An Improved Rectilinear Steiner Tree Algorithm for Terminals Located on a Convex Polygon

Shaodi Gao and K. Thulasiraman
Department of Electrical and Computer Engineering
Concordia University
1455 de Maisonneuve Blvd. West
Montreal, Canada H3G 1M8
shaodig@ece.condordia.ca

Abstract

A minimal rectilinear Steiner tree for a set of points in the plane is a minimum-length tree that connects these points by using horizontal and vertical line segments. This paper deals with the case in which all input points are located on the boundary of a convex rectilinear polygon. We present an algorithm that constructs a minimal rectilinear Steiner tree in $O(k^2n)$ time, where n is the number of points and k is the number of sides of the polygon. The best known algorithm, so far, runs in $O(k^4n)$ time.

1 Introduction

Given a set N of n points in the plane, a rectilinear Steiner tree (RST) for N is a tree that connects all points in N by using only vertical and horizontal line segments. These segments can intersect at points that do not belong to N. Such points are called *Steiner points*, while the original points in N are called *terminals*. The length of an RST is defined to be the total length of the segments in the tree (tree edges). A minimal rectilinear Steiner tree (MRST) for N is an RST of minimal length.

The rectilinear Steiner tree problem is to find an MRST for a given set of points. Due to its wide applications in practice, this problem has been studied intensively in recent years. Many of these applications occur in VLSI physical design where technological considerations restrict electrical interconnections of the terminals to be realized by wires running only in vertical and horizontal directions. In general, the rectilinear Steiner tree problem has been shown to be NP-complete [7]. However, optimal solutions for several several special cases can be found in polynomial time. For the case where all terminals are located on the boundary of a rectangle, Aho et al. [2] proposed the first algorithm running in (n^3) time. Agarwal and Shing [1] later presented a linear-time algorithm, which was then simplified by Cohoon et al [5]. A more general case restricts the terminals to be located on the boundary of a convex rectilinear polygon. The first algorithms for this case, which have $O(n^6)$ time complexity, were given independently by Bern [3] and Provan [11]. Their algorithms are a modification of the algorithm proposed by Dreyfus and Wagner [6] for finding Steiner trees in graphs. Bern [4] later reduced the running time to $O(n^5)$. Recently, Richards and Salowe [12] proposed a linear-time algorithm by adopting the ideas developed for the rectangular boundary case. But the time complexity of their algorithm also depends on the number k of the boundary sides. Actually, the algorithm runs in $O(k^4n)$. In this paper we present an $O(k^2n)$ algorithm. Our algorithm is an important improvement on the algorithm by Richards and Salowe if k is not a constant.

This paper is organized as follows. In Section 2, we introduce the notation that will be used in the paper. We also present the theoretical results by Richards and Salowe in [12]. Section 3 describes more topological properties of MRSTs that are necessary to develop our algorithm. In Section 4, solutions for some special subproblems are provided. In Section 5, we present the $O(k^2n)$ algorithm. We conclude with suggestions for further improvement.

2 Notation and Preliminary Results

Consider the boundary where all input terminals are located. The boundary B of a rectilinear polygon consists of a sequence of lines (line segments), called *boundary lines*, which point alternately in vertical and horizontal directions. Two adjacent lines in the sequence form a *boundary corner.* With respect to the interior of the polygon, a convex corner is called an *outer corner*, and a concave corner is called an *inner corner.* Terminals and boundary corners are called *nodes*, which divide boundary lines into *boundary edges.* There are four special lines of a convex rectilinear polygon, each of which connect two outer corners. The other boundary lines connect an inner corner with an outer corner. These four special boundary lines are in the positions as far to the north, south, east and west as possible on boundary B and they are called the north tab, the south tab, the east tab and the west tab, respectively. Throughout this paper we assume that (1) the interior of the polygon is connected, and (2) every outer corner has a terminal on it. These are not rigid restrictions, since any problem instance can be modified to meet them.

A rectilinear Steiner tree for terminals lying on boundary B also uses line segments inside B.

An *interior edge* is a line segment inside B that does not contain any terminals or Steiner points, except for its endpoints. An *interior line* consists of one or more adjacent and collinear interior edges. A *complete interior line* is an interior line with both endpoints on the boundary. Inside boundary B, a point of degree two is called a *corner-vertex*, a point of degree three is called a *T-vertex*, and a point of degree four is called a cross-vertex. In an RST T-vertices and cross-vertices are Steiner points. The two interior lines that meet at a corner-vertex form an *interior corner*, and the lines are referred to as the *legs* of the corner. If both legs intersect the boundary, the corner is called a *complete interior corner*. In the following, we may omit the word "interior" if the context is obvious. A line e incident to a second line l is said to *point* to the left if e lies entirely to the left of l. Similar definitions hold for the other directions. A set of two or more edges incident to a line are said to *alternate* along the line if no edges are incident to a cross-vertex and no two neighboring edges point in the same direction.

