundary of $B$.

**Type 1 - No internal lines**

For this case and other cases below, we make use of a structure called *GAP*. Given two points $a$ and $b$ on the boundary of $B$, let $B(a, b)$ be that portion of $B$ that is traversed from $a$ to $b$ in the clockwise direction. $GAP(a, b)$ is defined as $B(a, b)$ with the longest interval between two points (a point is either $a$, $b$ or the terminals between $a$ and $b$) removed. If $a = b$, then the entire boundary $B$ is traversed, i.e. $B(a, a) = B$.

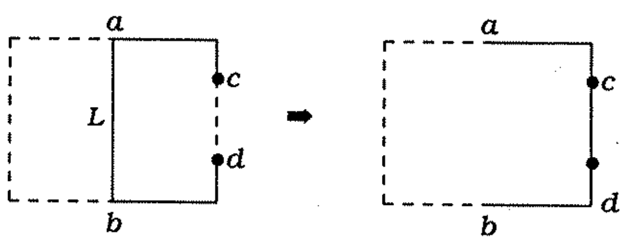
$GAP(a, b)$ can be determined in $O(n)$ time in the following manner. If both $a$ and $b$ are terminals, $GAP(a, b)$ is calculated by scanning the doubly-linked list from $a$ to $b$ and removing the largest interval. If at least one of $a$ and $b$ is not a terminal and there are at least two additional terminals on $B(a, b)$, $a$ is located between two adjacent terminals, $a_{cc}$ and $a_c$, and $b$ is located between two adjacent terminals, $b_{cc}$ and $b_c$. Terminal $a_{cc}$ is counterclockwise from $a$, $a_c$ is clockwise from $a$, $b_{cc}$ is counterclockwise from $b$ and $b_c$ is clockwise from $b$. The interval deleted is either the edge between $a$ to $a_c$, the edge between $b_{cc}$ to $b$, or the interval deleted from $GAP(a_c, b_{cc})$, whichever is the largest. If there is only one additional terminal $c$ on $B(a, b)$, then the interval deleted from $GAP(a, b)$ is either the edge from $a$ to $c$ or the edge from $c$ to $b$, whichever is largest. If there are no additional terminals on $B(a, b)$, then $GAP(a, b)$ is simply the points $a$ and $b$. Clearly, the optimal type 1 tree is just $GAP(a, a)$ for any terminal $a$.

**Type 2 - Exactly one complete interior line**

We describe an algorithm that computes an optimal Steiner tree if it includes a single vertical complete interior line. If no optimal Steiner tree for the input terminals contains a single complete vertical interior line, a suboptimal tree is returned. To find an optimal Steiner tree containing a single horizontal complete interior line, the point set is rotated by $\frac{\pi}{2}$ and the procedure for a vertical complete interior line is applied. There are 3 subcases.

**Type 2a - No incident interior lines**

In this case, one may slide the complete interior line appropriately so that it extends from $a$ to $b$, where at least one of $a$ and $b$ is a terminal. (We assume below that $a$ is on the top boundary of $B$ and $b$ is on the bottom boundary of $B$.) The rest of the optimal tree must be $GAP(a, b)$ and $GAP(b, a)$. It remains to decide on the optimal choice of $a$ and $b$; there are only $O(n)$ possibilities which are somewhat limited by topological considerations. If the optimal Steiner tree has one vertical complete interior line $L$ and no other incident interior lines, the interval deleted from $GAP(a, b)$ cannot be completely on the right boundary, and the interval deleted from $GAP(b, a)$ cannot be completely on the left boundary. To prove this, suppose that the interval deleted from $GAP(a, b)$ is completely on the right boundary. This implies that there are two points $c$ and $d$ on the right boundary which must be connected to $L$. Since there are no other interior lines, the exterior boundary from $a$ to $c$ must be connected, and the exterior boundary from $d$ to $b$ must be connected. Consequently, $L$ can be shifted to the right to either decrease total wire length or increase the use of exterior wire.




To simplify the computation, we will compute a restriction of $GAP(a, b)$, denoted $GAP'(a, b)$, which does not consider intervals on either the right or left boundaries.

