Journal of Global Optimization **21:** 385–396, 2001.

Polynomial Time Approximation Scheme for Symmetric Rectilinear Steiner Arborescence Problem

XIUZHEN CHENG[1], BHASKAR DASGUPTA[2] and BING LU[1]
Computer Science Department, University of Minnesota, Minneapolis, MN 55455, USA (e-mail: cheng,blu@cs.umn.edu); [2]*Rutgers University campus at Camden (e-mail: bhaskar@camden.rutgers.edu)*

(Received 4 January 2001; accepted in revised form 5 April 2001)

Abstract. The Symmetric Rectilinear Steiner Arborescence (SRStA) problem is defined as follows: given a set of terminals in the positive quadrant of the plane, connect them using horizontal and vertical lines such that each terminal can be reached from the origin via a y-monotone path and the total length of all the line segments is the minimum possible. Finding an SRStA has applications in VLSI design, in data structures used in some optimization algorithms and in dynamic server problems. In this paper, we provide a polynomial time approximation scheme for the SRStA problem, improving the previous best approximation ratio of 3 for this problem.

Key words: PTAS, Rectilinear Steiner Arborescence, Symmetric Rectilinear Steiner Arborescence, Guillotine, Approximation Algorithm

1. Introduction

In spite of large progress in the recent years, there is a number of gaps in our knowledge about the exact complexities of some Steiner problems in rectilinear metric. We propose to investigate one of these problems.

The problem of finding the *Symmetric Rectilinear Steiner Arborescence* (SRStA) problem can be stated as follows. We are given a set of n terminals in the positive quadrant of the plane. A path connecting two terminals is y-*monotone* [6] if it traverses a number of line segments, where each line segment is either vertical or horizontal, and during the traversal the y coordinate of the successive points are *never decreasing*. A feasible solution to the problem is a set of horizontal and/or vertical segments connecting all the n terminals to the origin o in which each terminal can be reached from o by a y-monotone path. Our goal is to find a feasible solution in which the sum of lengths of all the segments is the minimum possible. If instead we require the path connecting o to any point to be both x-monotone and y-monotone, then the problem is referred to as the *Rectilinear Steiner Arborescence* (RStA) problem (see Figure 1).

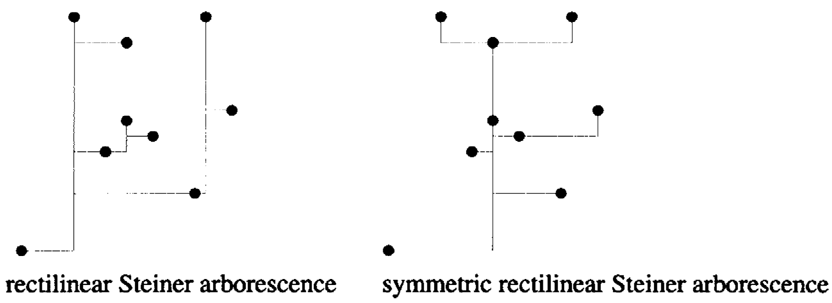

Figure 1. Rectilinear Steiner Arborescence (RStA) and Symmetric Rectilinear Steiner Arborescence (SRStA)

The history of the RStA problem is somewhat unusual, because after an exact algorithm was published by Trubin [12], Rao et al. [10] showed that this solution was in fact incorrect. Their paper describes a simple algorithm that offers approximate solutions within a factor of 2 of the optimum. The most recent results on RStA problem are the proof of NP-completeness of Shi and Su [11] and a polynomial time approximation scheme by Lu and Ruan [4].

For the SRStA, the best previously known approximation was given by Charikar et al. [2] which finds an approximate solution within factor 3 of the optimum.

In this paper, we provide a polynomial time approximation scheme (PTAS) for the SRStA problem. A PTAS for a problem of size n is an algorithm that, for every *constant* $\varepsilon > 0$, finds an approximate solution with an approximation factor of $1+\varepsilon$ in time polynomial in n. We apply the method proposed in [3,7–9]. For the sake of completeness, we briefly review the results of m-guillotine in Section 3.