For a given set of terminals there are many rectilinear Steiner trees of the same minimum length. A rectilinear Steiner tree can be transformed to another one without increasing the total length. An edge can be *slid* along two parallel lines to which the edge is incident so as to form an H-shape. To *flip* an interior corner formed by two edges e_1 and e_2 means to replace these edges by the other two edges of the rectangle which is determined by e_1 and e_2. In order to break ties among the MRSTs, the following rules will be used in the order they appear:

1. Choose the MRSTs with the maximum total length of boundary edges.
2. Choose the MRSTs with the maximum total node degree.
3. Choose the MRSTs whose interior corners always have the vertical leg to the left of the horizontal leg and whose interior edges cannot be slid to the left or to the top anymore.

By applying these tie-breaking rules the search space for an MRST can be reduced significantly. The following theorems present the theoretical ground for these reductions. Let a *topological component* of an RST be a connected portion of the tree which contains only interior edges. Theorem 1 restricts the variety of topological components of MRSTs, while Theorem 2 restricts the geometrical positions of the components. These theorems and the other results presented in this section have been developed by Richards and Salowe in [12].

Theorem 1 *For a given set of terminals on the boundary of a convex polygon, there is always an MRST that contains only the following types of topological components:*

1. *Interior lines with no incident edges.*

2. *Interior lines with incident edges. If there are exactly two edges incident to a line, the edges may be incident to the same Steiner vertex. Otherwise the incident edges must alternate along the line.*

3. *Complete interior corners with no incident edges.*

4. *Complete interior corners with zero or one edge incident to one leg and alternating edges incident to the other leg. The edge closest to the corner-vertex must point opposite to the other leg of the corner.*

The topological components of Type 1 is easy to handle, because each of them only consists of a single line. For all the other types, the placement of a topological component in an MRST is determined by its *backbones.* The backbone of a complete line with incident edges is the line itself; the backbone of a corner consists of its two legs and the incident edges closest to the corner-vertex. These backbones will be placed only on the so-called blue lines. Blue lines consist of two groups. The first group includes the horizontal and vertical lines emanating from inner corners, the second group contains the horizontal and vertical lines emanating from the terminals next to the endpoints of the lines in the first group. Obviously, there are altogether $O(k)$ blue lines.

Theorem 2 *There is an MRST in which all topological backbones of Types 2-4 are located on blue lines.*

Theorem 2 reveals that the number of different topological components for an MRST is only dependent on the number k of boundary sides. This theorem indicates the possibility of a running time that is linear in the number of terminals. But for a corner with edges incident to both legs, the backbone consists of four edges. If each of these edges is placed on blue lines independently, there are $O(k^4)$ possible placements for corners of this type. In the next section a detailed analysis on the dependency of different parts of a backbone will show that many of these placements are not necessary.

Figure 1 illustrates the topological components of Types 2-4 according to the parity of the number of the incident edges. An *odd (even)* line has an odd (even) number of incident edges. A *zero-even* corner has one leg with no incident edge and the other leg with an even number of incident edges. Similar definitions apply to *zero-odd*, *one-even* and *one-odd* corners.

The appearances of these corners and lines are closely related to the configuration of the four tabs on the boundary. Two tabs are said to be *separate* if they do not share a boundary corner. Otherwise, we call this pair of tabs as a *paired* tab. If three tabs are three consecutive boundary lines, we call them as a *triple* tab. A topological component which has m nodes $v_1, \cdots, v_m$ on boundary B divides it in m portions $B(v_1, v_2), \cdots, B(v_{m-1}, v_m)$. Each of these portions $B(v_i, v_{i+1})$ can join the simple path $F(v_i, v_{i+1})$ in the component between v_i and v_{i+1} to form a simple cycle. $F(v_i, v_{i+1})$ is called a *facet* of the component, and $B(v_i, v_{i+1})$ is said to be *enclosed* by the facet.