A simple right-to-left linear-time sweep can compute every $GAP'(a, b)$, and a similar left-to-right sweep can compute every $GAP'(b, a)$. To compute $GAP'(a, b)$ where either $a$, $b$ or both $a$ and $b$ are terminals, start at the rightmost terminal on either the top or bottom boundaries, and examine the points along the top and bottom boundaries in order of nonincreasing $x$-coordinate. As each point is being considered, one of two records, *top−gap* and *bottom−gap*, is updated. Record *top−gap* contains three fields. The first field contains the length of the largest interval between two consecutive terminals on the portion of $B$ beginning at the last point visited on the upper boundary and ending at the topmost point on the right boundary;

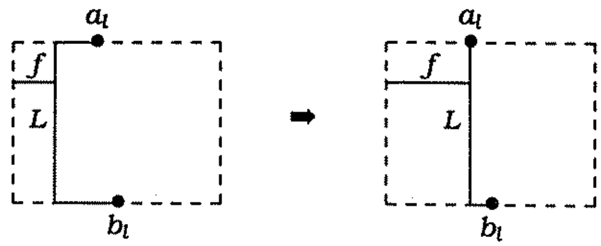
the second and third fields contain the terminals determining this interval. Record *bottom−gap* contains the analogous information for terminals on the bottom boundary. Upon examining a new point $d$, compute the point $d_o$ with the same $x$-coordinate as $d$ but on the opposite side of $B$. Without loss of generality, assume that $d$ is on the top boundary, so $GAP'(d, d_o)$ is being computed. The first field in *top−gap* is set to the maximum of the current contents of the first field of *top−gap* and the length of the interval from $d$ to its clockwise neighbor. If the first field of *top−gap* was changed, the second field is set to $d$ and the third field is set to the clockwise neighbor of $d$. The interval deleted from $GAP'(d, d_o)$ is the largest interval represented by *top−gap*, *bottom−gap*, and the interval from $d_o$ to the last terminal visited on the bottom side.

After $GAP'(a, b)$ and $GAP'(b, a)$ are computed for each terminal, the tree which minimizes the length of the union of $GAP'(a, b)$ and $GAP'(b, a)$ is chosen. If the optimal Steiner tree indeed has exactly one vertical complete interior line and no other interior lines, then $GAP'(a, b) = GAP(a, b)$ for the best computed choice of $a$ and $b$. Otherwise, $GAP'(a, b) \geq GAP(a, b)$. The procedure clearly runs in linear time.

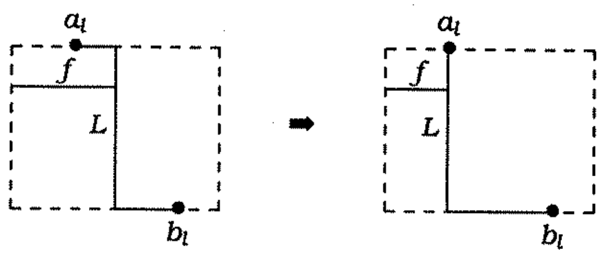
**Type 2b - One incident interior line**

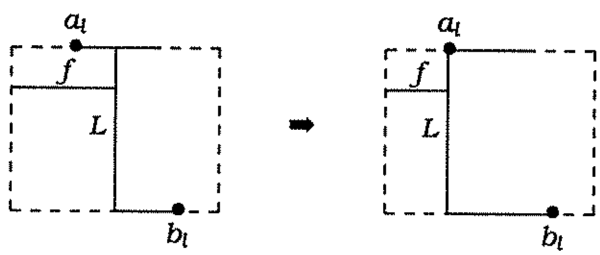
Consider a single interior line $f$ connecting the left side of boundary $B$ to the vertical complete interior line $L$. Assuming this is the optimal topology, the two left boundary corners are *forbidden* to appear in this optimal tree since either $f$ or a part $L$ may be slid to either decrease total wire length or increase exterior wire. Assume the endpoints of $L$ are $a$ and $b$; we use a sliding argument to show that at least one of $a$ and $b$ must be the leftmost terminal on its side of boundary $B$. Hence, the position of $L$ is determined.

Suppose that $a_l$ is the leftmost terminal on the top side of $B$, $b_l$ is the leftmost terminal on the bottom side of $B$, and that $a \neq a_l$ and $b \neq b_l$. Both $a_l$ and $b_l$ must be connected to the Steiner tree. If $a$ is to the left of both $a_l$ and $b_l$, then $L$ is connected to both $a_l$ and $b_l$ by exterior wire. Since the left corners are forbidden, there are no leftward exterior edges incident to $L$, so $L$ can be shifted towards the right to decrease total wire length as shown below.

