Journal of Universal Computer Science, vol. 13, no. 11 (2007), 1615-1627
submitted: 10/9/07, accepted: 19/11/07, appeared: 28/11/07 © J.UCS

Analysis of two Sweep-line Algorithms for Constructing Spanning Trees and Steiner Trees[1]

Adrian Dumitrescu[2]
(Department of Computer Science, University of Wisconsin-Milwaukee, USA
E-mail: `ad@cs.uwm.edu`)

Csaba D. Tóth
(Department of Mathematics, Massachusetts Institute of Technology, USA
E-mail: `toth@math.mit.edu`)

Abstract: We give a tight analysis of an old and popular sweep-line heuristic for constructing a spanning tree of a set of n points in the plane. The algorithm sweeps a vertical line across the input points from left to right, and each point is connected by a straight line segment to the closest point left of (or on) the sweep-line. If W denotes the weight the Euclidean minimum spanning tree (EMST), the spanning tree constructed by the sweep-line algorithm has weight $O(W \log n)$, and this bound is asymptotically tight. We then analyze a sweep-line heuristic for constructing a Steiner tree, in which a vertical line is swept across the input points from left to right, and each point is connected by a straight line segment to the closest point on edges or vertices of the current tree (on the left of the sweep line). We show that this algorithm achieves an approximation ratio of $O(\log n)$, and describe a class of instances where this ratio is $\Omega(\log n/ \log\log n)$. Our results give almost complete answers to two old open questions from the 1970s.
Key Words: Minimum spanning tree, minimum Steiner tree, sweep-line, heuristic, approximation ratio
Category: F.2.2, G.2.2

1 Introduction

Let S be a finite set of points in the plane. A *Euclidean Steiner tree* (EST) for S is a planar straight line graph spanning S. The *Euclidean Steiner tree problem* asks for the shortest such graph (where the edge lengths are measured in L_2 metric). The solutions take form of a tree, that includes all the points in S, called *terminals*, or *sites*, along with possibly some extra vertices, called *Steiner points*. It is known that in an optimal solution each Steiner point has degree 3, and any two consecutive incident edges form an 120° angle [Gilbert and Pollak 1968]. A Euclidean minimum spanning tree (EMST) for a point set can always serve as a suboptimal Euclidean Steiner tree.

The *rectilinear Steiner tree problem* asks for the shortest EST using the rectilinear metric (i.e., L_1 metric), in which the distance between two points $u(x_u, y_u)$

[1] C. S. Calude, G. Stefanescu, and M. Zimand (eds.). *Combinatorics and Related Areas. A Collection of Papers in Honour of the 65th Birthday of Ioan Tomescu.*
[2] Supported in part by NSF CAREER grant CCF-0444188.

and $v(x_v, y_v)$ is $d(u, v) = |x_u - x_v| + |y_u - y_v|$. The solution can be drawn as a *rectilinear Steiner tree* (RST), composed solely of horizontal and vertical edges (where the L_1 and L_2 lengths are the same). The RST problem was first suggested by Hanan [Hanan 1966], who also proved the following result on the structure of optimal solutions. Let $G(S)$ be the grid induced by the point set S by drawing a horizontal and a vertical line through each point of S and retaining only the finite segments between intersection points of these lines (in the axis-aligned bounding box of S). Hanan proved that there exists a shortest RST for S which uses only segments in $G(S)$ [Hanan 1966] (see also [Richards 1989, Servit 1981]). In particular, this implies for n not too large that exact solutions can be found using an exhaustive search (in exponential time) [Hanan 1966].

It is well known that a minimum spanning tree in a weighted graph can easily be computed in polynomial time [Kruskal 1956, Prim 1957]. Furthermore, $O(n \log n)$-time algorithms have been developed for graphs embedded in the plane with Euclidean and rectilinear distances, see the papers [Hwang 1976] and [Shamos and Hoey 1975]. This is unlikely to be the case for Steiner trees: indeed, it has been shown that the Euclidean Steiner tree problem is NP-hard [Garey et al. 1997], and the rectilinear Steiner tree problem is NP-complete [Garey and Johnson 1977]. Many heuristics and approximation algorithms have been proposed over time to deal with the Steiner tree problem, particularly with the rectilinear variant. We only mention a few references [Bern 1988], [Fößmeier and Kaufmann 1997], [Hanan 1965], [Hanan 1966], [Hwang 1979], [Richards 1989], [Servit 1981], [Wee et al. 1994], [Zelikowsky 1993]. The rectilinear variant has been motivated by a fundamental problem in circuit design: how to connect n given points on a board to make them electrically common using the least amount of wire [Richards 1989]. For a variety of technical and engineering reasons, the segments of such connections are horizontal and vertical; also the orientation of the board (the underlying coordinate system) is part of the input for a given problem [Hanan 1966], [Richards 1989], so one is allowed to rotate the point set only by multiples of $90°$.