Lemma 1 *Let F be a facet with two perpendicular edges e_1 and e_2 intersecting the boundary. If e_1 and e_2 are the only edges of F, or if they are incident to a corner to form F, then the boundary portion enclosed by F must contain at least one tab properly (i.e. the tab does not intersect e_1 or e_2).*

In this case F is said to *enclose* a tab properly. In Figure 1 the facets marked with T's have to contain at least one tab properly. This follows directly from Lemma 1. We summarize it in the following Lemma.

Lemma 2 *If there is a paired tab on the boundary, no one-odd corner can occur. If there is two paired tabs, no one-even or zero-odd corners can occur. If there is a triple tab, only lines without incident edges can occur.*

3 More Properties of Topological Components

In this section we present new theoretical results on different types of topological components which are necessary to develop our algorithm. In the following we distinguish two legs of a corner by calling the one with more incident edges as the *major leg* and the other the *minor leg*. The portion of boundary between the west and north tab is called *NW-side* of the boundary. The other three sides can be defined accordingly. If two sides are only separated by a tab, they are said to be *adjacent*; otherwise they are *opposite*. In order to simplify the discussion, we assume in the following proofs that the major leg of corners always takes the horizontal direction. The case of vertical direction can be treated in an analogous way.

Another tie-breaking rule will be applied to force edges incident to complete lines and corners to slide in certain directions. For a boundary point a and an interior line segment l, the direction along l in which the boundary comes close to l is called the *favorable direction for a with respect to l*. For an edge e incident to a line segment l, the favorable direction for e is the favorable direction for its endpoint on the boundary. Let s be the boundary edge that shares the node with e and points towards the favorable direction of e. Note that e and s are perpendicular edges. We say that e and s are *dual* to each other with respect to l. If a boundary edge s is longer than its dual edge with respect to l, then s is said to be *replacable* with respect to l. For an MRST that satisfies the conditions of Theorem 1 and Theorem 2, we slide all edges incident to complete lines and corners in their favorable directions as far as possible without increasing the tree length.

Lemma 3 *Let l be either a complete line or the major leg of a corner in a MRST. After being slid in their favorable directions as far as possible, the edges incident to l have the following properties:*

1. *Each incident edge is shorter than its dual boundary edge, which must not be present in the MRST.*

2. *Let $s_1, \cdots, s_v$ be the boundary edges above l that are replacable with respect to l and $e_1, \cdots, e_v$ be their dual edges, respectively. All the dual edges except one are present in the MRST. This absent edge is either e_1 or e_v, or it is beside the north tab. A similar claim is valid for the replacable edges below l.*

Proof:

(1) The boundary edge s is not present in the Steiner tree, otherwise e could be slid farther in the favorable direction. If s were shorter than e, we could replace e by s, which would maintain the connection and reduce the total length of the MRST. Therefore s must be longer than e.

(2) The replacable edges $s_1, \cdots, s_v$ divide the boundary into $v+1$ portions $p_0, p_1, \cdots, p_v$. At least the portions $p_1, \cdots, p_{v-1}$ have to be connected to l by edges incident to l (See Figure 2). If we consider the construction of this connection as a subproblem, then the boundary of this subproblem contains a triple tab. By Lemma 2 the connection only consists of interior lines of Type 1 and boundary edges.

In case that the north tab is not within the horizontal interval of l, $p_2, \cdots, p_{v-1}$ are on the same boundary side. In addition to the incident edges, these portions will be connected only by boundary edges. If for $2 \leq i \leq v$, edge e_i, which is the dual edge of the replacable boundary edge s_i, is not present in the tree, then either s_{i-1} or s_i must be used to connect p_{i-1} with the other parts of the tree. Since both s_{i-1} and s_i are longer than e_i, this tree cannot be an MRST.

In case that the north tab is within the horizontal interval of l, a similar argument holds except for p_i which contains the north tab. It may be noted that e_1 must be a tree edge in this case. The connection for p_i may contain some lines of Type 1. There are two possibilities to connect p_i with l: (1) all terminals in p_i are connected through the shorter edge of e_i and e_{i+1} or (2) the left part of p_i is connected through e_i and the right part through e_{i+1}. In the first case, the longer edge of e_i and e_{i+1} is not a tree edge. In the second case, both of them are present in the tree. In the next section we will describe how to find the connection for p_i.

The replacable boundary edges below l and their dual edges can be treated in the same way. ■

This proof has actually shown how to construct a complete line and a corner with incident edges, if the position of the line itself or the major leg of the corner is given. We call a line or a corner *critical* if the incident edges are slid in the favorable directions as far as possible and they have the properties of Lemma 3.