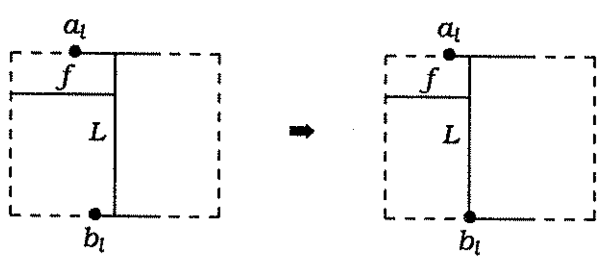



If $L$ is between $a_l$ and $b_l$, then there is one leftward exterior edge, one leftward interior edge $f$, and either one or two rightward exterior edges incident to $L$. In the first case, $L$ can be slid to the left to decrease wire length, and in the second case, $L$ can be slid to the left to increase exterior wire.




Finally, if $L$ is to the right of both $a_l$ and $b_l$, there are three edges incident to $L$ to its left, and at most two edges incident to $L$ to its right. $L$ can be slid to the left to decrease total wire length. Note that leftness was not used in the argument.



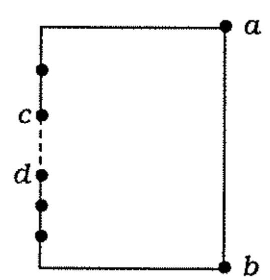

In addition to the placement of $L$, we can assume that the intersection point $c$ of $f$ and $B$ is the topmost terminal on the left side of $B$. Since the left boundary corners are forbidden and there is only one leftward interior line, all of the terminals on the left side of $B$ must be connected by a single vertical exterior line that runs from $c$ to the bottommost terminal on the left boundary. As a result, after $L$ is positioned, the entire left side of the Steiner tree is determined. The candidate optimal tree is completed in linear time by adding $GAP(a, b)$.

If there is a rightward incident line $f$ rather than a leftward incident line, we show that vertical complete interior line must be connected to the rightmost terminal on either the top or bottom of the rectangle. The argument in the preceding paragraphs for a leftward incident line did not invoke leftness and a symmetric argument can be applied here as well.

In order to compute the candidate optimal tree containing a rightward horizontal interior line, we simply reflect the set of terminals about the $y$-axis and run the algorithm above for a single leftward interior line. This algorithm computes two trees, one for each possible position of $L$. If the two candidate solutions have different lengths or the same length but different amounts of exterior wire, the better tree is reflected about the $y$-axis and returned. If the two candidate solutions have the same total length and use the same amount of exterior wire, the one with the rightmost interior line is reflected and returned.

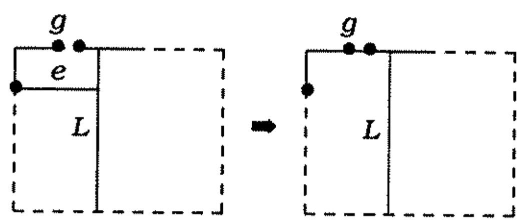
**Type 2c - Two or more incident interior lines**

By Theorem 1 the incident interior lines must alternate, with at least one horizontal interior line on each side of the vertical complete interior line. The algorithm begins by selecting a vertical complete interior line $L$ with endpoints $a$ and $b$. Assume that $a$ is on the top boundary of $B$ and $b$ is on the bottom boundary of $B$; we show below that there are only four possible positions for $L$. Given values for $a$ and $b$, the algorithm considers the left and right sides of $L$ independently. We describe the algorithm for the left side, and the algorithm for the right side is similar.



To compute the minimal length Steiner tree assuming $L$ is present and only exterior edges or horizontal interior edges are used, initialize the portion of the Steiner tree to the left of $L$ to be $GAP(b, a)$. Suppose that the interval omitted by $GAP(b, a)$ is the interval between $c$ and $d$ as shown. In general, $c$ and $d$ are not both on the left side of the boundary. Let the distance from $L$ to the left side of $B$ be $x$. For each terminal $v$ that lies on the left boundary between $c$ and upper left corner of $B$, inclusive, calculate the distance $y$ from $v$ to its clockwise neighbor. If $x < y$, remove the interval from $v$ to its clockwise neighbor from the solution and add a horizontal interior line from $L$ to $v$. A symmetric process handles the portion below $d$. This procedure uses $O(n)$ time since it makes only two passes over the points to the left of $L$.

To prove correctness, first note that all the terminals on the top side of $B$ must be connected by a single horizontal exterior line, and the same is true for the terminals on the bottom. To see this, suppose without loss of generality that there is a terminal $g$ on the top boundary to the left of $L$ not connected to its clockwise neighbor. Then $g$ is connected to $L$ by a horizontal interior line $e$ on the left of $L$ and the portion of $B$ between $e \cap B$ and $g$. Interior line $e$ may be slid upwards to decrease total wire length or increase exterior wire.