In 1976, Hwang [Hwang 1976] proved that the rectilinear Steiner ratio (the supremum ratio of the weights of a minimum rectilinear spanning tree and the minimum rectilinear Steiner tree over all finite point sets in the plane) is $3/2$. The analogous result for Euclidean Steiner ratio has been obtained only in 1990, when Du and Hwang [Du and Hwang 1992] showed that the ratio is $2/\sqrt{3} \approx 1.15$.

Optimal solutions to the rectilinear Steiner tree problem (with exponential running time) have been devised in several papers; Fößmeier and M. Kaufmann [Fößmeier and Kaufmann 1997] give an account of earlier and more up to date approaches. The earliest heuristic proposed for finding good approximations is a sweep-line algorithm attributed to Hanan [Hanan 1965], see also [Servit 1981], [Richards 1989]. Hanan's algorithm (cf. [Richards 1989]) processes the sites one at a time in increasing order of their x-coordinates. Initially, the first site is made a singleton tree. Each new site processed is connected to a closest point (in L_1 metric) of the current tree. To remove bias due to the left-to-right sweep direction, the procedure is repeated three times after rotating the point set by $90°$, and each time a new tree is constructed. Finally the best of the four trees is output by the algorithm. Richards showed that the algorithm can be implemented to run in $O(n \log n)$-time [Richards 1989]. Servit [Servit 1981] and Richards [Richards 1989] note that no tight or good bound were known on the approximation ratio of Hanan's sweep-line algorithm for approximating the minimum RST. We note there were no tight analysis for Hanan's heuristics under any other metric either.

In this paper we address this issue, and analyze the performance of two sweep-line algorithms proposed in the 1960s and 1970s for constructing an approximate minimum spanning tree and an approximate minimum Steiner tree, respectively. Although their performance on various instances was usually found satisfactory, bounds on their approximation ratio (the worst case ratio of the weights of the output tree and the optimum tree over all finite point sets in the plane) have been missing. The outlines of the two algorithms are as follows:

(A) Process the points one at a time in left to right order. Connect each point p to a closest point p' to the left of p, by a straight line segment.

(B) Process the points one at a time in left to right order. Connect each point p to a closest point (on vertices and edges) of the current tree on the left.

The trees generated by the two algorithms on a set of 8 points are depicted in Figure 1. It is easy to check that both (A) and (B) output planar straight line graphs under any metric L_q, $q \geq 1$. The second variant (B) has been primarily designed for constructing a rectilinear Steiner tree of the point set. It uses short-cut connections to the current tree on the left (and introduces Steiner points) with the intent of finding a lighter tree than the one obtained by algorithm (A).

2 The sweep-line algorithm for spanning tree construction

The sweep-line algorithm (A) processes the input points in left-to-right order and connects each point p to a closest point p' to the left of p by a straight line segment (i.e., p' is also an input point).

Theorem 1. *The approximation ratio achieved by the sweep-line algorithm (A) for spanning tree construction is $O(\log n)$. This ratio is asymptotically tight. The result holds under any metric L_q, $q \geq 1$.*

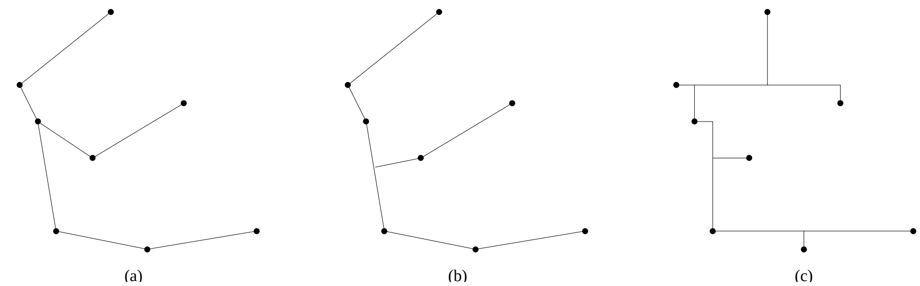

Figure 1: Three examples on a same point set. (a) A Euclidean spanning tree constructed by the sweep-line algorithm (using L_2 metric). (b) A Euclidean Steiner tree constructed by the sweep-line algorithm (using L_2 metric). (c) A rectilinear Steiner tree constructed by the sweep-line algorithm (using L_1 metric and L-shaped edges).