1.1. MOTIVATIONS AND APPLICATIONS

The SRStA and the RStA problems have a number of applications. An application that is mentioned quite often comes from VLSI design, where a RStA or SRStA is needed to minimize the maximum delay of the signal sent from the origin o to all the given terminals. A motivation for the on-line versions of these problems come from data structures used in some optimization algorithms where an object is optimized using successive iterations [1]. The SRStA problem has direct application in the offline dynamic server problem on the line [2]. On-line arborescence problems model real-life processes that have two dimensions. Below we briefly sketch two applications of these problems.

Offline dynamic server problem on the line: We need to maintain a dynamic collection of servers on a line L. The goal is to efficiently process a sequence of requests, arriving at integer times $t \in \{1, 2, 3, \dots\}$, which are points on L, where a server serves a request by moving to that point incurring a cost equal to the distance traveled. It is possible to create and/or destroy servers without incurring any cost in the following manner: *clone* a copy of a current server at a point and *merge* two servers present at the same point on L into one. After all the requests at a particular instant of time t has been served, the algorithm is also charged an additional *rental* cost equal to the number of servers currently present. The final goal of this problem is to serve a sequence of requests such that the total cost incurred is the minimum possible. The motivation for this problem comes from the video-on-demand application of Papadimitriou et al. [5]. Theorem 4.4 in [2] essentially show that an approximation algorithm for the SRStA problem with an approximation ratio of r provides an approximation algorithm for the offline dynamic server problem on the line with an approximation ratio of $2r$.

Real-life processes in two dimensions: As discussed by Berman and Coulston [1], as well as by Charikar et al. [2], on-line arborescence problems model real-life processes that have two dimensions: dimension x refers to location on a delivery route, DNA sequence etc., while dimension y refers to time. We can maintain supplies of a resource (like cache of videos or a saved precomputed instance of dynamic programming) on various places in x dimension, and then we receive at various times request for delivering the resource. There are costs associated with the storage of the resource, and with the distance traversed during a delivery. Dependent whether the movement can occur in one or two directions, we obtain an online RStA or SRStA.

2. Preliminary

Unless otherwise stated, all terminals lie in the positive quadrant of the plane. Given a set N of terminals, the Hanan grid $H(N)$ is the grid obtained by constructing horizontal and vertical lines through each point in N. Furthermore, it is bounded by x-axis, y-axis, the horizontal line through the highest point and the vertical line through the rightmost point. Let $I_{H(N)}$ denote the set of intersections in $H(N)$. These intersections are called *Hanan grid points*. It is obvious that $N \subseteq I_{H(N)}$.

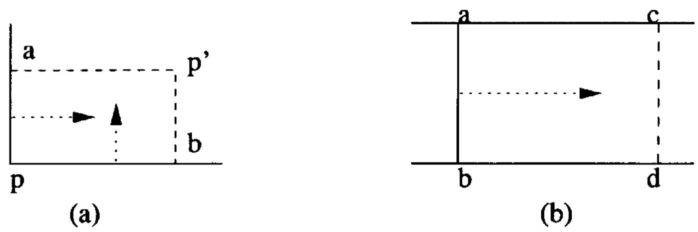

Figure 2. two operations: (a) flipping and (b) shifting.

Two operations are defined for RStA or SRStA (Figure 2): flipping and shifting. Flipping a corner p between two points a and b adjacent to p (Figure 2(a)) moves

pb to ap' and moves pa to bp'. Shifting a line segment ab moves ab along either axis direction until it is incident to a certain specified point (Figure 2(b)).

THEOREM 1 *There exists an optimal SRStA R^* such that every Steiner point in R^* belongs to $I_{H(N)}$.*