Let the tree computed by the algorithm above be $T$ and suppose that the best tree $T'$ with this topology, does not equal $T$. Furthermore, assume that all horizontal interior lines in $T$ and $T'$ are slid as far upwards as possible. This can be done without affecting the tie-breaking rules, and it ensures that any terminal strictly on the left boundary of $B$ incident to a horizontal interior line is not connected to its clockwise neighbor. Let $e$ be the lowest edge in $T$ but not in $T'$, and assume that there is horizontal interior line below $e$ in $T$. Edge $e$ must be either a horizontal interior line or an exterior edge with an endpoint on the left boundary of $B$.

If $e$ is a horizontal interior line incident to terminal $g$ on the left boundary, then $g$ must be connected to its clockwise neighbor $h$ in $T'$. However, the algorithm above added $e$ to $T$

because its length is strictly less than the distance from $g$ to $h$. We can therefore replace the edge in $T'$ from $g$ to $h$ by $e$ to yield a Steiner tree of smaller length, thereby contradicting the optimality of $T'$.

If $e$ is an exterior edge connected to terminals $g$ and $h$, both on the left side of $B$, then there must be a horizontal interior line $e'$ in $T'$ incident to $g$. By the construction of $T$, the length of $e'$ must be greater than or equal to the length of $e$. Replacing $e'$ by $e$ in $T'$ yields a tree with at least the same length and more exterior wire, again contradicting the optimality of $T'$. Since there is a horizontal interior line below $e$ in $T$, $e$ cannot be the lower left corner of $B$. If $e$ is the upper left corner, the argument used to replace an interior line in $T'$ by $e$ can be applied here as well.

If there are no horizontal interior lines below $e$ in $T$, then a similar argument, as in the above paragraphs, can be made to contradict the optimality of $T'$. In any case, $T$ used $e$ because it had length less than the alternative edge. Therefore, adding $e$ to $T'$ and removing a horizontal interior line yields a better Steiner tree, contradicting the optimality of $T'$.

It remains to show how to select $a$ and $b$. If $L$ is slid as far to the left as possible without increasing wire length, then there are only four possible locations for $L$: one of its endpoints is either the leftmost or rightmost terminal on the top or bottom of $B$. The argument is similar to the one in case 2b and is omitted. Hence we need examine at most 4 positions for $L$, so the total time for this case is $O(n)$.

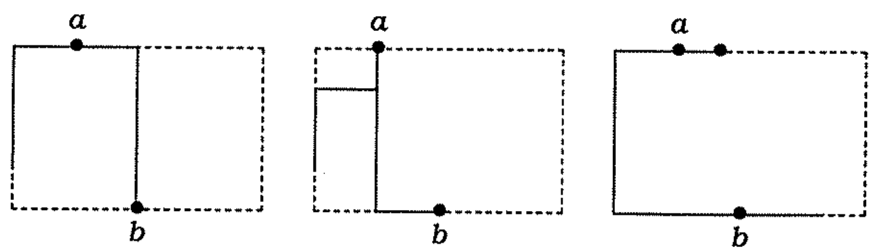
**Type 3 - Two intersecting complete interior lines**

Suppose the vertical complete interior line extends from $a$ to $b$ and the horizontal complete interior line extends from $c$ to $d$. Note that all four boundary corners are forbidden. We can assume that $a$ is the leftmost top terminal, otherwise we know $a$ is connected to a leftward tree edge and the segment from $a$ to the cross-vertex can be slid left. This creates a topology already handled in case 2c above. Similarly, we can assume $b$ is leftmost, and $c$ and $d$ are topmost. Since the remainder of the tree must use only exterior wire, each side of boundary $B$ is connected by an exterior line. The corresponding Steiner tree can be computed in linear time.