The upper bound easily follows from earlier results. In fact, we give two independent proofs: one is a reduction to the competitive ratio of an on-line greedy algorithm (placing the sweep-line algorithm into a broader class of algorithms), and the other one is a reduction to the gap theorem (with a more geometric flavor). See the details below. Even though the analyses of both the competitive ratio of on-line greedy algorithms and the gap theorem are tight, their lower bounds do not apply to the (more specialized) sweep-line heuristics. Here we present specific point sets for which algorithm (A) outputs a spanning tree which is $\Omega(\log n)$ times heavier than the EMST.

Remark 1. Since we are only interested in presenting asymptotic results, rather than exact constants of proportionality, we make use of the fact that any two metrics L_q, $q \geq 1$, are within constant factors from each other.

2.1 Upper bound via the gap theorem.

Chandra et al. [Chandra et al. 1995] (see also [Narasimhan and Smid 2007], pp. 108–119) introduced the *gap property*, which guarantees that a directed plane graph connecting n points in the plane is at most $O(\log n)$ times heavier than the EMST. For a directed segment edge (p, q), p is called the source, and q is called the sink. A set of directed edges satisfies the gap property, if the sources of any two distinct edges are "far" apart—relative to the length of the shorter of the two edges. Formally, for a real number $w \geq 0$, a set E of directed edges in the plane satisfies the gap property if and only if for any two distinct edges (p, q) and (r, s) in E, we have

$$|pr| > w \cdot \min\,(|pq|, |rs|).$$

By the *Gap Theorem* of Chandra et al. [Chandra et al. 1995], if S is a set of points in the plane, and $E \subseteq S \times S$ is a set of directed edges that satisfies the gap property for some $w > 0$, then the total weight $wt(E)$ of edges in E is bounded as

$$wt(E) < \left(1 + \frac{2}{w}\right) \cdot W \cdot \log n,$$

where W is the weight of the EMST of S.

Claim 1 *The sweep-line algorithm (A) computes a tree that satisfies the gap property with appropriately chosen edge directions and $w = 1$.*

Proof. Assume for simplicity that no two points have the same x-coordinate, so there is at most one point on the the sweep-line ℓ at any time. To apply the Gap Theorem, let E be the set of edges in the tree constructed by the sweep-line algorithm (A), where each edge (p, p') is oriented to the left. The algorithm connects each point p (on the sweep line ℓ) to the closest site p' left of ℓ. Let (p, q) and (r, s) be two edges in E, where r is to the right of p. By the rule of the algorithm the interior of the left half-disk of radius $|rs|$ centered at r is empty of sites. Therefore $|pr| \geq |rs| \geq \min(|pq|, |rs|)$. Hence E satisfies the gap property with $w = 1$. □

By the Gap Theorem, we have $wt(E) < 3 \cdot W \cdot \log n$, as required. The $O(\log n)$ ratio holds for any L_q metric, $q \geq 1$, by the same Gap Theorem combined with Remark 1.

2.2 Upper bound via on-line greedy algorithms.

The following is the *on-line Steiner tree problem* in the plane (according to [Alon and Azar 1993]): Suppose n points, $p_1, \ldots, p_n$ are presented one-by-one, and the objective is to construct, on-line, a connected graph that connects all of them, trying to minimize the total edge length. The points appear one at a time, p_i arriving at step i. In step 1, p_1 is a singleton graph T_1. In step $i \geq 2$, the on-line algorithm must augment T_{i-1} to a connected graph T_i that spans $p_1, p_2, \ldots, p_i$. If $A(p_1, \ldots, p_n)$ denotes the weight of T_n constructed by an on-line algorithm A, and $OPT(p_1, \ldots, p_n)$ denotes the length of an optimal (Euclidean) Steiner tree for this set of points, then the *competitive ratio* of the algorithm A is the supremum over all finite sequences $p_1, \ldots, p_n$, of the ratio