Corollary 1 *If the position of a line is determined, there are at most four possibilities to choose the incident edges for a critical line or corner.*

Lemma 4 *The minor leg of a zero-odd corner can always be transformed to hit an inner boundary corner without changing the total length. For a zero-odd corner with the minor leg ending at a given node, we need to consider at most two positions where the major leg can be located.*

Proof: The first claim of the lemma has been shown in the proof of the Theorem 2. Let l be the major leg which takes the horizontal direction and ends at node b. (See Figure 3)

We consider first that b is located on the SE-side. Assume l has been moved upward as far as possible. Let s be the boundary edge between b and the neighboring terminal b' from above. This edge is not present in a MRST, otherwise l can be moved upward further without increasing the total length. On the other hand, s must be longer than the rightmost incident edge f, because f can be replaced by s. By convexity s must also be longer than the leftmost incident edge e, hence e is on the NW-side of the boundary. The endpoint u of e is the leftmost terminal on the NW-side to the right of the minor leg; otherwise e could be slid further to the left, or the boundary edge incident to u from left could replace the portion of the major leg to the left of e. Therefore b is the highest terminal on the NE-side below u.

If b is on the NE-side of the boundary, then, for similar reasons, the boundary edge s between b and the neighboring terminal b' from below is not present in any MRST and s must be longer than the minor leg. Therefore b is the lowest terminal on the NE-side above a. ∎

Lemma 5 *For a given node a, we have to consider $O(1)$ one-odd corners whose minor leg ends at that node.*

Proof: Let l_v and l_h be the minor and major legs, respectively. Suppose l_v ends at a, l_h ends at b, and the edge e incident to l_v ends at c (See Figure 4). Lemma 2 implies that the boundary portion from a to c contains exactly one tab.

If a is on the SW-side, e must be on the NW-side of the boundary. The favorable direction for e is upward. Suppose e cannot be moved upward anymore without increasing the total length. Then the boundary edge s incident to c from above is not a Steiner tree edge and must be longer than e. Therefore s is a replacable edge with respect to l_v. If there is another replacable edge, we can show the corner does not belong to an MRST. Hence there can be at most one position of e, no matter where the major leg is located. On the other hand if the major leg is determined, the other end of s must lie above the corner-vertex; otherwise the upper portion of the vertical leg can be replaced by s to increase the total boundary length. Hence the endpoint c of e is the first terminal on the NW-side below the corner vertex. Similarly, we can show that the only position of e on the SE-side is the highest terminal below a, if a is on the SE-side of the boundary.

The location of e is determined independently from the position of the major leg. Together with the result of Lemma 4, we have shown there are a constant number of one-odd corners with the minor leg at the same node. ∎

We summarize the results in this section as follows. For each node on the boundary, there are $O(1)$ critical zero-odd and one-odd corners with the minor leg ending at that node. In addition, $O(1)$ critical complete lines with incident edges can intersect the same node. It may be recalled that there are $O(k)$ nodes located on blue lines. Therefore, the total number of these critical components is $O(k)$. However, there are more critical zero-even or one-even corners. For each position of the minor leg, there may be $O(k)$ positions for the major leg. (The major leg always ends at an inner boundary corner). Therefore, $O(k^2)$ zero-even and one-even corners are critical.

Theorem 3 *In order to construct an MRST, we have to consider $O(k^2)$ zero-even and one-even corners, and $O(k)$ other corners and lines with incident edges.*

4 Solutions for some special subproblems

In this section we describe procedures which solve some special subproblems. In particular, we will show how to construct subtrees which contain one *basic* topological component. Basic topological components are topological components except lines in Type 1 which neither lie on a blue line nor intersect two adjacent boundary sides. In the next section, these subtrees will be used to compose MRSTs.

A basic component divides the boundary into several parts. Each portion, together with the corresponding facets of the component, constitutes the boundary of a subproblem. The tabs of the original boundary remain those of the new boundary. The other tabs of the new boundary are edges of the facet or boundary edges hit by the facet and they are not separate. A facet that encloses less than two tabs of the original boundary, defines a subproblem with a triple tab. A facet that encloses two tabs may define a subproblem with a paired tab. In this section we solve all subproblems with a triple tab along with the construction of the component. We call the subtree produced in this way an extended component.

Boundary with a triple tab

First we describe how to construct an MRST for a boundary with a triple tab. By Lemma 2, we only need to consider the topological components of Type 1, i.e., lines without incident edges. Without loss of generality, we assume that the single tab is the north tab. The algorithm applies the sweep technique in the vertical and horizontal directions.