**Type 4 - Two or more parallel complete interior lines**

We assume the complete interior lines are vertical; the horizontal case is calculated by rotating the point set and applying the procedure for vertical complete interior lines. Consider the leftmost vertical line $L_l$, let the leftmost terminal on the top boundary be $a_l$, and let the leftmost terminal on the bottom boundary be $b_l$. If $L_l$ is slid as far to the left as possible, it can be in three possible positions: incident to $a_l$, incident to $b_l$, or along the left side of $B$. To prove this, consider the other locations for $L_l$, and recall that there is at most one leftward interior line $f$ incident to $L_l$. Suppose that $f$ exists. If there are two leftward exterior edges incident to $L_l$, $L_l$ can be shifted to the left to decrease total wire length. If there is only one leftward exterior edge incident to $L_l$, there may be either one or two rightward exterior edges incident to $L_l$. If there is one rightward exterior edge, then $L_l$ can be shifted to the left to increase exterior wire or decrease total wire length. If there are two rightward exterior edges, then $L_l$ can be shifted to the left to increase exterior wire. If no leftward exterior edge is incident to $L_l$, then there must be two rightward exterior edges incident to $L_l$, and $L_l$ must be to the left of both $a_l$ and $b_l$ because the left corners are forbidden. Since $L_l$ does not intersect any terminals, both $a$ and $b$ must be incident to an exterior edge. Therefore, $L_l$ can be shifted towards the right to decrease total wire length.

Now suppose that $f$ is not present. If there are two leftward exterior edges incident to $L_l$, then $L_l$ can be shifted towards the left to decrease leftness. If there is only one leftward exterior edge incident to $L_l$, then there is either one or two incident rightward exterior edges. If there is just one rightward exterior edge, then $L_l$ must be to the left of at least one of $a_l$ and $b_l$, say $b_l$. Otherwise, the Steiner tree would have to use the entire left boundary of $B$ to connect $a_l$ and $b_l$. Furthermore, if $L_l$ is to the left of $b_l$, $L_l$ must be to the right of $a_l$. If this is not the case, the total wire length can be reduced by replacing the leftward exterior edge connecting $L_l$ to the left boundary of $B$ by $f$. As a consequence, a leftward exterior edge must connect $a_l$ to $L_l$, and a rightward exterior edge must be incident to $b_l$. Therefore, $L_l$ can be shifted to the left to decrease leftness. If there are two rightward exterior edges, then $L_l$ can be shifted towards the right to decrease total wire length. Finally, it is not possible that there are no leftward exterior edges as there would be no way for the Steiner tree to connect to the terminals on the left boundary of $B$. In summary, the possibilities are shown below, where, without loss of generality, $a_l$ is not to the right of $b_l$.




Similarly, one can show that the rightmost vertical line, $L_r$, may be connected in one of three ways. Specifically, there is a Steiner tree in $\tau_1$ in which $L_r$ is connected to either $a_r$, the rightmost point on the top, $b_r$, the rightmost point on the bottom, or the right side of $B$. To prove this, suppose $L_r$ is located elsewhere, and assume without loss of generality that $a_r$ is to the right of $b_r$.

As in the case with $L_l$, there is at most one rightward incident interior edge $g$ incident to $L_r$. If $L_r$ is to the right of $a_r$, then $g$ exists. Otherwise, $L_r$ is connected to the right boundary of $B$ using one of the right corners. Since $a_r$ is to the left of the right boundary of $B$, replacing the exterior edge around the corner by $g$ yields a tree of smaller total wire length. Since $g$ is present, the corners are forbidden for the same reason as in type 2b. As there are no rightward exterior edges incident to $L_r$, there must be two leftward exterior edges incident to $L_r$, so $L_r$ can be shifted to the left to decrease total wire length.

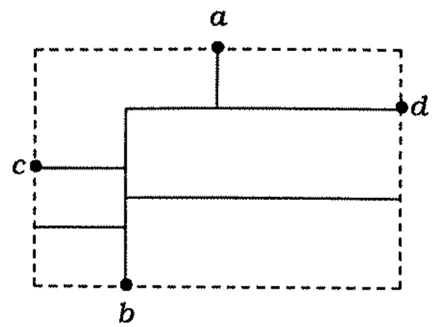
If $L_r$ is between $a_r$ and $b_r$, then $g$ may or may not exist. If $g$ does exist, the right corners are forbidden. As a consequence, there must be a rightward exterior edge connecting $L_r$ and $a_r$ and a leftward exterior edge connecting $L_r$ to $b_r$. $L_r$ can be shifted towards the right to either decrease total wire length or increase exterior wire. If $g$ does not exist, then there must be exterior edges connecting $L_r$ to $a_r$ and the terminals on the right boundary of $B$ which do not intersect the lower right corner. As before, there must be a leftward exterior edge connecting $L_r$ to $b_r$. If there is a second rightward exterior edge incident to $L_r$, then $L_r$ can be shifted to the left to decrease total wire length. If there is only one rightward exterior edge incident to $L_r$, then $L_r$ can be shifted to the left to decrease leftness.