$$\frac{A(p_1, \ldots, p_n)}{OPT(p_1, \ldots, p_n)}.$$

A simple algorithm for the on-line Steiner tree problem is the *Steiner greedy algorithm*: At step i, it connects p_i to a closest point on vertices and edges of T_{i-1}. Consider now the *vertex greedy algorithm*: At step i, it connects p_i to the closest element in the set $\{p_1, \ldots, p_{i-1}\}$. Obviously, the vertex greedy algorithm constructs a spanning tree of the given points, in particular a Steiner tree for this set of points. Moreover, the Steiner greedy algorithm performs at least as well as the vertex greedy algorithm (since it uses shortcuts). It was shown by Imase and Waxman [Imase and Waxman 1991] that the competitive ratio of the vertex greedy algorithm is $O(\log n)$ in any metric space; a short proof of this result was also given by Alon and Azar [Alon and Azar 1993].

The sweep-line algorithm (A) applies the vertex greedy algorithm assuming that the input points are presented in left-to-right order. Hence it outputs a spanning tree which is at most $O(\log n)$ times heavier than the EMST.

2.3 Lower bound construction.

We now prove that the $O(\log n)$ factor is tight for every integer n, by constructing a suitable bad instance S. We first consider the L_2 (Euclidean) metric. Let $L = 2^k$ for some integer k. Points of S are on $k+1$ parallel almost vertical lines of (large) negative slope. For simplicity we first set these lines as vertical and slightly rotate the entire point set counterclockwise at the end. Figure 2 shows an example with $n = 53$ points on 5 vertical lines. The x-coordinate of the ith line ℓ_i is $x_i = i\epsilon$, for a small $\varepsilon \leq 1/n$ and $i = 0, 1, \ldots, k$. We construct the point set recursively. Place two points on line 0, with y-coordinates 0 and 2^k. Assume the points on the lines ℓ_0 through ℓ_{i-1} have been placed, and we now place the points on line ℓ_i. First copy the points on line ℓ_{i-1} onto line i at the same y-coordinate. For any two consecutive points on ℓ_{i-1} with y-coordinates y and $y + 2^a$, place points on ℓ_i at y-coordinates $y + 2^b$, for all $b = 0, 1, \ldots, a-1$. Note that the distance between two consecutive points along any of the lines is always a power of 2; this invariant is maintained inductively.

Claim 2 *Algorithm (A) outputs a tree of length $\Omega(W \log n)$ for the points set S described above, where W denotes the weight of the EMST of S.*

Proof. Points are swept in the order $\ell_0, \ell_1, \ldots, \ell_k$, and for each line ℓ_i from top to bottom. Each point on ℓ_i except the topmost, is connected to the one above it on ℓ_i, unless there is a corresponding point with same y-coordinate on ℓ_{i-1} (this holds by construction). In case there is a corresponding point, this connection is made by the algorithm for the current point on ℓ_i.

The EMST (of weight W) consists of a segment along ℓ_k of length L and horizontal connections with all other points made from ℓ_k. Observe that $L \leq W \leq L + n\varepsilon \leq L + 1$, so $W \leq 2^k + 1$. Let T' be the tree constructed by the sweep-line algorithm, and let E, and W' denote its edge set and total weight. Next we establish a lower bound on W'.

For brevity, we refer to the almost vertical lines ℓ_i, $i = 0, 1, \ldots, k$, as *vertical*. For $i \geq 0$, denote by $T(2^i)$ the total cost of vertical edges in E "generated" by a vertical segment in E of length 2^i which connects two consecutive points with

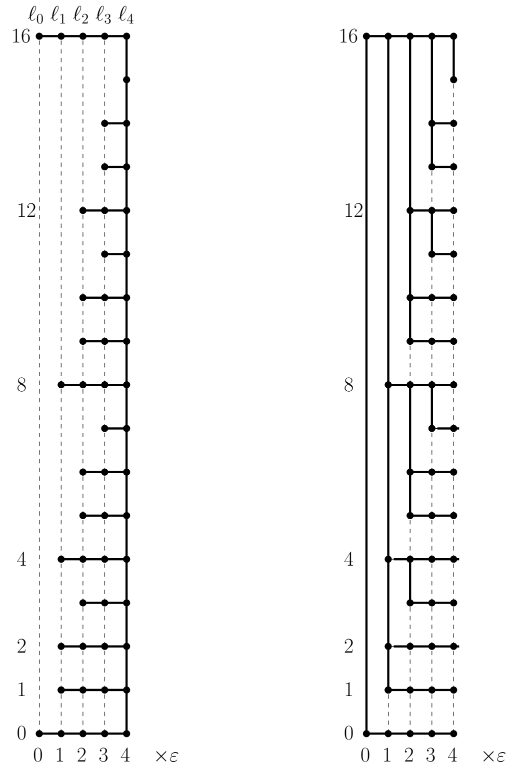

Figure 2: An example with $|S| = n = 53$ points. Left: a minimum spanning tree (EMST) of S of weight $16+o(1)$. Right: the spanning tree of weight $48+o(1)$ constructed by the sweep-line algorithm.