Proof. Suppose that R is any optimal SRStA with at least one Steiner point not belonging to $I_{H(N)}$. Let S denote the set of Steiner points not in $I_{H(N)}$. We will modify R recursively until all Steiner points in S are moved to Hanan grid points. Choose $p \in S$ such that $D(o, p) \geq D(o, s)$ for $\forall s \in S$, where $D(o, t)$ denote the path length from the origin o to point t in R. For an internal point p_i of an SRStA, the *in-edge* of p_i is defined as the edge $p_p p_i$ where p_p is p_i's unique parent. The *out-edge* of p_i is defined as the edge $p_i p_c$ where p_c is one of p_i's children. Note that the root of SRStA o can only have out-edges and the leaves of SRStA can only have in-edges. The *in-degree* of a point is defined as the number of in-edges incident to it, which is always one; the *out-degree* of a point is defined as the number of out-edges incident to it. Note that each Steiner point of SRStA has either degree three (one in-edge and two out-edges) or degree four (one in-edge and three out-edges). We prove the theorem by showing that R can be converted into another optimal SRStA R' such that all points of R' are in $I_{H(N)}$. Figures 3, 4, 5 explain all possible cases of out-edges at p. Note that in these figures, we use a directed edge from a to b to show that a is the parent of b.

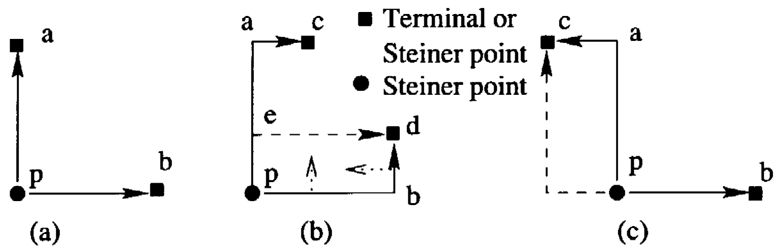

Figure 3. Proof of Theorem 1. Case 1: (a) neither of the out edges is a corner line; (b) either or both out-edges are corner lines (left to right or bottom to top); (c) only one out-edge is a corner line (right to left).

Case 1: One out-edge points from left to right and one out-edge points from bottom to top (see Figure 3). As shown in Figure 3(b) it is impossible for p to have corner line pc or pd where ac points from left to right and bd points from bottom to top. Otherwise, by flipping the corner a or b, an SRStA with less total length can be obtained contradicting that R is optimal. However, p may have a corner line pc as shown in Figure 3(c). By flipping the corner a, it becomes case 3 which will be discussed later. If neither of the out-edges is a corner line (Figure 3(a)), then the endpoints a and b must be either terminals or Steiner points in $I_{H(N)}$ according to our criteria of choosing p. If this is true, then p must be in $I_{H(N)}$, a contradiction.

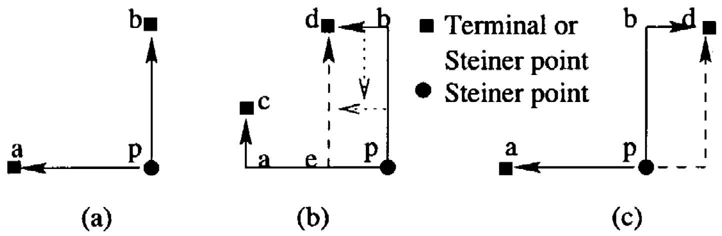

Figure 4. Proof of Theorem 1. Case 2: (a) neither of the out edges is a corner line; (b) either or both out-edges are corner lines (right to left or bottom to up); (c) only one out-edge is a corner line (left to right).

Case 2: One out-edge points from right to left and one out-edge points from bottom to top (see Figure 4). As shown in Figure 4(b) it is impossible for p to have corner line pc or pd where ac points from bottom to top and bd points from right to left. Otherwise, by flipping the corner a or b, an SRStA with less total length can be obtained contradicting that R is optimal. However, p may have a corner line pd as shown in Figure 3(c). By flipping the corner b, it becomes case 3 which will be discussed later. If neither of the out-edges is a corner line (Figure 3(a)), then the endpoints a and b must be either terminals or Steiner points in $I_{H(N)}$ according to our criteria of choosing p. If this is true, then p must be in $I_{H(N)}$, a contradiction.