Finally, if $L_r$ is to the left of $b_r$, then both $a_r$ and $b_r$ must be connected to $L_r$ by rightward exterior edges. If $g$ is present, $L_r$ can be shifted towards the right to decrease total wire length. If $g$ is not present, the configuration does not violate any of our tie-breaking rules. Nevertheless, we can shift $L_r$ until it reaches $b_r$ without increasing wire length or decreasing exterior wire, so there is a Steiner tree in $\tau_1$ of this type for which $L_r$ is incident to $b_r$.

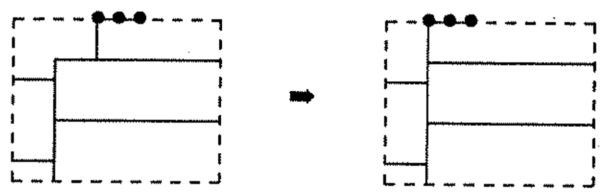
In total, there are nine possible locations of the leftmost and rightmost vertical lines. Given these lines, finding the remainder of the minimal Steiner tree can be reduced to the problem of determining the minimal Steiner tree for terminals along two parallel lines. This problem has been previously solved in $O(n)$ time using dynamic programming [AHO77], and their algorithm can be implemented to satisfy our tie-breaking rules. In order to obtain an optimal Steiner tree, determine the position of $L_r$ in the minimal length tree as stated above. If $L_r$ is incident to the leftmost of $a_r$ and $b_r$, then shift $L_r$ until it is adjacent with the nearest gap to its left.

**Type 5 - A complete interior corner**

We consider the case shown below, and the others are handled similarly. Note that we can assume the upper left corner is forbidden, otherwise either the segment at $c$ can be slid up to create a horizontal complete interior line or the segment at $a$ can be slid left to create a vertical complete interior line. Either situation could be considered an instance of case 2 above, and therefore, we assume they do not arise here. The upper right and the lower left corners are clearly forbidden. Therefore, $b$ is the leftmost terminal on the bottom side of the boundary, otherwise the vertical leg of the corner could be slid left. For a similar reason, $d$ is the topmost terminal on the right side of the boundary. We may assume by sliding that $a$ is leftmost and $c$ is topmost. Note that $b$ must be to the left of $a$.




In order to calculate the minimal Steiner tree in this case, slide $a$ so that it lies directly above $b$ and slide the other terminals on the top side of the boundary towards the left an equal distance.



We now have an instance of the topology of case 2c, and we can use that algorithm here. The minimal Steiner tree for this case can be found by sliding the terminals on the top side of the boundary back to their original locations, without increasing total wire length. The sliding of the top terminals can be performed in $O(n)$ time, so all four of the type 5 cases can be done in $O(n)$ time. The candidate Steiner tree is computed by shifting the top terminals back to their original positions.

## IV. Empirical Results

The above algorithm was implemented in C, and executed on a Sun3 workstation. The program is approximately 4500 lines of code and comments. Since the implementation was relatively straightforward, many tests for eliminating degeneracies were not included. Some possible efficiency considerations are suggested in the next section. Many small examples were solved in a fraction of a second. For example, the six-terminal net displayed in Figure 1 was routed in less than a tenth of a second. Figure 2 graphs the run-time of our program with respect to a test suite of twenty instances. An examination of the graph shows that the program's running time $t(n)$ in practice is a very slow-growing linear function—a rough approximation is $t(n) = n/40$ seconds.

## V. Code Improvements

While we have coded the algorithm as described above, we are aware of many improvements that could have been incorporated. Of course the performance remains linear but the constant of proportionality can be decreased. There are three classes of improvements. First, there are those based on the expected distribution of the terminals. Second, there may be additional constraints that optimal Steiner trees must satisfy, for each topology. So far we have only used as many as we needed to get the linear time bound. Third, there is code tuning, which we do not discuss further.

A surprising empirical observation is that for random instances of even moderate size, it is nearly always the case that the optimal topology is type 1. Conveniently, type 1 is the easiest to compute. As the number of terminals increases this becomes a near certainty. There are two implications. First, since complex topologies are rarely used, early efforts to exclude those cases will probably be rewarded. Second, a branch-and-bound approach could