y-coordinates y and $y+2^i$ (at distance 2^i) on some line ℓ_j; this total cost includes the lengths of vertical segments on $\ell_j, \ell_{j+1}, \ldots, \ell_k$ whose both endpoints have y-coordinates in $[y, y+2^i]$. Clearly $W' \geq T(2^k)$. By construction, $T(2^i)$ satisfies the following recurrence

$$T(2^i) = 2^i + \sum_{j=0}^{i-1} T(2^j).$$

Some initial values are easy to compute (and the reader can verify $T(2^4)$ on the figure):

$$T(2^0) = 1,\ T(2^1) = 2+1 = 3,\ T(2^2) = 4+3+1 = 8,\ T(2^3) = 8+7+3+1+1 = 20,$$

$$T(2^4) = 16+15+11+5+1 = 48, T(2^5) = 32+48+20+8+3+1 = 112.$$

Introduce a new variable

$$U(i) = e^{T(2^i)}.$$

The recurrence now becomes

$$U(i) = e^{2^i} \prod_{j=0}^{i-1} U(i),$$

with initial condition $U(0) = e$. This can be further rewritten as

$$U(i+1) = e^{2^i} [U(i)]^2.$$

Put now $U(i) = e^{V(i)}$, which yields the recurrence in $V(i)$

$$V(i+1) = 2V(i) + 2^i,$$

and whose corresponding initial condition is $V(0) = 1$. Solving for $V(i)$ yields

$$V(i) = 2^{i-1}(i+2),$$

and consequently

$$U(i) = e^{2^{i-1}(i+2)}.$$

Going further back, we recover

$$T(2^i) = 2^{i-1}(i+2).$$

Of course, we are only interested in $T(2^k) = 2^{k-1}(k+2)$. Clearly, the number of points in the constructed set S is at most $|S| \leq (k+1)(2^k+1)$, but in fact $|S| \leq k2^k$ is also easy to derive. This implies that $\log n \leq k + \log k$, hence $k \geq \log n - \log\log n \geq \frac{\log n}{2}$. The weight W' of the tree constructed by the sweep-line algorithm is

$$W' \geq T(2^k) = 2^{k-1}(k+2) \geq \frac{L}{2} \cdot \frac{\log n}{2} = \Omega(W \log n),$$

as claimed.

By Remark 1, the lower bound holds for any metric L_q, $q \geq 1$. □

Remark 2. Our $\Omega(\log n)$ lower bound on the approximation ratio holds also for the optimized version of the sweep-line algorithm, in which the best of the four trees (for each axis-aligned sweep direction) is output by the algorithm: The construction can be repeated along each of the four sides of a square of side length $L = 2^k$. Then W is about $4L$, while W' is still $\Omega(W \log n)$, as desired.

Remark 3. Our lower bound construction does not seem to extend for the more powerful algorithm that first chooses an orthogonal coordinate system in the plane and then performs the sweep-line algorithm (A). It is an open problem whether for any point set, there is a coordinate system in which algorithm (A) computes a spanning tree only $o(\log n)$ times heavier than the EMST.

3 The sweep-line algorithm for Steiner tree construction

The sweep-line algorithm (B) processes the input points in left-to-right order and constructs a Steiner tree by connecting each point p to the closest point (on vertices and edges) of the current tree. Two examples of the trees constructed using L_2 and L_1 distances are shown in Figure 1(b) and 1(c), respectively. The rectilinear variant is known as Hanan's heuristic for the minimum rectilinear Steiner tree problem [Hanan 1965]: each edge is drawn as an L-shape with the horizontal segment of the L-shape on the left of the vertical segment (i.e., the L-shape overlaps with the vertical line through the current point). Richards provided an implementation of this algorithm that runs in $O(n \log n)$ time [Richards 1989].