In the vertical direction we begin with a horizontal sweep line at the position of the north tab. After connecting the terminals on the tab, we move the sweep line downward. The line stops when it meets terminals. These terminals will be connected to the subtree constructed so far above the sweep line (See Figure 5). Depending on the length of the connection, each terminal t can either be connected to the subtree by the boundary edge incident to it or t can be connected to the other boundary side. The connection to the other boundary side should be the shortest path from t to any point of the boundary side. In addition to a complete interior line, this connection may also contain a part of the boundary edge incident to t. If the connection only contains an interior line, the other endpoint of this line may not be connected to the subtree yet. This endpoint should be added to the subtree. If the connection contains a boundary edge and an interior line, then the interior line ends at an inner boundary corner. This boundary corner is already in the subtree. We move the sweep line downward further after adding into the subtree all terminals on the line. We repeat this procedure until the south tab is reached. This produces an RST which has the minimum length under the condition that only interior lines parallel to the single tab are used.

Similarly, we sweep a vertical line from the west tab and a vertical line from the east tab. Both lines will be moved towards the north tab. It can be shown that the interior lines produced by the three separate sweeps do not cross each other. Therefore, an MRST can be constructed by putting the three portions together that are behind the innermost interior lines and produced by corresponding sweeps. The longest edge on the cycle composed of these three interior lines and boundary edges must be removed. The entire procedure takes a time that is linear in the number of nodes being treated.

Construction of an Extended One-Odd Corner

For each possible one-odd corner, we first fix the minor leg. It should be located on one of the $O(k)$ blue lines by Theorem 2. After the minor leg is chosen, Lemma 5 shows how to find the edge incident to the minor leg and Lemma 4 shows how to determine the position of the major leg (there are at most two possible positions). Then we can use the proof of Lemma 3 to find all edges incident to the major leg. All these operations take $O(n)$ time. Finally, the terminals enclosed by the facets of the corner can be connected independently. Each subproblem has a triple tab, since facets of an one-odd corner cannot enclose more than one tab of the original boundary. Therefore, all subproblems can be solved by the algorithm mentioned above. The running time for each subproblem is linear in the number of terminals to be connected, and hence, it is $O(n)$ for all subproblems. It takes $O(n)$ time to construct a minimum-length RST which has a particular one-odd corner. By Theorem 2 and Lemma 5, there are $O(k)$ different one-odd corners. Totally, the algorithm takes $O(kn)$ time to find a minimum-length RST among all of those RST which have an one-odd corner.

Construction of Extended Components of Other Types

Extended components of all other types can be constructed in a similar way. However, as shown in Figure 1, there may be one subproblem that does not contain a triple tab, if the topological component is a line with incident edges, a zero-odd corner or an one-even corner. Zero-even corners may even define two such subproblems. In the next section, we will show how to solve these subproblems. Therefore, these extended components usually are not complete Steiner trees. It takes $O(n)$ time to construct one extended component of any type. In case of zero-even corners, there may be $O(k)$ such corners with the minor leg intersecting the same node. Lemma 3 implies that all of them can be constructed in $O(n)$ time.

5 An $O(k^2n)$ algorithm

In this section we present an algorithm which finds an MRST for n terminals on the boundary of a convex polygon of k sides in $O(k^2n)$ time. First of all, we extend topological components further to a structure called tree blocks. Tree blocks can be put side by side to form an RST. Then we choose a minimum-length block among the blocks which contain the same boundary portions. It takes $O(k^3)$ time. After that a dynamic programming procedure combines the $O(k^2)$ selected tree blocks to find an MRST in $O(k^2)$ time.

Extended one-even corners are already complete Steiner trees and other extended components may also be complete trees. A facet is said to be *crucial* if it encloses two tabs. In general, extended lines with incident edges and extended zero-odd and one-even corners can have one crucial facet. Zero-even corners can have up to two crucial facets. The terminals enclosed by these facets are not connected in extended components. We put two or more extended components together to create a full tree. Two extended components are said to be disjoint if their boundary portions do not overlap. There are two possibilities for crucial facets to intersect the boundary: one is to intersect SW- and NE-sides, the other is to intersect NW- and SE-sides. Facets in the same tree have to take the same "orientation", otherwise they will cross each other. We will only consider the case where facets intersect SW- and NE-sides. The other case can be treated in a similar way.

It should be noted that corners with no incident edges are also basic components and they are called *simple corners.* A extended simple corner is the corner itself. The same holds for lines of Type 1 that are located on blue lines and intersect opposite sides of the boundary. They have two crucial facets.