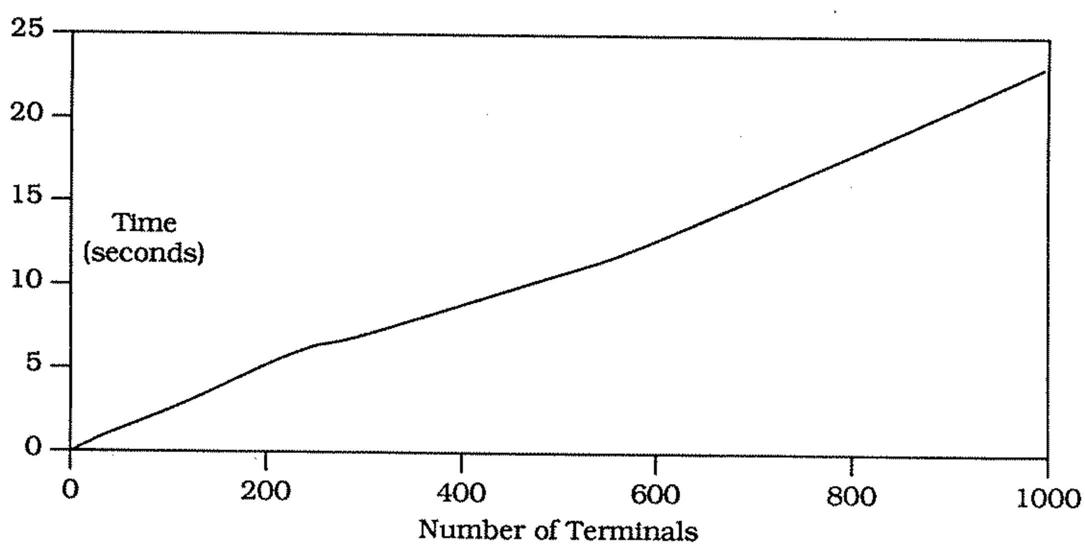



**Figure 2** — A graph of $t(n)$.

be used after beginning with the type 1 case. The latter approach involves much new coding.

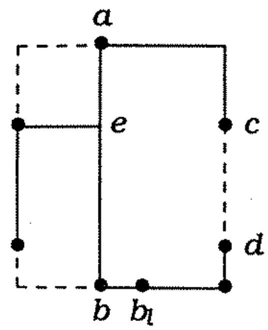
In some cases it may seem that using additional tests for detecting violated constraints, or other degeneracies, may cost more than the accrued savings. Of course any small improvement accumulates if the algorithm is used repeatedly. Further, our code creates a linked list representation of the tree for each case and subcase; it reports the tree with the best score. Topologies that violate constraints can simply be ignored, without calculating and comparing their costs. (There may be more economical encodings of each of the trees, so that we need only fully construct the best tree.)

We now present possible improvements for each type of topology. These suggestions are not exhaustive. They are merely sketched without proof; the interested reader can readily supply the details and proofs.

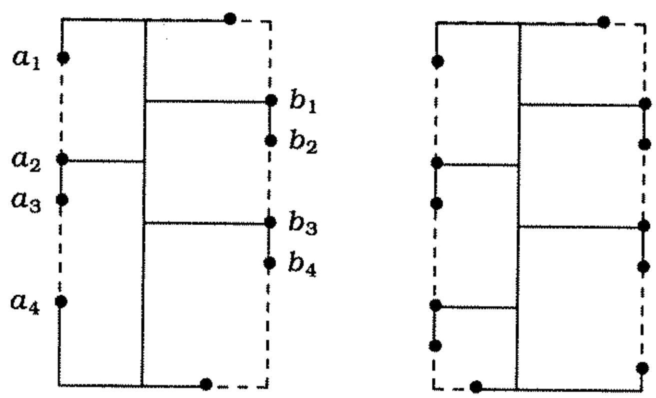
**Type 1:** This is already quite fast. It is possible to anticipate other *GAP* calculations that will be needed, e.g., corner gaps. They could be precomputed during this pass.

**Type 2a:** When possible, we will use the notation of Section III. Suppose the longest gap, as found in the computation of the type 1 tree, is entirely on the left or right boundary. Then that gap will be chosen leading to a contradiction of the constraint described in Section III. This simple test can immediatedly eliminate this case.

**Type 2b:** Let $dist(u, v)$ be the rectilinear distance from $u$ to $v$. Consider the leftward case shown. Recall the two corners are forbidden; hence the diagram is general, though $c$ and $d$ need not be on the right side. If $dist(c, d) \leq \max\{dist(a, e), dist(e, b_l)\}$ then the case can be excluded. Note that $e$ can be slid down to effectively increase $dist(a, e)$. Similar inequalities can be given for the corner gaps.




**Type 2c:** As implemented in Section III additional tests can be added, e.g., $dist(c, d) > x$ must be true. However we could use a different approach. There are so many constraints on this case that the details of the topology are nearly fixed, if one is willing to split this into subcases. Consider the subcase where the uppermost interior line is rightward and there is a left terminal above this line. There are two situations depending on the parity of the number of incident interior lines. Consider the odd case as shown.