Since the weights of a minimum spanning tree and a minimum Steiner tree are all within constant factors of each other under any metric L_q, $q \geq 1$, it is clear that the approximation ratio achieved by algorithm (B) is bounded by that of algorithm (A), hence it is also $O(\log n)$. We analyze both variants of algorithm (B), when connections are done by a segment, or by a an axis-aligned L-shape. We show that neither of these yields a constant ratio approximation algorithm for Steiner tree construction. The reader will observe that our lower bound construction precludes the algorithm in performing any point to tree (edge) connection, except the standard point to point connections from variant (A).

Theorem 2. *The worst case approximation ratio achieved by the sweep-line algorithm for Steiner tree construction is* $\Omega(\log n / \log \log n)$*. The result holds in every metric* L_q*,* $q \geq 1$*.*

Our lower bound construction is reminiscent of the point configuration proposed by Alon and Azar [Alon and Azar 1993] for the *on-line* Steiner tree problem of n points in the plane, that yields a $\Omega(\log n / \log \log n)$ lower bound on the competitive ratio of any on-line algorithm for that problem. The sites are revealed to the algorithm in increasing y-coordinates (similarly to the sweep-line algorithms), but an on-line algorithm is allowed to augment the graph with arbitrary new edges and Steiner points at each step, as opposed to the sweep-line algorithm (B) which is bound to do one direct connection to the tree. Hence Theorem 2 follows from the construction in [Alon and Azar 1993]. While the argument in [Alon and Azar 1993] is more elaborate and uses an adversary who selects the positions of the layers (lines) revealed to the on-line algorithm at various moments, our argument proceeds with a direct calculation of the resulting weight (and is simplified in the sense that the layers are at fixed positions, and the points on each layer are conveniently ordered).

Proof. Let $L = 2^k$, where $k = 2^{2^a}$ for some positive integer $a \geq 1$. Let

$$f(k) = \log k - \log \log k = 2^a - a,$$

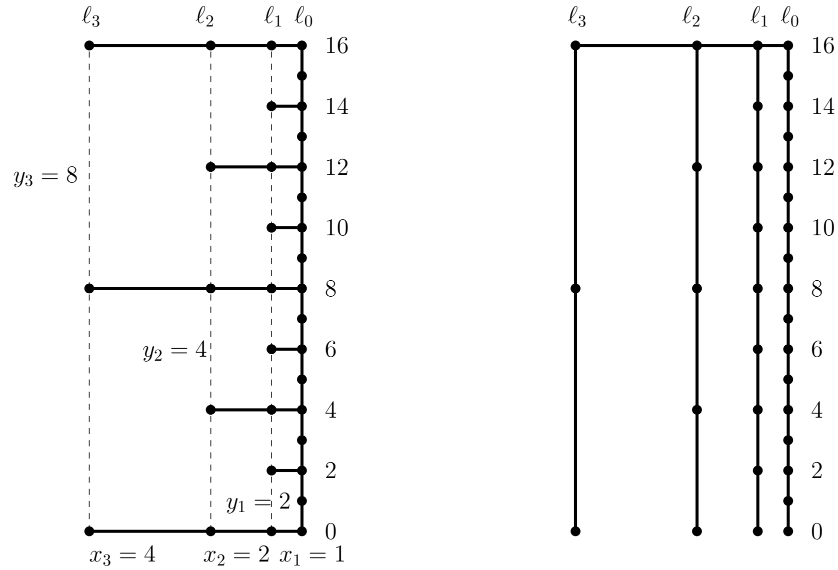

Figure 3: An example with $L = 2^4 = 16$, and $|S| = n = 34$ points. Left: a minimum spanning tree (EMST) of S of weight 47. Right: the Steiner tree of weight 55 constructed by the sweep-line algorithm (B).

and write $z = f(k)$. Clearly $z \geq 1$. Note that

$$2^z = \frac{k}{\log k}, \quad \text{hence} \quad z \cdot 2^z = \frac{(\log k - \log\log k) \cdot k}{\log k} \geq \frac{k}{2}.$$

The n points of S are placed on vertical lines $\ell_0, \ldots, \ell_s$, numbered from right to left, with $m_i + 1$ equidistant points on line ℓ_i having y-coordinates between 0 and L (see Figure 3). Thus m_i denotes the number of segments (between consecutive points) on ℓ_i and let $y_i = L/m_i$ be the common length of these segments. Denote by x_i the horizontal distance between ℓ_{i-1} and ℓ_i. The values of these parameters are

$$m_i = 2^{k-iz}, \; x_i = 2^{(i-1)z}, \; (i \geq 0)$$

The index s of the leftmost line ℓ_s is the largest i for which $m_i \geq 2$ holds. We thus have

$$\frac{k}{\log k} \leq s \leq \frac{2k}{\log k}.$$

Observe that $s \leq k$. The relation between n and k will be established in the final part of the proof.