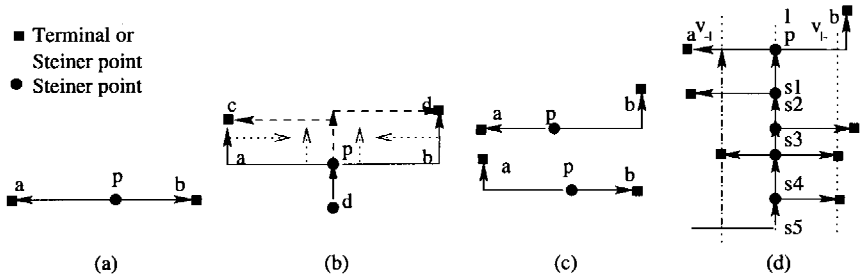

Figure 5. Proof of Theorem 1. Case 3: (a) neither of the out-edges is a corner line; (b) both out-edges are corner lines; (c) only one out-edge is a corner line; (d) move the line segment crossing p to left or right to make p lie in a Hanan grid point.

Case 3: One out-edge points from left to right and one out-edge points from right to left (see Figure 5). As shown in Figure 5(b), it is impossible for p to have two corner lines pc and pd. Otherwise, by flipping the corners a and b, an SRStA with less total length can be obtained contradicting that R is optimal. However, one of the out-edges of p may be a corner line (Figure 5(c)). Furthermore, we can assume that the in-edge of p is not a corner line. Otherwise, we can also get an SRStA with less total length by flipping the corner, contradicting to the optimality of R. In Figure 5(a)–(d), both a and b are in $I_{H(N)}$ according to our criteria of choosing p.

Let l be the vertical line through p. It is obvious that R overlaps with l in a set of closed intervals. The interval containing p is picked and let

$S = \{s_0 = p, s_1, s_2, \dots, s_t\}$ $(t \geq 1)$ be the set of points of R contained in the interval and $y_{s_0} > y_{s_1} > y_{s_2} > \cdots > y_{s_t}$ where y_{s_i} is the y-coordinate of s_i for $i = 0, 1, 2, \dots, t$. Note that s_t may be a corner. No s_i is in $I_{H(N)}$ since p is not in $I_{H(N)}$. Furthermore, let $H^{\dashv} = \{h_1^{\dashv}, h_2^{\dashv}, \dots, h_m^{\dashv}\}$ denote the set of horizontal segments in R incident on the points in S and are to the left of l. Similarly, let $H^{\vdash} = \{h_1^{\vdash}, h_2^{\vdash}, \cdots, h_n^{\vdash}\}$ be the set of horizontal segments in R incident on the points in S and are to the right side of l. It is easy to show that $m = n$ and each horizontal segment in $H^{\dashv}$ or $H^{\vdash}$ is on the Hanan grid by the optimality of R.

Let $v_{\dashv}$ ($v_{\vdash}$) be the closest vertical line in $H(N)$ on the left (right) of l (shown as dotted lines in Figure 5(d)). Therefore, all segments in $H^{\dashv}$ ($H^{\vdash}$) must intersect $v_{\dashv}$ ($v_{\vdash}$). Moreover, shifting the segment $s_0 s_t$ left or right between $v_{\dashv}$ and $v_{\vdash}$ will not change the total length of the resulting SRStA since $m = n$. Assume, without loss of generality, that $s_0 s_t$ is shifted to $v_{\dashv}$ and the overlapped Steiner points are removed, then each s_i, $i = 0, 1, \dots, t$, will be either in $I_{H(N)}$ or removed. Thus, a new optimal SRStA can be constructed using at least one fewer Steiner point not in $I_{H(N)}$.

Now the new SRStA has at least one fewer Steiner point not in $I_{H(N)}$ and its cost is the same as that of the original SRStA. Continue this procedure until all points not in $I_{H(N)}$ are considered. We will get an optimal SRStA R' such that all Steiner points in R' are Hanan grid points. □

3. *m*-Guillotine Subdivision

Du et al. [3] studied rectangular subdivisions and introduced the concept of 'guillotine' subdivision and claimed that any rectangular subdivision with cost L can be converted into a guillotine rectangular subdivision with cost at most $2L$ by adding a set of new edges whose total length is at most L. Moreover, the cost of the new edges is charged off to the original edge set of the subdivision. Mitchell [**?**] extended these concepts and ideas by defining m-guillotine subdivision and proving that an m-guillotine subdivision with cost at most $(1 + \frac{1}{m}) \cdot L$ can be obtained from a rectilinear subdivision whose cost is L. With m-guillotine subdivision, Mitchell [8, 9] found PTASs for various geometric optimization problems: TSP, Steiner Minimum Tree and k-MST, etc.