The two right corners are forbidden. The lowest left terminal above $b_1$ is $a_1$ and must be reached around the corner. Further, $b_2$ must be above $a_2$, $a_3$ must be above $b_3$, etc. The even case is virtually identical. It is remarkable that the horizontal interior lines are forced to be introduced alternately is such a regular pattern, without regard to the length of the incident horizontal lines. This can be easily implemented as a single "merge" of the right and left terminal sequences. Of course, to complete the analysis of type 2c trees, there are other similar subcases to consider. For example, if there is no $a_1$ above $b_1$ then it is suggested that the case of exactly two incident horizontal lines be handled separately, but three or more lines be treated the same as above.

**Type 3:** The constraints ensure that $a$ and $b$ have the same $x$-coordinate, by leftness. Further, $c$ and $d$ have the same $y$-coordinate, otherwise suppose $c$ is above $d$. The leftward interior side of the cross can be slid up, indicating that "this" tree would be found by case 2c above. In addition to these constraints we can readily produce inequalities for each of the four corner gaps.

**Type 4:** As described above this case is complicated by the fact that we begin in $\tau_1$ and do a (needless) postprocessing step to produce an optimal tree in $\tau_2$. We could instead modify the dynamic programming algorithm [AHO77] to have it detect that it has reached $b_r$ or $a_r$, whichever is leftmost. At that point it can compute several completions. Assume $a_r$ is not to the left of $b_r$ and $b_0$ is on the bottom, opposite of $a_r$. In each case the terminals properly on the right side are ignored. One completion is for the original points. A second completion is with $b_0$ introduced as a pseudo-terminal. The third completion is with the two right corner points introduced as pseudo-terminals. (Of course these pseudo-terminals may have already existed, eliminating the need for some cases.) In each case the right terminals are easily handled in a postprocessing step.

This modified procedure is executed for each of the three starting positions of the leftmost vertical line. Whenever fewer than two vertical interior lines are used, the output is ignored.

**Type 5:** The constraints on the positions of $a$ and $c$ have been discussed. Further the case 2c algorithm must return a topology that has a horizontal interior line connected to $d$. This could be checked initially with an inequality involving the upper right corner gap and the $x$-coordinates of $a$ and $b$.

## VI. Conclusions

We have presented a linear-time algorithm to find a minimal Steiner tree for a set of terminals on the boundary of a rectangle. It is superior to previous work in its clarity, speed, and simplicity. The analysis was based on first principles and the use of two tie-breaking rules, one of which appears novel.

## VII. Acknowledgements

The work of J. P. Cohoon has been supported by in part the National Science Foundation through grant DMC 8505354 and by the Virginia Center of Innovative Technology through grant PDC-86-002-02. Their support is greatly appreciated.

## VIII. References

[AGAR86] P. K. Agarwal and M. T. Shing, "Algorithms for the Special Cases of Rectilinear Steiner Trees: I. Points on the Boundary of a Rectilinear Rectangle", 86-17, University of California at Santa Barbara, 1986.

[AHO77] A. V. Aho, M. R. Garey and F. K. Hwang, Rectilinear Steiner Trees: Efficient Special-Case Algorithms, *Networks*, *7*, 1977, 37-58.

[BREU77] M. A. Breuer, Min-Cut Placement, *Design Automation & Fault-Tolerant Computing*, *1*(4), October 1977, 343-362.

[DUNL85] A. E. Dunlop and B. W. Kernighan, A Procedure for Placement of Standard-Cell VLSI Circuits, *IEEE Transactions on Computer-Aided Design*, *CAD-4*(1), January 1985, 92-98.

[GARE77] M. R. Garey and D. S. Johnson, The Rectilinear Steiner Tree Problem is NP-Complete, *SIAM Journal of Applied Mathematics*, *32*, 1977.

[HU85] T. C. Hu and M. T. Shing, "A Decomposition Algorithm for Circuit Routing", in *VLSI Circuit Layout: Theory and Design*, T. C. Hu and E. S. Kuh (editors), IEEE Press, NY, NY, 1985, 144-152.

[HWAN76] F. K. Hwang, On Steiner Minimal Trees with Rectilinear Distance, *SIAM Journal of Applied Mathematics*, *30*, 1976, 104-114.

[PREA88] B. T. Preas and M. J. Lorenzetti, editors, *Physical Design Automation of VLSI Systems*, Benjamin/Cummings, Menlo Park, CA, 1988.