To transform the point set so that no two points have the same x-coordinates (and the sweep-line encounters the points in a unique order), we slightly enlarge the horizontal distance between consecutive vertical lines by ε and then slightly rotate each line counter-clockwise about its lowest point by an angle of ε^2 for a sufficiently small $\varepsilon > 0$. We will however omit these slight adjustments for simplicity and without loss of generality when analyzing the algorithm.

The sweep-line algorithm connects the top point on vertical line ℓ_i (for $i = 0, \ldots, s-1$) to the top point of ℓ_{i+1}, and then in descending order, connects each point on ℓ_i to the next point above it on the same line. Then it moves to the next vertical line to the right. Consequently, the weight W' of the tree constructed by the sweep-line algorithm includes a weight of L on each of the lines $\ell_0, \ldots, \ell_k$, hence

$$W' \geq (s+1)L \geq \frac{k}{\log k} \cdot L.$$

The weight W of the EMST of S is (without really needing a justification, since we only need an upper bound on W):

$$W = m_0 y_0 + \sum_{i=1}^{s} (m_i + 1) x_i = L + \sum_{i=1}^{s} m_i x_i + \sum_{i=1}^{s} x_i.$$

For $i \geq 1$, we have

$$m_i x_i = 2^{k-iz} \cdot 2^{(i-1)z} = 2^{k-z}.$$

The sum $\sum_{i=1}^{s} m_i x_i$ is then bounded from above as follows

$$\sum_{i=1}^{s} m_i x_i = s \cdot 2^{k-z} \leq \frac{2k}{\log k} \cdot \frac{L}{2^z} = L \cdot \frac{2k}{\log k} \cdot \frac{\log k}{k} = 2L.$$

Now since $m_i \geq 2$ for each $i = 1, \ldots, s$, this also implies that

$$\sum_{i=1}^{s} x_i \leq \frac{1}{2} \sum_{i=1}^{s} m_i x_i \leq L.$$

Overall,

$$W \leq L + 2L + L = 4L.$$

The relation between the number of points n and k is: $2^k \leq n \leq s2^k$. We derive that $2^k \leq n \leq k2^k \leq 2^{2k}$, or equivalently,

$$\frac{\log n}{2} \leq k \leq \log n.$$

We conclude that the ratio between the two weights is

$$\frac{W'}{W} \geq \frac{k}{\log k} \cdot \frac{L}{4L} = \Omega\left(\frac{k}{\log k}\right) = \Omega\left(\frac{\log n}{\log \log n}\right),$$

as required.

Again, the lower bound holds under any metric L_q, $q \geq 1$, by Remark 1. $\square$

Remark 4. With the same modification of the construction described in Section 2, our $\Omega(\log n/\log\log n)$ lower bound on the approximation ratio holds also for the optimized version of the sweep-line algorithm (B), in which the best of the four trees (for each axis-aligned sweep direction) is output by the algorithm.

Remark 5. A construction similar to ours in Theorem 2 yields a $\Omega(\log n/\text{loglog}\, n)$ lower bound on the approximation ratio of the *insertion method* for constructing TSP tours of n points (i.e., in the Euclidean traveling salesman problem). The insertion method [Rosenkrantz et al. 1977] adds the points of S one by one in some order to the current tour until a complete tour is obtained. The new point is inserted between two consecutive points of the current tour, which yields the minimum increase in the cost of the tour. Conform to results obtained by Bafna et al. [Bafna et al. 1994] and Azar [Azar 1994], some insertion orders produce tours of length $\Omega(\log n/\log\log n)$ times the optimal. These are in fact generated by processing the points in a sweep-line order; see also [Bern and Eppstein 1997].

References

[Alon and Azar 1993] Alon, N., Azar, Y.: "On-line Steiner trees in the Euclidean plane"; Discrete & Computational Geometry 10 (1993), 113-121.

[Azar 1994] Azar, Y.: "Lower bounds for insertion methods for TSP"; Combinatorics, Probability, and Computing 3 (1994), 285-292.