We will use m-guillotine subdivision to design a PTAS for SRStA problem. For simplicity and convenience, we will use similar notations as those in [8, 9].

Let $\mathcal{R}$ be a bounded rectilinear polygon with rectilinear holes: non-overlapping rectilinear polygons, rectilinear trees and points. A *rectilinear polygonal subdivision* R of $\mathcal{R}$ is defined as a finite set of non-crossing horizontal and vertical segments that lie inside $\mathcal{R}$. Without loss of generality, we assume that a rectilinear polygonal subdivision R is restricted to the unit square, B. Let E denote the set of edge segments of R and V denote the set of vertices of R. A *window* is defined as an axis-aligned and bounded rectangle W and $W \subseteq B$. A line (horizontal or vertical) l is a cut of E if $l \cap int(W) \neq \emptyset$. Let ξ be the number of intersections of a cut line l with $E \cap int(W)$. The intersections are denoted by $p_1, p_2, \dots, p_\xi$ along l. For a cut l, the m-span $\sigma_m(l)$ of l is defined as: if $\xi \leq 2(m-1)$, then $\sigma_m(l) = \emptyset$; Otherwise, $\sigma_m(l)$ is the line segment $p_m p_{\xi-m+1}$, where m is a positive integer. If $\sigma_m(l) \subseteq E$, the cut l is an m-perfect cut with respect to W. A rectilinear polygonal subdivision R is an *m-guillotine subdivision* with respect to W if either $E \cap int(W) = \emptyset$ or R is an m-guillotine subdivision with respect to windows $W \cap P^+$ and $W \cap P^-$, where P^+ and P^- are the closed half-planes induced by a perfect cut l. R is an m-guillotine if it is an m-guillotine with respect to B. A point p on a cut l is m-dark with respect to l and W if there are at least m intersections with E on each side of p along $l^\perp \cap int(W)$, where $l^\perp$ is perpendicular to l and passes through p. A cut l is favorable if the total length of the m-dark portion of l is at least as that of $\sigma_m(l)$. An optimal m-guillotine rectilinear subdivision can be found by dynamic programming in polynomial time.

In [8], Mitchell provided the following lemma and theorem which assert the existence of favorable cut lines and $(1 + \frac{1}{m})$ factor m-guillotine subdivision respectively. For completeness, we give the proof here.

LEMMA 1 *There exists a favorable cut for any rectilinear polygonal subdivision R and window W.*

Proof. As we assumed before, B is a unit square. Let $f(x)$ $(g(y))$ be the length of the m-span of the vertical(horizontal) line through x (y) where $x, y \in [0, 1]$. Sets R_x and R_y contain all points of B which are m-dark with respect to horizontal and vertical cuts, respectively. $A_x = \int_0^1 f(x)dx$ $(A_y = \int_0^1 g(y)dy)$ is the area of R_x (R_y). Without loss of generality, assume $A_x \geq A_y$. Note that the area of the region R_x can also be calculated as $A_x = \int_0^1 h(y)dy$, where $h(y)$ is the length of the intersection of R_x with a horizontal line through y. Therefore, $\int_0^1 h(y)dy \leq \int_0^1 g(y)dy > 0$. Thus, it is impossible that for all $y \in [0, 1]$, $h(y) < g(y)$. So there must exist a $y' \in [0, 1]$ such that $h(y') \geq g(y')$. The horizontal line through y' is a favorable cut. □

Based on this lemma, Mitchell [8] further proved the following theorem.

THEOREM 2 *Given any $m > 0$, for a rectilinear subdivision R with edge set E of length L, there exists an m-guillotine rectilinear subdivision R_G with edge set E_G of length L_G. Furthermore, $E \subseteq E_G$ and $L_G \leq (1 + \frac{1}{m})L$.*

Proof. R will be recursively converted into an m-guillotine subdivision R_G by adding a set of horizontal/vertical edges E'. The total length of E' is at most $\frac{1}{m}L$. If there exists a perfect cut l, then we can choose it and recursively proceed on each side of l. Otherwise, we choose a favorable cut line l. We assume, without loss of generality, that l is horizontal. For an open m-dark subsegment ab of l, we can charge off $\frac{1}{2m}$ of the length of ab to each of the first m subsegments lying above ab

and to each of the first m subsegments lying below ab. The m-span of l is added to the new edge set and l becomes the boundary of new child windows. Thus, no partion of E will be charged more than once from each side. Since the total length of all m-spans of all favorable cuts is at most $\frac{1}{m}L$, the total length of the new edge set is at most $\frac{1}{m}L$. □