A facet is called a *left facet* if it encloses the west tab. We associate each extended component with its left facet. Usually, there are more than one component having the same left facet. In the following we show that there are $O(k)$ left facets. We begin with simple corners.

A simple corner intersects inner boundary corners on opposite boundary sides. From each inner boundary corner there can be $O(k)$ simple corners which intersect the opposite boundary. But only $O(1)$ of them may occur in MRSTs. We call the portion of boundary between two neighboring terminals an *interval.* An interval consists of one or two (a vertical and a horizontal) boundary edges.

Lemma 6 *Let C and C' be two simple corners which begin at node a and end at node b and node b', respectively (See Figure 6). Let $B(t,t')$ be the smallest boundary portion including b and b'. If $len(C) > len(C')$ and the maximal interval length in $B(t,t')$ is smaller than $len(C)$, then C cannot be a part of any MRST.*

Assume for contradiction that there is an MRST containing C. If boundary edges of an interval are not present in the tree, we can replace C by these edges to reduce the total length. That is because the length of the missing boundary edges is smaller than $len(C)$. If all boundary edges of $B(t,t')$ are edges of the MRST, we can replace C by C' to reduce the length. ∎

A simple corner C is called *locally minimal,* if there is no corner C' which together with C satisfies the conditions of Lemma 6. This lemma indicates that only locally minimal simple corners have to be considered. If there are more than one locally minimal corner of the same length from the same node, we break ties by choosing the one with the leftmost ending node.

Lemma 7 *There are at most five locally minimal simple corners from a single inner boundary corner.*

Proof idea: At most five corners from a single node can intersect intervals which are longer than the corners. ∎

Lemma 8 *There are $O(k)$ different left facets.*

Proof: Left facets can be divided into four groups: (1) simple corners, (2) facets composed of two edges, (3) facets composed of the minor leg and the first edge incident to the major leg of a corner, and (4) facets composed of edges incident to both legs of a corner.

Lemma 7 shows that each node is associated with at most five simple corners. $O(k)$ nodes can be associated with $O(k)$ simple corners. In each of the remaining groups, a facet can be uniquely associated with one of its endpoints. For instance a facet in (2) consists of an edge incident to a line or a major leg. By Lemma 3 the choice of the line (leg) also determines the position of the incident edge. Therefore there are also $O(k)$ facets from these groups. ∎

If an RST consists of more than one extended topological components, we can order them according to the left facets. Two neighboring components may still have to be connected by some boundary edges and lines that are not located on blue lines. Finding this connection can be considered as a special problem which bounded by two crucial facets and the boundary edges between them. Both facets form two paired tabs of the new boundary. Therefore, the new boundary has only two sides, which are portions of the original boundary. The solution of such a problem is called a *glue* of basic components. In [12] algorithms are presented for a less restricted problem, called the *simple Steiner tree problem.* A simple Steiner tree is composed of boundary edges and lines of Type 1. If the boundary has no more than three sides, a minimal simple Steiner tree can be constructed in $O(n)$ time after an $O(n+k^2)$ time preprocessing. Because there are $O(k)$ crucial facets, the number of different glues is bounded by $O(k^2)$. By applying the algorithm in [12] we can construct all glues in $O(k^2n)$ time.

Now we match an extended component with a glue whose left boundary is also the crucial facet of the component. The structure thus produced is called a *tree block.* Blocks can then be put side by side to build a complete tree. Each tree block begins with a left facet F_1 and ends with another left facet F_2 (See Figure 7). There may exist $O(k)$ different basic components which have F_1 as their left facet. Each of these components can be combined with a glue to form a tree block between F_1 and F_2. We try all of the $O(k)$ combinations to find out the minimum-length tree block $TB(F_1, F_2)$ between F_1 and F_2. It takes $O(k)$ time for each pair (F_1, F_2) of left facets. Because there are $O(k)$ left facets, the number of pairs (F_1, F_2) is $O(k^2)$. All minimum-length tree blocks can be constructed in $O(k^3)$ time. This does not exceed the aimed time complexity of $O(k^2n)$.