[Bafna et al. 1994] Bafna, V., Kalyanasundaram, B., Pruhs, K.: "Not all insertion methods yield constant approximate tours in the Euclidean plane"; Theoretical Computer Science 125 (1994), 345-353.

[Bern 1988] Bern, M. W.: "Two probabilistic results on rectilinear Steiner trees"; Algorithmica 3 (1988), 191-204.

[Bern and Eppstein 1997] Bern, M., Eppstein, D.: "Approximation algorithms for geometric problems"; In: Approximation Algorithms for NP-hard Problems (Hochbaum, D. S., Ed.), PWS Publishing Company, Boston, MA (1997), 296-345.

[Chandra et al. 1995] Chandra, B., Das, G., Narasimhan, G., Soares, J.: "New sparseness results on graph spanners"; International Journal on Computational Geometry and Applications 5 (1995), 125-144.

[Du and Hwang 1992] Du, D. -Z., Hwang, F. K.: "A proof of Gilbert-Pollak's conjecture on the Steiner ratio"; Algorithmica 7 (1992), 121-135.

[Fößmeier and Kaufmann 1997] Fößmeier, U., Kaufmann, M.: "Solving rectilinear Steiner tree problems exactly in theory and practice"; Proc. 5th European Symposium on Algorithms (ESA'97), LNCS 1284, Springer-Verlag (1997), 171-185.

[Garey et al. 1997] Garey, M. R., Graham, R. I., Johnson, D. S.: "The complexity of computing Steiner minimal trees"; SIAM Journal on Applied Mathematics 32 (1977), 835-859.

[Garey and Johnson 1977] Garey, M. R., Johnson, D. S.: "The rectilinear Steiner tree problem is NP-complete"; SIAM Journal on Applied Mathematics 32 (1977), 826-834.

[Gilbert and Pollak 1968] Gilbert, E. N., Pollak, H. O.: "Steiner minimal trees"; SIAM Journal on Applied Mathematics 16 (1968), 1-29.

[Hanan 1965] Hanan, M.: "Net wiring for large scale integrated circuits"; IBM Technical Report RC 1375 (1965), 1-17.

[Hanan 1966] Hanan, M.: "On Steiner's problem with rectilinear distance"; SIAM Journal on Applied Mathematics 14 (1966), 255-265.

[Hwang 1976] Hwang, F. K.: "On Steiner minimal trees with rectilinear distance"; SIAM Journal on Applied Mathematics 30 (1976), 104-114.

[Hwang 1979] Hwang, F. K.: "An $O(n \log n)$ algorithm for suboptimal rectilinear Steiner trees"; IEEE Transactions on Circuits and Systems 26 (1979), 75-77.

[Imase and Waxman 1991] Imase, M., Waxman, B. M.: "Dynamic Steiner tree problem"; SIAM Journal on Discrete Mathematics 4 (1991), 369-384.

[Kruskal 1956] Kruskal, J. B.: "On the shortest spanning subtree and the traveling salesman problem"; Proc. American Mathematical Society 7 (1956), 48-50.

[Narasimhan and Smid 2007] Narasimhan, G., Smid, M.: "Geometric Spanner Networks"; Cambridge University Press (2007).

[Prim 1957] Prim, R. C.: "Shortest connection networks and some generalizations"; Bell System Technical Journal 36 (1957), 1389-1401.

[Richards 1989] Richards, D. S.: "Fast heuristic algorithms for rectilinear Steiner trees"; Algorithmica 4 (1989), 191-207.

[Rosenkrantz et al. 1977] Rosenkrantz, D. J., Stearns, R. E., Lewis, P. M.: "An analysis of several heuristics for the traveling salesman problem"; SIAM Journal on Computing 6 (1977), 563-581.

[Servit 1981] Servit, M.: "Heuristic algorithms for rectilinear Steiner trees"; Digital Processes 7 (1981), 21-32.

[Shamos and Hoey 1975] Shamos, M. I., Hoey, D.: "Closest point problems"; Proc. 16th Annual Symposium on Foundations of Computer Science (FOCS'75), IEEE (1975), 151-162.

[Wee et al. 1994] Wee, Y. C., Chaiken, S., Ravi, S. S.: "Rectilinear Steiner tree heuristics and minimum spanning tree algorithms using geographic nearest neighbors"; Algorithmica 12 (1994), 421-435.

[Zelikowsky 1993] Zelikowsky, A. Z.: "An 11/6 approximation algorithm for the network Steiner problem"; Algorithmica 9 (1993), 463-470.