4. Main Result

THEOREM 3 *There is an approximation algorithm for the SRStA problem that runs in $O(n^{10m+5})$ time and produces a solution with an approximation ratio of at most $1 + \frac{1}{m}$ for any fixed positive integer m.*

Proof. We assume, without loss of generality, that no two points of the input set N of n points lie on a common horizontal line or a vertical line. Otherwise, the points in N can be slightly perturbed. Let R^* be an optimal SRStA with edge set E_{R^*} of total length L_{R^*}. The proof of the theorem consists of two steps.

Step 1. R^* is transformed into an m-guillotine rectilinear subdivision R which is also a feasible SRStA. In other words, R is a rectilinear Steiner arborescence which is y-monotone. We call R an *m-guillotine SRStA*. The cost of R is at most $\left(1 + \frac{1}{m}\right) \cdot L_{R^*}$. Denote the edge set of R by E_R whose cost is L_R.

Step 2. Due to the recursive structure of m-guillotine rectilinear subdivision, we can apply dynamic programming to find the optimal m-guillotine SRStA.

Step 1 proves the existence of the m-guillotine SRStA while Step 2 finds the optimal m-guillotine SRStA. The following two subsections demonstrate these two steps in detail. The running time of dynamic programming is $O(n^{10m+5})$ and $L_R \leq (1 + \frac{1}{m})L_{R^*}$. □

4.1. THE EXISTENCE OF THE m-GUILLOTINE SRSTA WITH COST AT MOST $\left(1 + \frac{1}{m}\right) \cdot l_{r^*}$

It is obvious that R^* is a bounded rectilinear polygon with rectilinear holes. From the proof of Theorem 2, m-spans of cut lines can be added to the edge set E^* to make R^* m-guillotine. However, when we add these line segments to E^*, we must modify the current rectilinear polygonal subdivision to make it feasible. That is, when we add a line segment (an m-span of some cut line), we must force the result graph to be a y-monotone rectilinear Steiner arborescence. An m-guillotine subdivision which is also an SRStA is referred to as an m-guillotine SRStA.

LEMMA 2 *Let R be an SRStA with cost L_R and S be an m-span (vertical or horizontal) with length s of a cut line l of R. Then, R can be modified to another feasible SRStA R' such that R' contains S and the cost of R' is at most $L_R + s$.*

Proof. We will consider the following two cases.

Case 1. S is horizontal (Figure 4.1(a)). Assume S crosses R at $p_1, p_2, \ldots, p_t$, with increasing x-coordinates, and p_1, p_t are the two end points of S. If $t = 1$, we are done. Now assume $t > 1$. For each p_i, $i = 1, 2, \ldots, t$, if p_i is not a terminal, add p_i to R as a Steiner point whose degree is at least 3. Pick p_c, the first point whose in-edge is not inside $\sigma_m(l)$, as the crucial point. If $\sigma_m(l) \neq \emptyset$, then such a point must exist. Otherwise, R is not connected or there exists at least one path which is not y-monotone in R from the origin o to some point among $p_1, p_2, \ldots, p_t$. For each i, $i \neq c$, delete the in-edge of p_i if the in-edge is not inside $\sigma_m(l)$. Now each nonterminal p_i has degree at most 3. For all $i > c$, the in-edge of p_i is $p_{i-1}p_i$. For all $i < c$, the in-edge of p_i is $p_{i+1}p_i$.

Case 2. S is vertical (Figure 4.1(b)). The argument is similar to that of case 1 except that we pick p_1 as the crucial point. For each $i > 1$, we delete the in-edge of p_i if the in-edge is not inside $\sigma_m(l)$ and pick $p_{i-1}p_i$ as its in-edge.