Finally, we use the $O(k^2)$ minimum-length tree blocks to construct an MRST by applying dynamic programming. It may be recalled that all left facets which we now consider intersect the SW- and NE-sides. These facets can be ordered, mainly according to their positions on the SW-side. If two left facets intersect the same node on the SW-side, they will be ordered according to their endpoints on the NE-side. Suppose $F_1, \cdots, F_m$ is the sequence of all left facets sorted according to the order from right to left. Let $H(F_i)$ denote the minimum-length subtree which contains the half of the boundary to the left of F_i. The dynamic programming algorithm computes $H(F_i)$ from left to right. For each $H(F_i)$, it compares all combinations $H(F_j) \cup TB(F_i, F_j)$, for $j < i$. This step takes $O(k^2)$ time.

It can, therefore, be seen that the total amount of time to carry out all the steps involved in finding an MRST is $O(k^2 n)$

6 Conclusion

We have given an $O(k^2 n)$ time algorithm to construct a minimal rectilinear Steiner tree for a set of n terminals located on the boundary of a convex rectilinear polygon of k sides. The bottleneck in the present algorithm is the procedure for tree blocks. It takes $O(k^3)$ time to construct $O(k^2)$ minimum-length blocks. We are now working on speeding up this procedure and expect to reduce the total running time to $O(kn)$ in the near future.

References

[1] P. K. Agarwal and M. T. Shing, "Algorithms for special cases of rectilinear Steiner trees: I. Points on the boundary of a rectilinear rectangle", *Networks,* **20**, pp. 453-485, (1990).

[2] A. V. Aho, M. R. Garey, and F. K. Hwang, "Rectilinear Steiner trees: Efficient special-case algorithms", *Networks,* **7**, pp. 37-58, (1977).

[3] M. W. Bern, "Network design problems: Steiner trees and spanning k-trees", PhD Dissertation, University of California, Berkeley, (1987).

[4] M. W. Bern, "Faster exact algorithms for Steiner trees in planar networks", *Networks,* **20**, pp. 109-120, (1990).

[5] J. P. Cohoon, D. S. Richards, and J. S. Salowe, "An optimal Steiner tree algorithm for a net whose terminals lie on the perimeter of a rectangle", *IEEE Transactions on Computer-Aided Design*, **9**, pp. 398-407, (1990).

[6] S. E. Dreyfus and R. A. Wagner, "The Steiner problem in graphs," *Networks,* **1**, pp. 196-207, (1972).

[7] M. R. Garey and D .S. Johnson, "The rectilinear Steiner tree problem is NP-complete," *SIAM J. Appl. Math.* **32**, pp. 826-834, (1977).

[8] M. Hanan, "On Steiner's problem with rectilinear distance", *SIAM J. Appl. Math.* **14**, pp. 255-265, (1966).

[9] F. K. Hwang, "On Steiner minimal trees with rectilinear distance", *SIAM J. Appl. Math.* **30**, pp. 104-114, (1976).

[10] F. K. Hwang and D. S. Richards, "Steiner tree problems", *Networks,* **22**, pp. 55-89, (1992).

[11] J. S. Provan, "Convexity and the Steiner tree problem", *Networks* **18**, pp. 55-72, (1988).

[12] D. S. Richards and J. S. Salowe, "A linear-time algorithm to construct a rectilinear Steiner tree for k-extremal points", *Algorithmica,* **7**, pp. 246-276, (1992).