In both modifications, only line segment S is added to R. Note that the modification can introduce nonterminal point whose degree is 1 or 2. Thus we need to prune the graph by deleting this kind of edges and remove nonterminal point with degree 2 to make all Steiner points in the result SRStA have degree at least 3. The resulting feasible SRStA has cost at most $L_R + s$. □

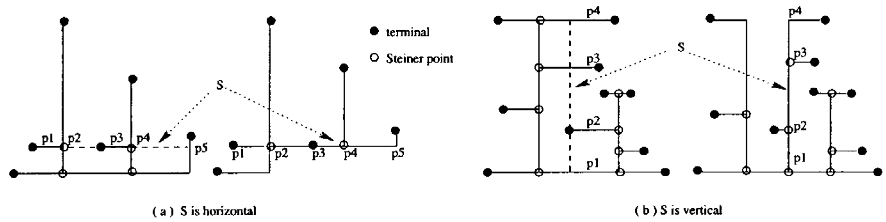

Figure 6. A line segment with length s is added to an SRStA. The result graph is a feasible SRStA with cost increase at most s. In (a), p_2 is the crucial point; in (b), p_1 is the crucial point. Note that in (a), p_3 is a terminal. Before the addition of S, its in-edge is p_4p_3; After the addition of S, its in-edge becomes p_2p_3.

This lemma forces the feasibility of a symmetric rectilinear Steiner arborescence when line segments are added to a feasible SRStA. The following lemma describes how to make an optimal SRStA m-guillotine with little cost increase.

LEMMA 3 *There exists a $\left(1 + \frac{1}{m}\right)$-approximate feasible SRStA which is an m-guillotine subdivision.*

Proof. We start from an optimal SRStA R^* and modify it until it is m-guillotine. The proof is similar to that of Theorem 2, except that when we add the m-span of each cut line, we need to apply Lemma 2 to force the feasibility of the result subdivision. Note that the cost increase due to the addition of m-spans is charged off to the original edge set E_{R^*}, and the m-perfect or m-span cut lines are chosen according to the original SRStA R^*. The resulting division is an SRStA whose cost is increased by a factor of at most $\frac{1}{m}$. □

4.2. DYNAMIC PROGRAMMING

During the transformation of R^*, if there exists a perfect cut line, it can be used directly. Otherwise, there always exists a favorable cut line l by Lemma 1 and its m-span $\sigma_m(l)$ can be chosen as new segment to be added into E^*. Furthermore, the favorable cut line can be selected to pass through either a terminal in N or the midpoint of some horizontal or vertical interval defined by consecutive coordinates of points in N (according to Theorem 1). The discretization of cuts and the connectedness property allow us to divide the problem into smaller subproblems and apply dynamic programming to find an optimal m-guillotine rectilinear subdivision.

Let $x_1 < x_2 < \cdots < x_{2n-1}$ ($y_1 < y_2 < \cdots < y_{2n-1}$) denote the sorted x (y) coordinates of points in N and the $n-1$ midpoints of the intervals defined by points in N. An instance of this subproblem is specified by the following inputs:

(a) A rectangle $\mathcal{R}(l, r, b, t)$ (denoted by $\mathcal{R}$) determined by x_l, x_r, y_b, y_t, where $x_l < x_r$ and $y_b < y_t$.

(b) Boundary information. At most $k \le 2m$ distinct points in each edge of $\mathcal{R}$, together with at most one segment which connects the middle two points if $k = 2m$. These points are determined by coordinates x_j, y_k where $1 \le j, k \le 2n-1$.

(c) Connectivity constraints. Defined as a partition, $\mathcal{P}$, of the set of points on all four sides of $\mathcal{R}$. In each subset of the partition, the point with smallest x and y coordinates is called a *subroot*. The SRStA containing all points in this subset must root at this point.

The goal of the subproblem is to find a minimum length m-guillotine SRStA with multiple components such that (i) each connected component is an SRStA which connects to all points in some subset in $\mathcal{P}$ and some terminals inside $\mathcal{R}$ and which is rooted at the subroot of the subset; (ii) all components contain only horizontal and vertical lines lying inside $\mathcal{R}$ and connect all terminals inside $\mathcal{R}$, all boundary points and the possible boundary segment, if it exists, according to the partition $\mathcal{P}$; (iii) collectly all connected components form multiple m-guillotine SRStAs. The total number of subproblems is bounded by $O(n^4 \cdot (n^{2n})^4)$ since the number of partitions is $O(1)$ for fixed m.

The initial problem is specified by the bounding box with no nonterminal points in the boundary and the connectivity constraint is empty. Note that there are at least one terminal residing in the bounding box $\mathcal{B}$ and the origin o in N must be located in the lower edge of $\mathcal{B}$. The output is one connected component which is an m-guillotine SRStA connecting all terminals inside $\mathcal{R}$. The base subproblem is specified by (a′) a rectangle $\mathcal{R}$ containing no terminal inside; (b′) constant number of points (at most $8m$) and constant number of segments (at most 4) in the boundary of $\mathcal{R}$; (c′) constant number of boundary connectivity constraints. Thus it can be solved in a brute-force maner. For all other subproblems, we can find the m-guillotine SRStA inductively, optimizing over the set of subproblems which are defined by (a′) a cut line which divides $\mathcal{R}$ into two rectangles; (b′) at most $2m$ points, and at most one segment which connects to the two middle points if the number of points is exactly $2m$, in the cut line. (c′) $O(1)$ choices of boundary connectivity constraints for the two new rectangles. Note that the partition must respect to the original partition in (c).

Each subproblem takes time $O(n^{2m+1})$ since there are $O(n)$ choices of a cut line in (a') and $O(n^{2m})$ choices of points in the cut line. As mentioned previously, the number of subproblems is bounded by $O(n^{8m+4})$. Thus the total running time for dynamic programming is $O(n^{10m+5})$.

5. Conclusion and Open Problems

The NP-completeness proof for RStA is rather delicate and it is quite open whether SRStA is NP-complete or not. Of larger practical importance is the question of what approximation ratios can be obtained for the SRStA problem in better than $O(n^2)$ time.

References

1. Berman, P. and Coulston, C. (1997), On-line Algorithms for Steiner Tree Problems, *Proc. of the 29th ACM Symp. on Theory of Computing*, pp. 344–353.
2. Charikar, M., Halperin, D. and Motwani, R. (1998), The Dynamic Servers Problem, *Proc. of 9th ACM-SIAM Symp. on Discrete Algorithms*, pp. 410–419.
3. Du, D.-Z., Pan, L.-Q. and Shing, M.-T. (1986), Minimum edge length guillotine rectangular partition. Report 02418-86, Math. Sci. Res. Inst., Univ. California, Berkeley, CA.
4. Lu, B. and Ruan, l. (2000), Polynomial Time Approximation Scheme for Rectilinear Steiner Arborescence problem, *J. of Combinatorial Optimization* 4: 357–363.
5. Papadimitriou, C. H., Ramanathan, S. and Rangan, P.V. (1995), Optimal Information Delivery, *Proc. of the 6th Annual International Symposium on Algorithms and Computation*, pp. 181–187.
6. Preparata, F.P. and Shamos, M.I. (1988), *Computational Geometry — An Introduction*, Springer-Verlag, Berlin.
7. Mitchell, J.S.B. (1996), Guillotine subdivisions approximate polygonal subdivisions: a simple new method for the geometric k-MST problem. *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pp. 402–408.
8. Mitchell, J.S.B. Guillotine subdivisions approximate polygonal subdivisions: a simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. Manuscript, Available at http://ams.sunysb.edu/~jsbm/jsbm.html
9. Mitchell, J.S.B. Guillotine subdivisions approximate polygonal subdivisions: Part II – a simple polynomial-time approximation scheme for geometric k-MST, TSP, and related problems. Available at http://ams.sunysb.edu/~jsbm/jsbm.html

10. Rao, S., Sadayappan, P., Hwang F., and Shor, P. (1992), The Rectilinear Steiner Arborescence Problem, *Algorithmica*, pp. 277–288.
11. Shi, W. and Su, C. (2000), The rectilinear Steiner Arborescence Problem is NP-complete, *Proc. 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 780–787.
12. Trubin, V.A. (1985) Subclass of the Steiner Problems on a Plane with Rectilinear metric, *Cybernatics* 21: 320–322 (translated from *Kibernetika* 21(3): 1985, pp. 37–40).