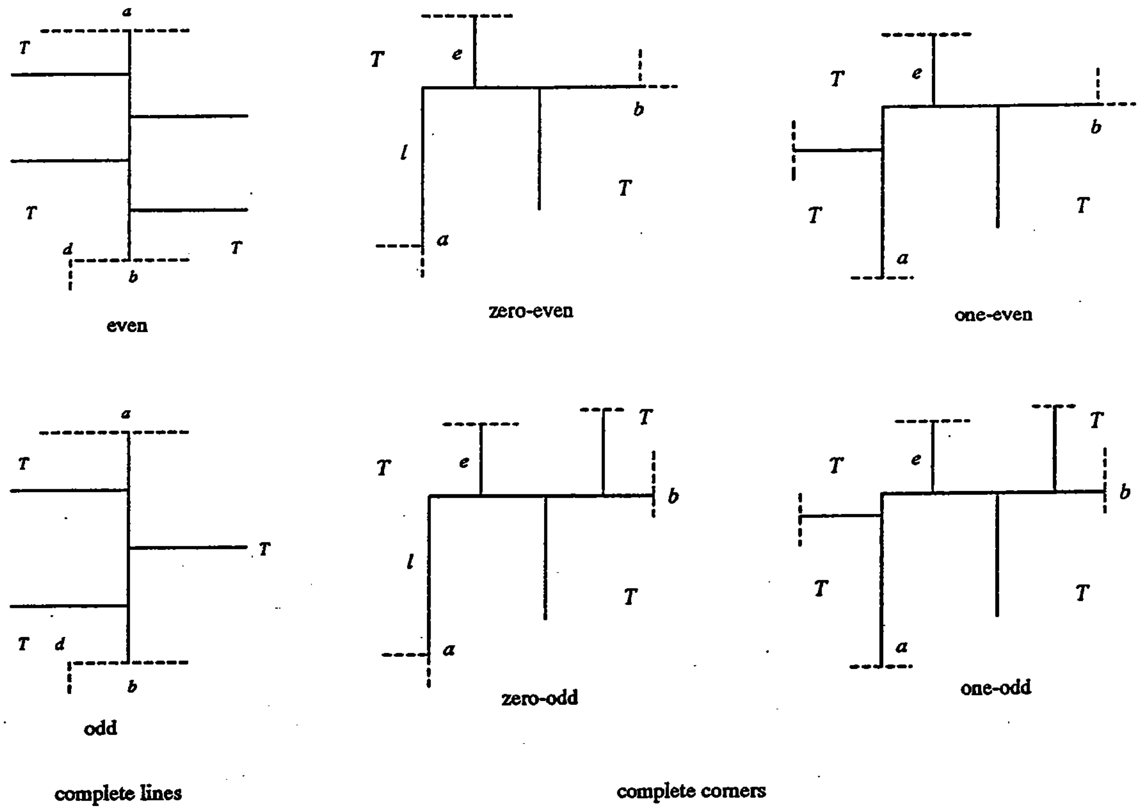

Figure 1: Topological components

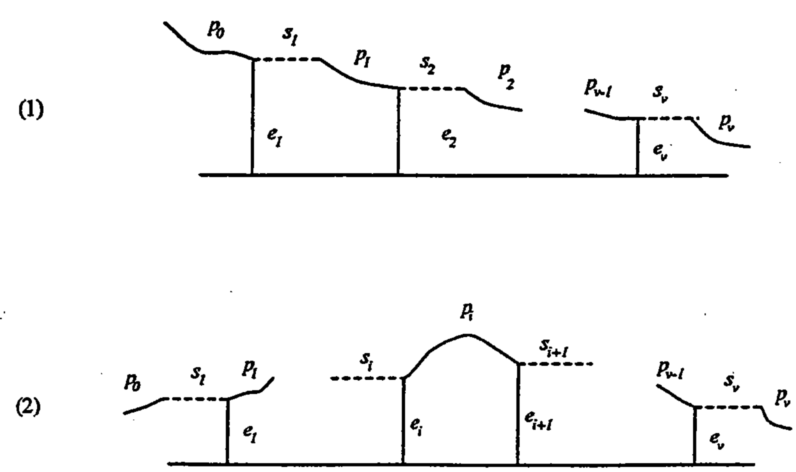

Figure 2: Lines with incident edges (only the upper edges are shown). (1) The boundary portions do not contain any tab. (2) The boundary portions contain a tab.

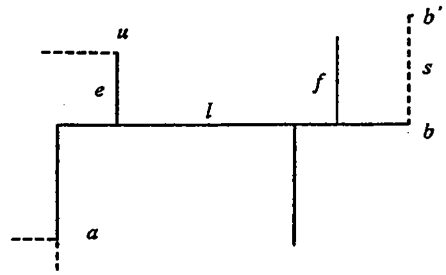

Figure 3: A zero-odd corner.

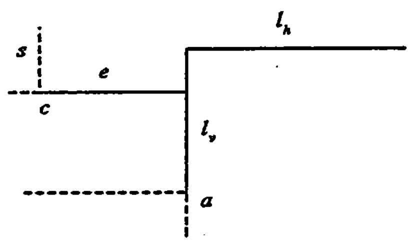

Figure 4: The minor leg of a one-odd corner.

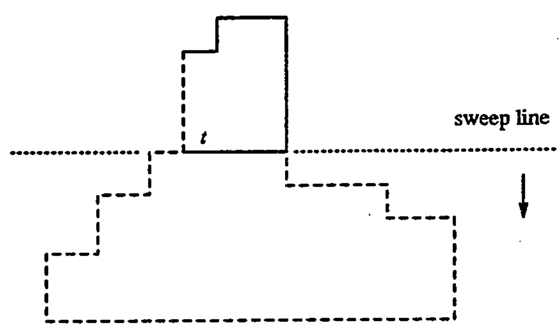

Figure 5: Boundary with a triple tab.